UNIVERSITY OF MICHIGAN

COMPUTING CENTER

MTS OPERATORS' MANUAL

Revised   Feb 21, 1995

NOTE: Revision bars, "|", will be used to point out information which has been entered, changed or deleted at the last two or three printings of this manual.

For additions or corrections,
contact Stephen M. Donnelly

Contents

Quick Reference Guide

OVERVIEW OF THE SYSTEM

MTS is a system developed by the University of Michigan to take advantage of dynamic relocation hardware. Basically, MTS allows a large number of users to make use of the computer as if they each had their own computer. It does this by using Virtual Memory and Multi-Tasking Techniques.

The virtual memory concept is simply a method of simulating a computer with an extremely large memory. This is done by dividing both REAL and USER memory into pages (of 4096 bytes each). USER pages are loaded, if possible, into real memory; however, since users may request more pages of memory than is, or ever will be, available in the real computer, a large number of user pages must be kept on Direct Access storage devices (primarily Fixed Head Files, or their Solid-State replacements, but a disk may be also used under heavy loading). Of course, the CPUs can only work with instructions and data contained in real memory; thus, a method must be devised to make User memory pages available to the CPU when the CPU requires them. Pages are made to move back and forth between real memory and the paging device under control of the Paging Drum Processor (PDP) job program. The supervisor communicates with the PDP telling it what pages should be read from the paging device and what pages are eligible to be moved out to be moved out to the paging device.

Multi-Tasking is a technique that allows the computer to be used by many users at the same time. Since the CPUs can only perform one instruction at a time (in a single-processor system), it may seem at first glance that there is little to be gained from Multi-Tasking. This, of course, is not true, the reason being that the CPU spends much of it's time doing nothing except waiting for I/O operations to be completed. Multi-Tasking allows this CPU wait-time to be used by other users of the system. In addition, more than one I/O operation can be going on at once. For example, while the CPU is busy doing a compute function, many I/O devices can be going at full speed.

Unfortunately, something cannot be gained for nothing and both Paging and Multi-Tasking require resources of the computer (time and memory) to manage or supervise these functions. Normally the gains far outweigh the losses and throughput is increased. But, under extremely heavy load, the CPU time required to manage Paging may rise to as much as thirty or forty percent of the total available time. When this happens, user pages reside in real memory for relatively short periods of time and thus user programs run for relatively short periods of time (until a reference is made to a page that is not in real memory). This condition is called Thrashing and is very undesirable.

OVERVIEW OF THE SYSTEM

The core of the MTS system is a program formally called the University of Michigan Multiprogramming Supervisor (UMMPS). UMMPS has the characteristics of a 'supervisor' in that 1) it operates in supervisor state (and is almost the only code in the operating system that does so), 2) it deals with all interrupts, and 3) it allocates resources such as devices, CPU time and storage. UMMPS executes JOBS which are usually initiated from the operator's console. Each job runs in problem state and calls upon the supervisor for its services. In the following, the terms JOB and TASK are synonymous. We shall call a JOB PROGRAM that section of code that can be invoked (with the appropriate command) from the operator's console. Each invocation of a JOB PROGRAM is called a JOB or TASK. UMMPS oversees all the computer's multiple-user-oriented activities, including a system called the Michigan Terminal System (MTS). Although MTS is a complex set of programs in itself, to UMMPS it is merely one of a number of job programs. Functionally, then, we have a hierarchy of command levels. UMMPS is in complete control of the system; it allocates the use of all hardware. MTS is one of several job programs that run under the supervision of UMMPS. The typical user of MTS communicates with the system via MTS commands in his command stream (examples of such commands are $SIG, $CRE, $EMP, $LIS, $RUN) ; under normal circumstances such (MTS) commands are meaningless if entered at the operator's console since it is UMMPS (or the OPERATOR job), not MTS, that services the console. The user never sees UMMPS; the interface between MTS and UMMPS is hidden from him. When he makes a request of MTS, MTS asks UMMPS for whatever it needs to handle that request. The MTS job program is reentrant. Thus, there can be many MTS tasks using the same MTS job program.

A user may operate within MTS in one of two modes: conversational mode or batch mode. When operating in conversational mode, the user communicates directly with MTS via a terminal in real time --- each command typed by the user at his terminal is immediately processed by MTS, and the results of this processing are made known to the user at his terminal. The user may decide what needs to be done and enter the appropriate MTS command. The user, thus, carries on a "conversation" in real time with MTS via his terminal; hence, the term "conversational mode" is used to describe this type of interaction.

In batch mode, the user used to keypunch a card deck which contained all the MTS commands, translator source statements, data lines, and object modules which are needed in order to accomplish his described task. The deck was then submitted to MTS for processing (note: the *BATCH* MTS pseudo-device may be used to submit a Batch job from a user's terminal). The user's card deck was then read into the computer (by a job program named HASP) and saved in a file for later processing. The deck and a receipt were returned to the user. The user's job is then processed by MTS together with many other batch jobs and eventually the results of this processing (the job output) may be retrieved by the user with a receipt.

The MTS batch function is controlled by the Resource Manager (RM). The Resource Manager replaces the HASP (Houston Automatic Spooling System). HASP was originally developed for the IBM OS system, has been revised to interface with the MTS system and accept MTS jobs to run in batch. (IBM OS/SVS/MVS/VM/CMS jobs jobs cannot be run directly under the MTS system.)

A user communicates with MTS by means of a command language. A typical job in batch mode will have command lines, a source program, data, and possibly an object program. The commands, placed in a precise sequence, obtain the system's attention, tell it what to do, how to do it, and what data to use. The command language employed is essentially the same whether the user, in the past, punched it on a card for batch processing or types it directly on an interactive terminal. For example, every user starts with the command '$SIGNON' and ends with the command '$SIGNOFF'. For convenience, these and other commands can be abbreviated.

Input data and programs may be included in a source deck or they may be stored in the MTS file storage system. Three major types of files exist in the MTS system: 1) public files, 2) private-permanent files, and 3) private-temporary files.

Public files, which usually contain a system program (such as the FORTRAN compiler) in loadable form, are available to all users. For the most part, they can read, but not change, the contents of public files --- some public files can be extended (or added-to) by the normal user (in this manner information can be added to an existing file but none of the previous contents will be destroyed). Private permanent files are normally available only to the user who has created them, to read from, write into, modify, or destroy as he wishes. Access to a user's private permanent files is through the use of his i.d. The user may, through the use of the $PERMIT command, expand access to or restrict access by other users in various ways -- one may even restrict one's own access to one's <u>own</u> file. Temporary private files may be created by a user during the processing of a job. These are automatically destroyed at SIGNOFF-time.

All three types of files may be organized as either line files or sequential files. Line files have a very useful indexed-sequential structure --- i.e., each logical record within a line file has a unique line number associated with it so that a user may <u>easily</u> request access to only one or a range of the lines in the file. Perhaps the main advantage of line files is that they are easily edited. Sequential files are used for storing large amounts of data that need not be in a form that may be easily edited.

<u>Note</u>: Sequential files are seldom used any longer since the

lines in a line file can be much longer than in the past.

Which form a user chooses for a private file depends on the nature of data and the ease of access (for editing purposes) that one wishes to have with respect to that file. When a file is created for a user, the user may specify the initial amount of storage to be set aside for it. If it is not specified, MTS assigns a small amount of storage, and will automatically expand the size later if necessary until some limit is reached.

Virtual machine capablity is available under the MTS system.

MTS provides the users with memory for problem programs in a restricted fashion, i.e., there *is* a maximum size. The limits imposed have no relation to the physical real memory size of the computer.

MTS has a comprehensive accounting facility as an integral part of the system. Only authorized persons who have been assigned a signon-id, account number, and project number can use the system. At registration time, maximums can be established for such items as disk space, terminal time, and plotting time. In addition, an expiration date can be set for a user (useful for students taking a programming course) as well as a maximum dollar charge.

The charge is debited from the user's account for each job run. Charges for the job are based on such computer resources as CPU time, memory usage, connect time (for terminals), cards read, lines printed, and cards punched.

Note: Cards are all but obsolete.

Additional charges are made for disk storage. Users have access to their own accounting information at any time. A facility also exists whereby a project director or course instructor can allocate and manipulate funds, disk space, and other resources among users within his project.

The Job Program and the Task

A "job program" is a program which can be started from the main operator's console or at the request of any running task. A "task" is the running invocation of a job program. There is one job program for each system function; depending upon the function that a job program is supposed to perform, there may be one or more invocations (one or more tasks) of that job program running at the same time. Another name for a "task" is a "job"; the two terms will be used interchangably throughout this manual.

The most important job program is the MTS job program. There is one invocation of this job program (one "MTS task") for every terminal line connected to the system; in addition, a new MTS task is created each time a batch job is run. Other examples of job programs include the OPERATOR job, which controls a terminal and accepts commands to manage the system, the Disk Manager, which manages access to the system's disks, and the Paging Drum Processor, which manages paging activity.

A "job number" or "task number" is associated with every running task. The job number is assigned at the time a task is started, and may be used to refer to the job in controlling the system. For example, the STOP job takes as a parameter the number of some other job which is to be terminated.

A task is started by a request from the main (or backup) operator's console (which are described later). A task request consists of the name of the job program to be started, optionally followed by one or more parameters. A complete description of each job program is given later in this manual.

Once started, a task may continue execution indefinitely, or it may terminate (stop running). Whether and when a job terminates depends upon the particular job program, upon the parameters given upon the job request, and upon the actions of the system operator.

The Main and Backup Consoles

The main console is a 3270 terminal which is controlled by the OPERATOR job (i.e., by an invocation of the OPERATOR job program). The main console (also called the Operator's Console) is the primary terminal used to control the action of the system. Most of the commands issued from the operator's console are job requests; a complete description of the operator's console is given later in this manual.

Occasionally the operator's console may become unusable, either because of hardware problems or programming errors in the system. In such circumstances, a second terminal, the "backup console" may be started; the system can then be controlled from the backup console until the main operator's console can be made to work again.

Sometimes the system may automatically start the backup console, if it determines that the main console is unusable; if not, an operator can start the backup console by issuing a signal to the machine. Normally, the signal consists of an external interrupt, that is, use the ALT/IRPT keys on the IBM console.

UM operators should consult the section "Running MTS on the ES3090/300J "

Forcing a Superdump

Occasionally the system may detect a serious error in its operation; in this case, a message is printed on the main (or backup, if active) operator's console. The system requests that a tape be provided for a "superdump" -- a dump of all of real memory. Following the superdump, the system terminates operation and must be reloaded.

An operator can force the system to take a superdump at any time by sending a special signal to the CPU. Normally this is only done when a system problem occurs and all possible recovery measures have failed. On most machines, the signal consists of a restart interrupt (the "RESTART" key). On the ES3090/300J, use ALT/RSTRT.

Mainframe and Peripheral Power

Powering the System Down

After the systems have been shut down it is possible to turn power off and on to the IBM ES3090/300J and the I/O equipment.

To turn IBM ES3090/300J power off-

1. Go to the Processor Controller and press the yellow Power Off button.

   The above action turns off the CPUs, the processor and the MGs but not any of the I/O equipment such as disks and tapes.

2. Power down the IBM 3380 disk drives - go to the 3880 IBM "Disk Controller" and turn the Subsystem Power switch off. This should power down the IBM disk system.

3. Power down the IBM Tape Drives - go to the 3803 Tape Control Unit.

   a. Turn the DC switch off first.

   b. Turn the AC switch off.

4. Power down the Amdahl disk equipment

   a. Make sure that the Amdahl disks are offline to the system, unless the system is already powered down.
      1. (If only one disk is to be powered off, offline it, open its door and turn off the associated switch. The green switch puts the disk into and out of READY state. Power will still be available until the power supply is turned off. To properly turn this off, call an Amdahl representative.)

   b. Open each disk drive door.

   c. Turn off all green tipped switches that are READY.

   d. Open the Control Unit door.

   e. Make sure that the Local/Remote switch is in LOCAL. If the switch had been in remote, the disks should have come down with the main system.

   f. Turn the silver switch labelled BOTH to the "off" positioned which should sequence the disk string down. Do this with both control units.

Note: There are two control units called "Directors" in each Control Unit box.

5. Power down the Intel virtual memory equipment.

Powering the System Up

Note: Before powering up the ES3090/300J , do the following:

Select B2 from the "system definition" frame (22) before an LPAR POWER-ON-RESET (POR).

Type "west" for the zone definition and the current time in the ZONE OFFSET slot. This allows the ES3090/300J to POR LPAR. Following this, the SYSTEM CONFIGURATION FRAME should be displayed by pressing the INDEX key and typing O5. Now that the SYSTEM CONFIGURATION FRAME is present, type A2 for the POWER-ON-RESET. This should be done on the UM side only since LPAR doesn't exist on the "B" side of the 3090.

To power up the IBM ES3090/300J -

1. Go to the Processor Controller and press the green Power On button. This action will power up the CPUs, the MGs and the Processor.

2. Go to the 3880 Disk Controller and press the Subsystem Power switch. This should power on the 3380 disks. If some of the disks do not power up, open the panel doors to the set of drives and turn the Front Drive and Rear Drive switches off and back on.

   If you have to use the Front and Rear Drive switches to power up the disks, then you'll have to go back to the Disk Controller and, one at a time, turn the Storage Direct Power switches (SD1 and SD2) off and back on.

Running UM-MTS on the ES3090/300J

There are two ways to run MTS on the 3090: native, and under control of VM. Currently at UM, we are running UM-MTS native, and running UB-MTS under control of VM/XA. The following discussion applies only to running UM-MTS native on the 3090; see the sections "Running MTS under VM" for information on running UB-MTS under VM.

In order to run MTS native on the 3090, you need to know how to do five things:

1) Load MTS

2) Enable the time-of-day clock set control

3) Switch to a backup console

4) Force a superdump

5) Stop the CPU after a shutdown

Loading UM-MTS

This should only be done after powering up the ES3090/300J or after shutting UM-MTS down. To load UM-MTS perform the following steps:

1. Go to the ES3090/300J console for the UM side of the 3090.

2. Press the INDEX key

3. Select the OPRCTL frame by typing 9 and pressing ENTER

4. Make sure that the load address on the OPRCTL frame is correct. The load address should be given on a note near the UM-MTS operator's console.

   If the load address is incorrect:

   a. type L1 and press ENTER. Your cursor will move to the LOAD UNIT ADDR field.

   b. Enter the load address and press ENTER.

5. Start the load command by typing "O4" and pressing enter. (That's the letter "oh", not the number zero.)

6. After a moment, the message "Load Complete" should

appear on the ES3090/300J console.

7. A minute or so later, the message "Do You want to run the current system" should appear on the UM-MTS operator console.

8. See the section "Loading MTS" to proceed from here.

Setting the TOD clock

If you change the setting of the TOD clock during a reload (see the section "Loading MTS"), UM-MTS may tell you to "Enable the TOD clock". To do this:

1. Go to the ES3090/300J console for the UM side of the 3090.

2. Press the "alt" key and, while holding it down, the "TOD" key. Under some circumstances, you may need to enable the TOD clock twice. If UM-MTS prompts you a second time, follow the above procedure again.

Switching to the Backup Console

To start the backup UM-MTS operator console at any time, you must make sure the proper terminal is attached to UM-MTS and signal an external interrupt.

1. Go to the ES3090/300J console for the UM side of the 3090.

2. Press the "alt" key and, while holding it down, press the "IRPT" key.

3. The backup console should appear on the terminal.

Forcing a UM-MTS Superdump

To cause a UM-MTS superdump, you must signal a restart interrupt.

To do this:

1. Go to the ES3090/300J console for the UM side of the 3090.

2. Press the "alt" key and, while holding it down, press the "RSTRT" key.

Stopping the CPU

You must do this as the last step in shutting down UM-MTS:

Go to the ES3090/300J console for the UM side of the 3090.

2. Press the "alt" key and, while holding it down, press the "STOP" key.

Note: There is no longer a UB system.

Running UB-MTS under VM/XA

There are two ways to run MTS on the 3090: native, and under control of VM/XA. Currently at UM, we are running UM-MTS native, and running UB-MTS under control of VM. The following discussion applies only to running UB-MTS under VM/XA on the 3090; see the section "Running MTS on the 3090" for information on running UM-MTS native.

VM/XA is an IBM operating system that runs on the ES3090/300J and controls its operation. One of VM's functions is to provide an environment in which UB-MTS can run. For more information on VM, see the section "Running VM/XA".

In order to run UB-MTS under VM/XA, you need to know how to do the following:

1. Load UB-MTS.

2. Switch to a backup UB-MTS console.

3. Force a UB-MTS superdump.

4. Stop the UB-MTS virtual machine after a shutdown.

5. Attach and Detach tape drives.

6. Control I/O devices.

It should be noted that there is never any need to set the TOD clock during an UB-MTS reload when running under VM/XA. The clock must be set when you load VM/XA. If you notice that the clock is set incorrectly when you are reloading UB-MTS, you will need to reload VM/XA to set it properly.

Some of the above procedures require that you find a VM/XA terminal and sign on to VM/XA. The 3278 terminal located behind the tape drives is connexted to both UM-MTS and VM/XA. The black box next to the terminal has a switch that indicates whether the terminal is connected to UM-MTS or VM/XA. To switch the terminal to VM/XA:

1. Turn the switch on the terminal's black box to "VM/XA".

2. Switch the TEST/NORMAL switch to the right of the screen to TEST and then back to NORMAL.

3. At this point, the VM/XA logo should appear on the screen.

After VM has been loaded and is running, one can use the following commands from the UB operators' console-

```
%cp q names           ( this shows what is running
under VM)

%cp attach           (attach devices to VM)

%cp detach           (detach devices from VM)

%cp q tapes           (VM tape information)
```

Attaching and Detaching Tape drives: When UB-MTS runs under VM, it is very important that no tape drives be attached to MTS except for the ones that UB-MTS is using at the moment. If a tape drive is attached to UB-MTS but in use from the UM-MTS system, it is possible that VM cound cause the tape to be rewound unexpectedly; this would obviously cause problems for the UM-MTS user of the tape.

To avoid problems of this sort, it is necessary to issue an explicit ATTACH or DETACH command at the VM operator's console to make a tape drive available or unavailable to UB-MTS. These commands are issued automatically by MTS for tapes that are normally mounted; however, the commands must be issued by the operator under other circumstances. For example, if an MTS task that is using a tape SNARKs or is BLASTed, MTS will be unable to automatically issue the required DETACH command, so the operator will need to issue the DETACH command at the VM operator's console. Another example of a case in which it would be necessary to issue manual ATTACH and DETACH commands would be to run job programs like the STAT job, or to issue WTM commands to write tape marks.

VM refers to tape drives by number, rather than by name as does MTS. The following table lists the MTS name and the VM device number for each tape drive; the device number must be specified on the ATTACH and DETACH commands as shown below.

Name	Number	Name	Number	Name	Number
T900	0680	T907	0687	T914	068E
T901	0681	T908	0688	T915	068F
T902	0682	T909	0689	T920	1680
T903	0683	T910	068A	T921	1681
T904	0684	T911	068B	T922	1682
T905	0685	T912	068C	T923	1683
T906	0686	T913	068D		

Determining whether a tape drive is attached to UB-MTS: if you are unsure about whether a tape drive is attached to UB-MTS or not, issue the following command at the VM operator's console:

```
QUERY 0680-068F 1680-168F
```

Some possible responses are ("nnnn" indicates the device number):

```
TAPE nnnn OFFLINE

TAPE nnnn FREE

TAPE nnnn ATTACHED TO UB-MTS nnnn R/W
```

The last message ("TAPE nnnn ATTACHED") indicates that a drive is attached to UB-MTS.

The ATTACH command: To attach a tape drive to UB-MTS, issue the following command at the VM operator's console (where "nnnn" is the device number for the tape drive):

```
ATTACH nnnn TO UB-MTS nnnn
```

The DETACH command: To detach a tape drive from UB-MTS, issue the following command at the VM operator's console (where "nnnn" is the device number for the tape drive):

```
DETACH nnnn FROM UB-MTS
```

Loading UB-MTS

The loading of UB-MTS should only be attempted after loading VM/XA or after shutting UB-MTS down.

To load UB-MTS:

1. Go to the VM/XA operators' console.

2. If you are not sure whether the UB-MTS virtual machine is logged on, type "QUERY UB-MTS" and press ENTER.

   If the UB-MTS virtual machine is not logged on, VM/XA will respond with the message "IB-MTS NOT LOGGED ON".

   If the UB-MTS virtual machine is not logged on, type "AUTOLOG UB-MTS" and press ENTER. Then skip to stop 13.

   If the UB-MTS virtual machine is logged on, proceed with step 3.

3. If the UB-MTS virtual machine is already logged on, do the following: (or from VM/XA operators' console, type "force ub-mts", then "autolog ub-mts").

4. Go to a terminal connected to VM/XA.

5. At the LOGON prompt, type "UB-MTS" and press ENTER to logon to the UB-MTS virtual machine console for the UB-MTS virtual machine.

6. The terminal should respond with the message

   "Enter Password".

7. Type the password and press ENTER.

   If you don't know the password, ask your shift supervisor.

8. The terminal should respond with the message "Reconnected at (time)".

9. Hold down the ALT key and press the PA1 key. The word "CP READ" should appear in the lower right corner of the screen.

10. Type "IPL 52C" and press ENTER.

11. Hold down the ALT key and press the PA1 key. The word "CP READ" should appear in the lower right corner of

the screen.

12. Type "DISC" and press ENTER. The message "DISCONNECT AT time" should appear. In a minute or so, the VM/XA logo should appear again.

13. At this point, the message "Do you want to run the current system?' should appear on the UB-MTS operator console.

14. See the section on "Loading MTS" to proceed.

<u>Switching to a backup UB-MTS operator console:</u>

To start the backup UB-MTS operator console, you must signal an external interrupt. To do this:

1. Go to a terminal connnected to VM/XA.

2. At the LOGON prompt, type "UB-MTS" and press ENTER to logon to the UB-MTS virtual machine console for the UB-MTS virtual machine.

3. The terminal should respond with the message "Enter Password".

4. Type the password and press ENTER.

   If you don't know the password, ask your shift supervisor.

5. The terminal should respond with the message "Reconnected at (time)".

6. Hold down the ALT key and press the PA1 key. The word "CP READ" should appear in the lower right corner of the screen.

7. Type "EXTERNAL" and press ENTER.

8. Hold down the ALT key and press the PA1 key. The word "CP READ" should appear in the lower right corner of the screen.

9. Type "DISC" and press ENTER.

10. Hold down the ALT key and press the CLEAR key. The VM logo should appear.

11. Backspace over the word "LOGON", type "DIAL UB-MTS 1F6F", and press ENTER.

12. The backup console should start on the terminal.

••Special notes:

1. If the UB-MTS backup console doesn't appear right away on the screen, you may need to hold down the ALT and press CLEAR.

2. Under certain circumstances, the VM/XA logo may reappear after step 12. In that case, go back to step 10 and try again.

3. When you're done using the backup console (i.e., the regular UB-MTS operator's console is back in service), switch the TEST/NORMAL switch to the right of the screen from NORMAL to TEST and back to NORMAL. In a moment, the VM/XA logo should appear.

4. If you need to reload UB-MTS while the backup console is running, use the TEST/NORMAL switch to get back to the VM logo, and then proceed normally.

Forcing a UB-MTS Superdump

To initiate a UB-MTS superdump, you must signal a restart interrupt.

To do this:

1. Go to a terminal connected to VM/XA.

2. At the LOGON prompt, type "UB-MTS" and press ENTER to logon to the UB-MTS virtual machine console for the UB-MTS virtual machine.

3. The terminal should respond with the message "Enter Password".

4. Type the password and press ENTER.

   If you don't know the password, ask your shift supervisor.

5. The terminal should respond with the message "Reconnected at (time)".

6. Mount a dump tape on a tape drive.

7. Hold down the ALT key and press the PA1 key. The word "CP READ" should appear in the lower right corner of

the screen.

8. Type "ATTACH nnnn * nnnn" where "nnnn" is the device number of the tape drive where the dump tape is located. (See the section "Attaching and Detaching Tape Drives" for a table of the tape drives and device numbers.)

   For example, to take a dump using T901, the command would be:

   ```
   ATTACH 0681 * 0681
   ```

   Note that the device number appears twice in the ATTACH command.

9. Hold down the ALT key and press the PA1 key. The word "CP READ" should appear in the lower right corner of the screen.

10. Type "SYSTEM RESTART" and press ENTER.

11. The UB-MTS operator's console should request a UB-MTS superdump.

12. Hold down the ALT key and press the PA1 key. The word "CP READ" should appear in the lower right corner of the screen.

13. Type "DISC" and press ENTER. The message "DISCONNECT AT time" should appear. In a minute or so, the VM/XA logo should appear again.

Stopping the CPU - UB-MTS-VM/XA

You must do this as the last step in shutting down the UB-MTS system:

1. If you are planning to reload UB-MTS right away, just proceed to the instructions on "Running UB-MTS under VM/XA: Loading MTS".

2. If you are planning to shut VM/XA down or don't plan to run UB-MTS for a while, perform the following steps:

3. Go to the VM/XA operator console.

4. Type "FORCE UB-MTS" and press ENTER.

VM/XA-UB-MTS Attaching and Detaching Tape Drives

When UB-MTS runs under VM/XA, it is very important that no tape drives be attached to UB-MTS except for the ones that UB-MTS is using at the moment. If a tape drive is attached to UB-MTS but in use from the UM-MTS system, it is possible that VM/XA could cause the tape to be rewound unexpectedly; this would obviously cause problems for the UM-MTS user of the tape.

To avoid problems of this sort, it is necessary to issue an explicit ATTACH or DETACH command at the VM/XA operator console to make a tape drive available or unavailable to UB-MTS. These commands are issued automatically by UB-MTS for tapes that are normally mounted; however, the commands must be issued by the operator under other circumstances. For example, it will be necessary to issue explicit ATTACH and DETACH commands in taking a UB-MTS superdump.

VM/XA refers to tape drives by number, rather than by name as does MTS. The following table lists the MTS name and the VM/XA device number for each tape drive; the device number must be specified on the ATTACH and DETACH commands as shown below.

Name	Number	Name	Number	Name	Number	Name	Number
T900	0680	T909	0689	T922	1682	T947	1087
T901	0681	T910	068A	T923	1683	T948	1088
T902	0682	T911	068B	T940	1080	T949	1089
T903	0683	T912	068C	T941	1081	T950	108A
T904	0684	T913	068D	T942	1082	T951	108B
T905	0685	T914	068E	T943	1083	T952	108C
T906	0686	T915	068F	T944	1084	T953	108D
T907	0697	T920	1680	T945	1085	T954	108E
T908	0698	T921	1681	T946	1086	T955	108F

Determining whether a tape drive is attached to UB-MTS: if you are unsure about whether a tape drive is attached to UB-MTS or not, issue the following command at the VM/XA operator console:

```
QUERY 0680-068F 1680-168F
```

Some possible responses are ("nnnn" indicates the device number):

```
TAPE nnnn OFFLINE

TAPE nnnn FREE

TAPE nnnn ATTACHED TO UB-MTS nnnn R/W
```

The last message ("TAPE nnnn ATTACHED") indicates that a drive is attached to UB-MTS. See the section "Controlling Devices" for a description of the meaning of each of these messages.

The ATTACH command: To attach a tape drive to UB-MTS, issue the following command at the VM/XA operator console (where "nnnn" is the device number for the tape drive):

```
ATTACH nnnn TO UB-MTS nnnn
```

Note that the device number appears twice in the ATTACH command.

The DETACH command: To detach a tape drive from UB-MTS, issue the following command at the VM/XA operator console (where "nnnn" is the device number for the tape drive):

```
DETACH nnnn FROM UB-MTS
```

Controlling I/O Devices (UB-VM/XA')

When UB-MTS runs under VM/XA, VM/XA has ultimate control over all of the I/O devices attached to UB-MTS. In some cases it is possible for a device to be unavailable to UB-MTS because VM/XA is unable to communicate with it. In such a case, the problem must be resolved at the VM/XA level (from the VM/XA operator console) before UB-MTS commands (such as ONLINE, OFFLINE, UNITS, and so on) from the UB-MTS operator console can be used.

One very common problem is that certain devices (particularly PCPs) may be marked offline to VM/XA when VM/XA is loaded. In this case, the device will not be available to UB-MTS until it is placed online to VM/XA and attached to the UB-MTS virtual machine from the VM/XA operator console.

VM/XA refers to devices by number, rather than by name as does

MTS. A list of device names and numbers is kept in the machine room.

Determining the status of a VM/XA device: Type the command

```
QUERY nnnn or QUERY nnnn-mmmm
```

at the VM/XA operator console to determine whether one or more devices are on or offline to VM/XA. ("nnnn" and "nnnn-mmmm" indicate a device number or a range of device numbers.) Here are the most common responses (where "dev" is a device type ("TAPE", etc.) and "nnnn" indicates the device number).

1. "dev nnnn OFFLINE" - device is offline to VM/XA.

2. "dev nnnn FREE" - device is online to VM/XA, but not in use by any user (including UB-MTS).

3. "dev nnnn ATTACHED UB-MTS nnnn" - device is online and in use by UB-MTS.

Onlining a device: To make a device online to VM/XA, type the following command at the VM/XA operator console:

```
VARY ONLINE nnnn or VARY ONLINE nnnn-mmmm
```

where "nnnn" and "nnnn-mmmm" indicate a device number or range of device numbers. This command can sometimes fail for no apparent reason, so if you get an error message, just try again.

Attaching a device to UB-MTS: Once a device has been varied online to VM/XA, it must be attached to UB-MTS before it can be used. To attach a device, type the following command on the VM/XA operator console:

```
ATTACH nnnn UB-MTS or ATTACH nnnn-mmmm UB-MTS
```

where "nnnn" and "nnnn-mmmm" indicates a device number or range of device numbers.

Loading MTS

To start the loading process, you should follow the instructions given in the section "Loading UM-MTS on the 3090" or "Loading UB-MTS under VM/XA".

Once the message "Do you want to run the current system" is displayed on the MTS operator console, proceed as follows:

1. The message "Do you want to run the current system" is

displayed.

2. Normally you should just press ENTER.

3. Several lines of text will be displayed. There will be a short pause while MTS checks all of the I/O devices. If any devices, particularly the UMnet PCPs are hung, this pause may take up to 5-10 minutes.

4. If you are running "native", i.e., loading UM-MTS, eventually the screen will clear and MTS will print the message "Local time is ...." If you are loading UB-MTS, this prompt will be suppressed since the time can only be changed by shutting VM down and reloading. See the sections "Shutting VM/XA Down" and "Loading VM/XA" for information on how to do this.

5. If the time and date are correct, press ENTER and proceed with step 6. If the time or date is incorrect and you are loading UM-MTS (i.e., you are NOT running under VM/XA), then you can reset the clock. To do this, enter the command

```
SET ZONE=zzz DATE=mm/dd/yy TIME=hh/mm/ss
```

and press ENTER.

"zzz" should be the time zone (EST or EDT, in Michigan)

"mm", "dd", and "yy" are the month, date and year

"hh", "mm", and "ss" are the hour, minute and second

5c. UM-MTS will print the message "Enable TOD clock".

5d. Go to the ES3090/300J console for the UM side of the ES3090/300J and press the "alt" key and, while holding it down, the "TOD" key.

5e. UM-MTS should display the new time and date; if it's OK, press ENTER; if not, go back to step 5b.

5f. UM-MTS may tell you to enable the TOD clock a second time; if so, repeat step 5d.

6. At this point the MTS operator console and printer should start.

Note: Looking at the UM ES3090/300J console, in the lower left section of the screen, see the "Command" word. Just below this is displayed either UM or MVS. Be sure that you are IPLing the correct system. To change the systems-

UB-MTS UNDER VM

```
     setlp um         (for the UM/MTS system)

     setlp mvs          (for the MVS system)
```

Running VM/XA on the 3090

VM/XA is an IBM operating system which runs on the 3090. At Michigan, we run UM-MTS on one half of the ES3090/300J and VM/XA on the other half. UB-MTS and DOS/VSE runs under control of VM/XA.

VM/XA must be loaded and running before UB-MTS can be loaded. UB-MTS must be shut down before VM/XA can be shut down.

At the present, there are three things that operators will need to do to run VM/XA:

1. Monitor the VM/XA operator console

2. Load VM/XA

3. Shut VM/XA down

Monitoring the VM/XA operator console: Messages are written to the VM/XA operator console one line at a time. When the screen is full, the word "HOLDING" is printed at the lower right-hand corner, and no more information is displayed until the operator holds down the "ALT" key and presses "PA2". It is important to check the VM/XA operator console every few minutes to make sure that no backlog of messages builds up. Some may be important.

In general, important error messages on the VM/XA operator console will be highlighted. In any case, the operator should try to keep an eye on the VM/XA operator console and look for messages that may indicate a potential problem.

The VM/XA operator console is an area in which we do not have much experience; all we can do is try to build up some experience as we go. This writeup will be updated to describe common VM/XA problems as soon as we have run VM/XA for long enough to discover what kind of problems commonly occur.

There are two binders on top of the 3370 disks in the first floor machine room which contain three VM/XA manuals which contain information on running VM/XA:

Real System Operation - Contains information on reloading, shutting down, and controlling VM/XA.

Virtual System Operation - Contains information on running a guest operating system. (This information will be of more use to systems programmers).

System Messages and Codes - Describes all VM/XA messages.

Vector Processor

To check the status of the Vector Processor on the VM/UB side-

Use the VM 3278 and type "q vec"

Response should be "Vector facility 03 online"

If you must put the Vector Processor online, type-

```
vary online vec 03
```

To check the Processor status on the VM/UB side of the 3090, use the VM (3278) console and type-

```
q proc
```

Response should be

```
          PROC 04 MASTER

          PROC 03 DEDICATED TO UB-MTS - CPU 02

          PROC 05 DEDICATED TO UB-MTS - CPU 01
```

Loading VM/XA

This should only be done after VM/XA has been shut down or after the ES3090/300J has been powered off and back on again.

1. Go to the ES3090/300J console for the UB side of the 3090.

2. Press the INDEX key

3. Select the OPRCTL frame by typing 9 and pressing ENTER

4. Make sure that the load address on the OPRCTL frame is correct. If it is not:

   a. Type L1 and press ENTER. Your cursor will move to the LOAD UNIT ADDR field.

   b. Enter the load address and press ENTER.

      The load address should be indicated by a note near the MTS operator consoles.

5. Start the LOAD CLEAR function by typing "O7" and pressing ENTER. (Note that the command requires the letter "O" (oh) and not the digit "0" (zero).)

6. After a moment, the message "Load Complete" should appear on the ES3090/300J console.

7. Go to the VM/XA operator's console.

   In a minute or two, the following lines should appear on the VM/XA operator console:

```
VM/XA Systems Facility Release 2
START ((COLD|WARM|FORCE) (DRAIN) (DISABLE) (NODIRECT))
```

8. Normally, you should press the ENTER key here. However, if you are reloading VM/XA following a VM/XA system crash, you will need to type "FORCE". Never type "COLD" without explicit instructions from a systems programmer.

9. If the time-of-day clock is set the following message will be printed:

```
NOW hh:mm:ss tmz weekday mm/dd/yy
CHANGE TOD CLOCK (Y|N)
```

   If the time and date are correct, press ENTER and skip to step 11. If the time or date is incorrect, type "YES" and press ENTER.

10a. VM/XA will print:

```
SET DATE MM/DD/YY
```

   You should type "mm/dd/yy" and press ENTER, where "mm" is the current month, "dd" is the current day, and "yy" is the current year.

10b. VM/XA will print:

```
SET TIME HH:MM:SS
```

Type "hh:mm:ss" and press ENTER, where "hh" is the current hour (0-23), "mm" is the current minute, and "ss" is the current second.

10c. VM/XA will print:

```
PRESS "TOD ENABLE SET" KEY AT DESIGNATED INSTANT
```

Go to the ES3090/300J console for the UB side of the 3090, hold down the "ALT" key, and press the "TOD" key.

10d. Go back to the VM/XA operator's console.

VM/XA will again prompt:

```
NOW hh:mm:ss tmz weekday mm/dd/yy
CHANGE TOD CLOCK (Y|N)
```

If the time and date are now correct, press ENTER. If not, type "YES" and press ENTER, and go back to step 10a.

10e. In a moment, VM/XA will print:

```
CLOCK SYNCHRONIZATION - PRESS "TOD ENABLE SET" KEY
```

Go to the ES3090/300J console for the UB side of the 3090, hold down the "ALT" key, and press the "TOD" key.

Now go back to the VM/XA operator's console.

11. After the time is set correctly, other messages will be printed on the VM/XA operator's console. Certain errors could occur at this point. Common error messages might include:

```
HCPINI070W MACHINE CHECK DURING SYSTEM INITIALIZATION

HCPxxx320W ALTERNATE PROCESSOR INITIALIZATION
     UNSUCCESSFUL

HCPRET581E RECORDING RETRIEVAL DISK FULL, USER user

HCPCRC583I RECORDING recording name userid USER NOT
     RETRIEVING

HCPxxx625E PROCESSOR nn TOD CLOCK SYNCHRONIZATION
     FAILURE, VARYING OFFLINE
```

If any of these errors occur, or you suspect that any other VM/XA operator console message indicates an error, see the section "VM/XA Error Messages".

12. If no errors are detected, VM/XA will ring the alarm for the UB side of the ES3090/300J from one to three times. Each time it does so, go to the ES3090/300J console for the UB side of the ES3090/300J and press the ENTER key.

    The ringing of the alarm in this case is a normal part of VM/XA operation and doesn't indicate a problem.

13. At this point, VM/XA is up and running, and you can load UB-MTS and DOS/VSE. To do this, see the sections "Loading UB-MTS Under VM/XA" and "Loading DOS/VSE Under VM/XA". If you entered "force" in step 8 above, type the following command and press "enter".

    ```
    autolog ub-mts
    ```

Note: Do not "enable" anything: VM will automatically enable the correct ports.

VM/XA Error Messages

Every message that is printed on the VM/XA operator console is identified by a 7-character code. The form of a message code is:

```
   HCPxxxnnns

   where:  "HCP" indicates a message from the VM//XA control
                 program

           "xxx" is a three-character name of the part of VM//XA
                 that sent the message
           "nnn" is the message number

           "s"   is one of the following "severity codes":

                 "A":  Immediate Physical Action
                 "D"   Decision
                 "E"   Error

                 "I"   Information
```

An "immediate physical action" message indicates something that must be done (for example, readying a tape drive). A "decision" message is a prompt or question to which you must enter some answer (for example, a message to which you must respond "YES" or "NO"). An "error" message is one that indicates some definite error, either in a command issued at the VM/XA operator console or inside of VM/XA. An "information" message describes the result of a command or tells you about the status of VM/XA.

A few of the errors that might occur during a VM/XA system reload are described below. For a complete list of VM/XA messages (including suggested operator and programmer actions in each case) see the manual "System Messages and Codes Reference" which is located on top of the 3370 disk drives.

Message: HCPINI070E MACHINE CHECK DURING SYSTEM INITIALIZATION

Response: This message indicates that a hardware error was detected during a VM/XA reload. Try to reload again; if that doesn't work, call a firefighter (who will probably tell you to call IBM for service).

Message: HCPxxx320E ALTERNATE PROCESSOR INITIALIZATION UNSUCCESSFUL

Message: HCPxxx625E PROCESSOR nn TOD CLOCK SYNCHRONIZATION FAILURE, VARYING OFFLINE

Response: These messages indicate that VM/XA was able to reload successfully, but that one of the two ES3090/300J CPUs was left offline. To correct this:

1. Go to the VM/XA operator console

2. Type "VARY ONLINE PROC 04" and press ENTER.

3. VM/XA should ask you to enable the TOD clock.

4. Go to the ES3090/300J console for the UB side of the 3090 and press the TOD key.

5. If any further errors occur, call a firefighter.

Message: HCPRET581E RECORDING RETRIEVAL DISK FULL, USER user

Message: HCPCRC583I RECORDING recording name userid USER NOT RETRIEVING

Response: These messages indicate that VM/XA isn't properly storing system statistics records. You should inform a firefighter that this message has been printed. (This isn't a serious error, so wait if no one's around.)

Shutting VM/XA down

There are two things to keep in mind when scheduling a shutdown of VM/XA. First, VM/XA can not be shut down until after UB-MTS and DOS/VSE are shut down. Therefore, you must always schedule UB-MTS and DOS/VSE shutdowns as well as a VM/XA shutdown. Secondly, there may be users on the VM/XA system. There will probably be Computing Center programmers, but you need to give them some advance notice of the VM/XA shutdown.

To shut VM/XA down, perform the following steps:

1. As soon as the shutdown is scheduled, put in a signon message (called a "logmsg" in VM/XA terms) on VM/XA. To do this, type the following command at the VM/XA operator console:

```
WNG All message message
```

where "message" is the message that you want to be printed.

For example:

```
WNG All message VM/XA shutdown at 6pm, back up at 7
```

2. As time approaches the scheduled shutdown, send broadcasts ("warnings" in VM/XA terms) to the users signed on to VM/XA. To do this, type the following command at the VM/XA operator console:

   ```
   WNG All message message
   ```

   where "message" is the message you want to send.

   For example:

   ```
   WNG ALL VM/XA shutdown at 6pm, back up at 7
   ```

   You should send warnings at about the same frequency as you would send broadcasts for a UB-MTS shutdown.

3. At any time, you can display the number of users signed on to VM/XA by typing:

   ```
   QUERY USERS
   ```

   on the VM/XA operator console. You can display the userids signed on by typing:

   ```
   QUERY NAMES
   ```

4. DO NOT PROCEED ANY FURTHER UNTIL UB-MTS AND DOS/VSE HAVE BEEN SHUT DOWN.

5. Clear the LOGMSG you put in earlier by typing:

   ```
   WNG All message NULL
   ```

   on the VM/XA operator console.

6. To actually shut VM/XA down, type:

   ```
   SHUTDOWN
   ```

   on the VM/XA operator console.

   VM/XA will ring the ES3090/300J alarm once or twice during the shutdown process. Each time it does so, go to the ES3090/300J console for the UB side of the ES3090/300J and press ENTER.

After the alarm has rung a second time, the message "SYSTEM SHUTDOWN COMPLETE" should be displayed on the VM/XA operator console.

7. VM/XA is down at this point, and you can reload or power the ES3090/300J off.

VSE on the ES3090/300J (Running of)

Running VSE on the 3090

VSE is an IBM operating system which runs on the 3090. At Michigan, VSE runs under control of VM/XA, on the same half of the machine as UB-MTS.

VM/XA must be loaded and running before VSE can be loaded. VSE must be shut down before VM/XA can be shut down. Currently, we run two VSE machines - VSE1 and VSE2.

Note: Currently, we have promised the library to keep VSE available Monday through Friday from 7 AM to 4 AM the following morning, so we need to avoid scheduling reloads during that time period.

VSE tape mount requests have priority over all UM and UB mount requests. A VSE mount request should be given the next available tape drive, regardless of the UM and UB tape mount queues.

There are three things that operators need to do to run VSE:

1. Load VSE1 and VSE2

2. Shut VSE1 and VSE2 down

3. Mount Tapes

Loading VSE1 and VSE2

This should only be done after loading VM/XA or after shutting VSE1 and VSE2 down.

1. Go to the VSE1 operator's console (a 3278 with the label "VSE1 CONSOLE" on the front). The VM/XA logo should be displayed.

2. At the LOGON prompt, type "VSE1" and press ENTER.

3. The terminal should respond with the message "Enter Password."

4. Type the password and press ENTER. (If you don't know the password, ask your shift supervisor.)

5. The terminal should respond with the message "LOGON AT

(time)" and the word "MORE..." should be displayed in the lower right part of the terminal.

6. Hold down the ALT key and press the PA2 (field mark) key. The screen should clear and then begin displaying messages as VSE1 loads.

7. After a few messages are printed on the console, VSE1 will display the message:

```
0I80D LOCK FILE ON 537: DUPLICATE NAME ON VOLUME
     DOS.LOCK.FILE


     0D07D ENTER RESPONSE
```

   Type "DELETE" at this point and press the ENTER key.

8. Shortly after the "LOCK FILE" message, you MAY see a message like:

```
0I37D LABEL AREA ON 422: OVERLAP ON UNEXPIRED FILE
     DOS.LABEL.FILE.FF3712723090.AREA1
```

   If this message is displayed, VSE will stop loading (i.e., nothing else will be printed on the VSE console after this message) until you reply.

   Type "DELETE" and press the ENTER key.

   This message, if it appears at all, will be printed in the first minute or so after "LOCK FILE" message. If you don't see it in the first minute or so, it's safe to assume that it won't appear.

9. Regardless of whether or not the "LABEL AREA" message appears, after a few minutes you should see the message:

```
F1 001 1Q12I VSE/POWER INITIATION COMPLETED FOR
     SYSID=1
```

   At this point, you can go to the VSE2 console and start loading VSE2. VSE2 is loaded the same way as VSE1, except that you logon to VSE2 instead of VSE1, and you should never get a "LOCK FILE" message while you are loading VSE2. You MAY get a "LABEL AREA" message, as with VSE1, within the first few minutes of loading. If you do get a "LABEL AREA" message, you reply by typing "DELETE" and pressing the ENTER key.

10. After the "POWER INITIATION COMPLETED" messages, there

is nothing else to do as part of loading VSE1 and VSE2. A large number of additional messages will be printed by both VSE's. The final messages should be:

```
F4 004 DFH1500 - CONTROL IS BEING GIVEN TO CICS
```

This should appear on each VSE. It takes 5 or 10 minutes for VSE1 and VSE2 to load. The "F4" pertains to Mirlyn - the Library's on-line catalog system. On VSE1, you may see the message:

```
F2 025 DFH1500 - CONTROL IS BEING GIVEN TO CICS
```

The number immediately following the "F2" may not always be "025" as in the example. But the "F2" means that the part of VSE1 used by programmers is up - it does not mean that Mirlyn is up. If you miss a particular message on the console, you can look back by typing the letter "D" and then hitting ENTER. Each time you hit ENTER, the previous screen of console messages will be displayed. You can get out of "console redisplay mode" by typing "D E" and then hitting ENTER.

To check the status of VSE1 or VSE2 from the network terminal; at the "which host" prompt, type-

```
hUbc1       (VSE1 status) or

hUbc6       (VSE2 status)
```

Shutting VSE2 Down

Currently we have promised the library to keep VSE available from 7 AM to 4 AM weekdays, so we need to avoid scheduling VSE reloads during that time.

1. Go to the VSE2 operator's console.

2. Type "MSG F4" and press ENTER.

3. VSE should reply:

```
MSG F4AR 015 1I40I READY
     F4-004
```

4. The "004" part of "F4-004" is the reply ID. Occasionally, instead of displaying "F4-004" (or whatever) in a new message, VSE will simply highlight

the "F4-004" if it already appears somewhere on the screen. So if you don't see a message like the one in the example, look for a highlighted message. Once you know what the reply ID is, type the following command, substituting the reply ID in place of the "4":

```
4 CMSG MSG='MIRLYN GOING DOWN IN x
MINUTES',ROUTE=ALL,SEND
```

and press ENTER. After waiting "x" minutes, type the following, substituting the reply ID in place of the "4":

```
4 CEMT P SHUT,I
```

and press ENTER. (Sending the message is desirable, but not absolutely necessary.)

5. Type "Z NET,QUICK" and press ENTER.

6. The shutdown will take a minute or two to complete. (While the shutdown of VSE2 is in progress, you can begin the shutdown of VSE1.) Eventually VSE2 will print something like:

```
F3 018 EB02I VTAM IS NOW INACTIVE

F3 003 EOJ VTAMSTART

       DATE (...) TIME (....)

F1 001 1Q34I F3 WAITING FOR WORK
```

7. If 5 minutes or so have gone by, and you still don't see the messages indicating the end of VTAM, you have to cancel it. To do this, type "CANCEL F3" and then hit ENTER. VSE will respond with a message something like:

```
AR+015 1I55D CANCEL VTAM ?  REPLY YES OR NO
```

In this example, the "015" is the reply ID. Answer the prompt by typing:

```
15 YES
```

substituting the actual reply ID for "15". VTAM should then terminate promptly.

8. Once VTAM has stopped, type "PEND" and press ENTER.

9. A few more messages will be printed as VSE finishes shutting down.

11. Hold down the ALT key and press the PA1 (DUP) key. This will put you in "CP READ" mode.

12. Type "LOGOFF" and press ENTER.

At this point, VSE2 is completely shut down.

Shutting VSE1 Down

Note: VSE2 must be shutdown before VSE1.

1. Go to the VSE1 operator's console.

2. Type "/TC" and press ENTER.

3. VSE should respond with something like:

```
F2 022 K103I ENTER TERMINAL CONTROL OPERATOR COMMAND
     F2-025
```

4. The "025" part of "F2-025" is known as the "reply ID". (It won't always be 25.) Type the following command, substituting the reply ID in place of the "25":

```
25 CEMT P SHUT,I
```

and press ENTER.

MSG F4

5. At this point, the instructions for shutting down VSE1 are exactly the same as for shutting down VSE2, starting with point number 2.

After VSE1 and VSE2 are completely shut down, you can reload or shut VM/XA down.

Mounting Tapes (VSE1)

All tape use under VSE1 will be done from batch jobs. VSE1 can only run one batch job at a time, so there will never be more than one VSE1 tape user at a time (although that one user may mount more than one tape at a time).

For a while, at least, VSE1 users will call before submitting batch jobs that use tapes and leave their name and phone number. If you run into any problems, call the user back and describe the problem (in addition to taking the steps detailed below).

Since VSE1 runs under control of VM/XA, you must issue VM/XA ATTACH commands from the VM/XA operator's console before replying to VSE1 mount requests. (This is just like mounting tapes for UB-MTS, except that UB-MTS issues the ATTACH command automatically in most cases; under VSE, the ATTACH commands must always be issued by an operator.)

VSE1 mount commands are not really generated by VSE; instead, they are comment lines which the VSE1 programmer puts into the VSE1 batch job. Since each "mount request" is just words typed by a VSE1 programmer, the exact format of each mount request may vary slightly, although the overall request should contain the same information as an MTS mount request.

VSE1 refers to tape drives by number; VSE1 tape drives are numbered 680, 681, 682, and so on. A VSE1 mount request will request that a tape be mounted on a particular drive number; however, the VM ATTACH command can be used to attach any real tape drive to any VSE1 tape drive number, so that any tape drive can be used to satisfy any VSE1 mount request.

A single VSE1 mount request may request more than one tape; all of the tape drives must be ATTACHed, and all of the tapes must be mounted on the drives, before you reply to the VSE1 mount request.

The following steps describe the VSE1 mount process in detail:

1. VSE1 will display a mount request on the VSE1 operator's console in a format similar to:

   ```
   BG 000 *

   BG 000 * PLEASE MOUNT S1234 'NOTIS DUMP A' (RING=IN)
        on 680

   BG 000 // PAUSE AND S5678 'LIBRARY DATA' (RING=OUT) on
        681

   BG-000
   ```

   The last line or two will be highlighted.

   The number at the end of the line on each mount request is called the virtual device number or "vdev". You will need to remember which tape is to be mounted on which "vdev".

2. Go and get the tapes from the racks and check the labels just as if you were mounting MTS tapes. If the rack numbers and labels are OK, skip to step 4. Otherwise, proceed with step 3.

3. If there is any error in the rack number, external

label, or anything else in the mount request, kill the batch job by performing the following:

Type "0 CANCEL" and press ENTER. (That's a zero, not an "oh".)

VSE1 should type:

```
0 CANCELBG 000 1I20I JOB name CANCELLED DUE TO
     OPERATOR INTERVENTION

BG 000 1I00D READY FOR COMMUNICATIONS.

BG-000
```

Type "0" and press ENTER. (That's a zero, not an "oh".)

VSE1 will print a few lines and kill the batch job.

Tape mount processing for this job is complete; at this point, just wait for the next job.

4. If the mount request makes sense and the rack numbers and external labels match, continue processing the mount request.

5. First, determine how many tape drives will be needed. Wait until enough drives are available. If there aren't enough drives available, stop mounting tapes on UM and UB until you have enough drives.

   Mount the tapes on the drives and ready the drives. Keep track of which tape has been placed on which drive.

6. Go to the VM/XA operator's console. For each tape mounted, issue an attach command:

   ```
   ATTACH rdev TO VSE1 vdev
   ```

   where "rdev" is the number of the real tape drive on which the tape was mounted (see the section "Running UB-MTS under VM/XA: Attaching and Detaching Tape Drives" for the numbers corresponding to each real tape drive) and "vdev" is the number of the tape drive requested by the VSE1 mount request.

   It is important to make sure that each tape is ATTACHed to the same "vdev" as indicated in the mount request on the VSE console.

7. Go to the VSE1 operator's console and type "0" and press ENTER. (That's a zero, not an "oh".)

8. The VSE1 batch job should resume operation and use the tapes.

9. When VSE1 rewinds and unloads a tape, go to the VM/XA operator's console and issue a detach command to remove the tape drive from VSE:

   DETACH rdev FROM VSE1 where "rdev" is the number of the real tape drive.

Loading MTS

To start the loading process, you should follow the instructions given in the section "Loading MTS on the 3090" or "Loading MTS under VM".

Once the message "Do you want to run the current system" is displayed on the console, proceed as follows:

1) The message "Do you want to run the current system" is displayed.

2) Normally you should just press ENTER.

3) Several lines of text will be displayed. There will be a short pause while MTS checks all of the I/O devices. If any devices, particularly the UMnet PCPs are hung, this pause may take up to 5-10 minutes.

4) Eventually the screen will clear and MTS will print the message "Local time is ...".

5) If the time and date are correct, press ENTER and proceed with step 6.

   a) If the time or date is incorrect and you are loading UB-MTS (i.e., you are running under VM), then the only way to reset the time is to shut VM down and reload it. See the sections "Shutting VM Down" and "Loading VM" for information on how to do this.

   b) If the time or date is incorrect and you are loading UM-MTS (i.e., you are not running under VM), then you can reset the clock. To do this, enter the command

   ```
   SET ZONE=zzz DATE=mm/dd/yy TIME=hh/mm/ss
   ```

   and press ENTER.

   "zzz" should be the time zone (EST or EDT, at UM)

   "mm", "dd", and "yy" are the month, date and year,

   "hh", "mm", and "ss" are the hour, minute and second

   c) The system will print the message "Enable TOD clock".

   d) Go to the console for the UM side of the ES3090/300J and press the "alt" key and, while holding it down, the "TOD" key.

e) The system should display the new time and date; if it's OK, press ENTER; if not, go back to step 5b.

f) The system may tell you to enable the TOD clock a second time; if so, repeat step 5d.

At this point the operator's console should start.

6) MTS will print the new date and time as it did in step 8. If everything is all right, press "enter". Otherwise, type "no" and go back to step 9.

7) At this point, the OPERATOR job should start up and you can proceed with system startup.

HR	MIN	AM/PM
0-12	0-59	{AM|PM}

MO	DAY	YR(Optional)
1-12	1-31	(last two digits of year)

Time-Zone

Either a positive or negative number, giving the number of minutes relative to GMT or one of the following codes:

EST-Eastern Standard Time
EDT-Eastern Daylight Time
CST-Central Standard Time
CDT-Central Daylight Time
MST-Mountain Standard Time
MDT-Mountain Daylight Time
PST-Pacific Standard Time
BST-British Summer Time
GMT-Greenwich Mean Time
ADT-Atlantic Daylight Time
AST-Atlantic Standard Time

EXAMPLE IPL:

Enter "MTSIPL" or "MTSIPL xxx" where xxx is the plug address of MTS001.

```
DO YOU WISH TO RUN THE CURRENT SYSTEM (YES OR NO)?

   yes
      **UMMPS/370 mm-dd-yy

  LOCAL TIME: HH:MM:SS mm-dd-yy; GMT: HH:MM:SS MM-DD-YY.  OK?
```

The operator may then type-

```
   ok, yes, or a blank line to accept the time as given-
   No - to reject the time given-
   Set time=hh:mm:ssapzzz date=mm/dd/yy zone=zzz

   Where hh is the hour (00-23 or 00-11)
         mm is the minutes (00-59)
         ss is seconds (00-59)
         ap is am or pm (if omitted, 24-hour time is assumed)
         zzz is the time zone (as specified earlier)
         mmm is month (1-12 or Jan, Feb, etc.)
         dd is day (1-31)
         yy is year (0-99)
```

HELP - to get information on the above commands

2. VERIFY TIME/DATE THEN ENABLE TOD CLOCK

After this is done, the request for verification of the time and date will be repeated.

(Note: 1. and 2. usually aren't necessary and are skipped.)

The PN, QN, TN, P11, T11, G11, SIGNONM, OFFLINE, ONLINE and UNITS tasks may be run anytime after the message about the date and time being set. Not until the POST-IPL SYSTEM LOADED OK message should you initiate MTS or HASP tasks, however. When the IPL procedure is complete, the machine will enter the wait state with X'FF0' in the right half of the PSW. The system is now up except you have no terminal or batch capability as yet. You may want to load the print buffers now. To do this, type in the chain or train name (there is a corresponding job program) with the printer name as the first parameter and ECHO as an optional second parameter, e.g.,

```
tn ptrx echo
```

The above example is for 1403 type printers. For 3211 type printers, ECHO is not necessary and default is 6 lines per inch. If 8 lines per inch is desired on 3211 type printers, use the parameter 8LPI.

```
p11 ptrx [8lpi]
t11 ptrx [8lpi]
g11 ptrx [8lpi]
```

For more information on Print Buffer loading, look for appropriate heading in the "Contents".

(The next line will appear on the operators' console on the

VSE

second floor.

```
OPERATOR: STARTED AT hh:mm:ss ON dd-mm-yy MODEL xxx SERIAL xxxxxx
OPERATOR: USING DISPLAY DS02 & PRINTER PTRx (3284)
OPERATOR: STARTED AT hh:mm:ss ON dd-mm-yy
OPERATOR: USING DISPLAY DS32 & PRINTER PTRx (3284)
     MTS   0RELOAD AT hh:mm:ss on dd-mm-yy

                      SYSTEM : UL123 WRITTEN AT hh:mm:ss on TUE JUL  12
     MTS    ENTER INITIALS AND REASON FOR RELOADING
(Type your initials, a space and the reason for loading)
     PDP PAG001 ON Dnnn
     MTS   CONTENTS OF IPL FILE "*IPL.0"
     MTS  UL153 DENTRY=45680 PSECT=100008 VIRTUAL=A0000....1432A8
     MTS WRITTEN BY ID W163 AT hh:mm:ss FRI JUL 15/83
     COM 18:10 JUL 14 - CHANGES TO STACK DESCRIPTOR
     COM FORMAT REQUIRED CHANGES
     COM  TO STACK POPPING IN CMDSHELL AND SYGNOF.
     COM CHANGES TO RESET THINGS PROPERLY IF THE USER HITS ATTENTION
     COM AT THE "ENTER PASSWORD" PROMPT.
     COM CHANGES TO TSA TO SNARK IF IT HAS A RATE NUMBER OF 0
     COM (HAPPENS ON NDSK SYSTEMS).
     COM CHANGES RELATING TO THE STACK MAINTAINED BY MACPUSH/MACPOP

     COM THESE MAKE THE KUDGLE EVE KLUDGIER, BUT WILL GO AWAY EVENTUALL
     MTS FILESAVE DIRECTORY UPDATE COMPLETED

  THE ACCOUNTING FILE IS AVAILABLE


                  *** POST-IPL SYSTEM LOADED OK.
```

(When this comment appears, start HASP but don't release execution.

(This latest comment appears with the successful loading of shared virtual memory.

```
       *CMD:   THE CMDSTAT PICKUP PROGRAM HAS BEEN STARTED.
```

This comment may not always appear at this point but will be somewhere near here. It appears after the cleanup procedure for CMDSTAT is finished and *CMD begins. This is to let you know that when you let out the lines, things may stop until *CMD gets started.

```
         MTS   31 RECORDS
```

(Don't let lines out or do an "mts *rms" or "mts *rmd" until *NAL finishes!)

```
00018 07:17:38 MTS  *NAL Complete (There are seven of these)
```

```
   rm

mts *rms (Starts Resource Manager)


mts *rmp (Start Resource Manager Processors)


mts *las


mts *mns
```

(The accounting file must be available, the "POST IPL SYSTEM" must have been loaded, and the number of "records" printed on the console before "execution" can be released.)

```
$release ex
```

SHUTTING DOWN MTS

1) You should attempt to perform each of the steps outlined here, but under some circumstances that won't always be possible. Do the best you can, but always do step 7 unless there is absolutely no alternative.

2) From 30 minutes to several hours before the shutdown put in a signon message warning users of the shutdown. Send broadcasts 60, 30, 10, 5, and 2 minutes before the shutdown to repeat the warning. Update the recorded status message to include information on when MTS will shutdown and when you expect it to be back up.

3) Five minutes before the shutdown hold batch execution ($HOLD EX) and run the SHUTDOWN job without any parameters. Change the status recording and turn on the flashing green light. Run *DRN and drain the local printers.

3.1) mts *rmd (Shuts down Resource Manager)

4) If members of the systems programming staff are working to finish last minute changes before the shutdown, check with them before going ahead with the rest of this step. At the scheduled shutdown time broadcast one last warning, $RERUN any batch jobs that haven't finished execution, wait from 30 seconds to one minute and then run SHUTDOWN ALL. Turn on the solid

red light and change the recorded status message.

5) After the message SHUTDOWN ALL COMPLETE is printed at the console run *SDB or *SDM, depending on the system, (MTS *SDM,*SDB) to place shutdown messages in the PCPs. This procedure is used to enter a one-line UMnet/UMnet message which will be printed for any user attemting to access which ever system is down. It will not disconnect any users currently connected to the UB or UM systems.

Note: If one or more terminals is still active, "shutdown won't finish. See following text.

```
00002 23:45:49 OPERATOR /T M
...  ...  ...  ...  ...  ...  ...  ...  ...  ...  ...  ...
...  ...  ...  ...  ...  ...  ...  ...  ...
00002 23:45:50 OPERATOR - 00657 MTS 000ABD20 57 I/O on AD01
NLOG WOPN *SLV ; AD01
00002 23:45:50 OPERATOR - 00658 MTS 000ABE80 57 I/O on AE00
NLOG WOPN *SLV ; AE00
00002 23:45:50 OPERATOR - 01492 MTS 000AC140 37 I/O on DS0E
BUSY; DS0E
00002 23:45:50 OPERATOR - 00666 MTS 000AC560 57 I/O on AE01
NLOG WOPN *SLV ; AE01
00002 23:45:50 OPERATOR - 00667 MTS 000AC6C0 57 I/O on AF02
NLOG WOPN *SLV ; AF02
00002 23:45:50 OPERATOR - 00672 MTS 000AC820 57 I/O on AN01
NLOG WOPN *SLV ; AN01
00002 23:45:50 OPERATOR - 00673 MTS 000AC980 57 I/O on AE02
NLOG WOPN *SLV ; AE02
00002 23:45:50 OPERATOR - 00674 MTS 000ACAE0 57 I/O on AE03
NLOG WOPN *SLV ; AE03
00002 23:45:50 OPERATOR - 00675 MTS 000ACC40 57 I/O on AE04
NLOG WOPN *SLV ; AE04
00002 23:45:54 OPERATOR - 02824 MTS 000AE240 141 I/O on AN0A
POST WMTS *NETM; AN0A
00002 23:45:55 OPERATOR - 01408 MTS 000AE920 130 I/O Wait.
HOST WMTS *FTPS; AU42
...  ...  ...  ...  ...  ...  ...  ...  ...  ...  ...  ...
...  ...  ...  ...  ...  ...  ...  ...  ...  ...  ...
23:48:32 07858..STOP 1408
00002 23:48:37 OPERATOR /T 1408
00002 23:48:38 OPERATOR - Specified job(s) not found
23:48:48 07859..SHUTDOWN ALL
07859 23:48:48 SHUTDOWN The NETP MM job reports that 1 ports
were affected.
00002 23:50:19 OPERATOR /T 7859
00002 23:50:20 OPERATOR - 07859 SHUTDOWN 000773A0 2 Time Wait
23:50:47 07861..TABLMOD
07859 23:50:48 SHUTDOWN One or more batch or terminal tasks
hasn't stopped.  SHUTDOWN is NOT complete.
07859 23:50:48 SHUTDOWN Reissue the "SHUTDOWN ALL" after the
tasks have terminated or been STOPped or RERUN.
```

```
00002 23:50:59 OPERATOR /T 7859
00002 23:51:00 OPERATOR - Specified job(s) not found
07861 23:50:48 TABLMOD..*** Cancelled ***
      23:51:56 07862..BLAST 1492
01492 23:51:56 MTS Stopped: BLST 98.024 14.626
      23:52:03 07863..SHUTDOWN ALL
07863 23:52:04 SHUTDOWN The NETP MM job reports that 1 ports
were affected.
```

6) After running *SDM use the command "/T M" to see if all MTS jobs have terminated (you can ignore any jobs that don't have a valid ccid or that have the word BUSY). Take job dumps of jobs that won't STOP (and log them using *DMP) or check with a systems programmer.

   If *SDM cannot be run, see the descrition of the "SHUTDOWN" job program.

7) NEVER SKIP THIS STEP. If you are shutting UM-MTS down (i.e., if MTS is running on the 3090), go to the ES3090/300J console for UM and press the STOP key. Wait until all of the CPUs on the UM side of the machine have stopped before proceeding.

   If you are shutting UB-MTS down (i.e., if MTS is running under control of VM), go to the VM operators' console and issue the command "FORCE UB-MTS". VM should respond "COMMAND COMPLETE".

   At this point,MTS is down, and you could reload.

8) Wait for the CPU status to show STOP or WAIT STOP. If after 15 seconds or so the machine still hasn't stopped, make a note of the fact for IBM and go ahead.

   At this point, MTS is down and you could IPL. If you need to power the ES3090/300J off, go on to step 9.

EMERGENCY POWER-OFF MEASURES

To be re-written.

VIRTUAL MEMORY

The INTEL 3805 and 3825 are used
by MTS as DIRECT ACCESS STORAGE
devices for paging.

After IPL, only the PDP, Paging Drum Processor, uses the 3805s and 3825s and any problems releated to these devices will therefore affect the PDP.

NOTE: See the sections on PGET and PDROP for adding or removing virtual paging devices.

At present we have four virtual paging devices. These include two 3805s and two 3825s. One of the 3825s is in a 3805 cabinet with two motor generators.

Additional information can be obtained from manuals put out by INTEL. These manuals are located in book shelves near the INTEL equipment.

NOTE: Each Intel box has its own manual. These manuals are on a shelf in the first floor machine room. Each box has a sticker with a serial number which should correspond to a serial number on one of the manuals. At present the numbers are 105, v01319, 117 and 299.

INTEL 3805 OPERATOR ACTIONS

The INTEL 3805

The Intel 3805 can either emulate an IBM 2305 or can be used in NATIVE mode. It runs in NATIVE mode at this installation. This means that it is addressed differently and that is is not formatted in the same way as the STC 4305 when the system is loaded.

Usually, when one looks at the operator panel on the 3805, FBA 370 will be displayed. are changed periodically by INTEL personnel. FBA 370 is the MICROCODE RELEASE number for this machine FB01 & 2. This RELEASE numbers will change as we receive new versions of microcode. All of the numbers displayed on this panel are in hexidecimal.

When keying information into the service panel, do so slowly as there is a pause between registration and display of the information. As you key information into the panel, it will be displayed even before you press enter.

There are four tasks that operators may have to do on this panel. These are-

- COLD starts
- WARM starts
- Function 95 (Confirms what was entered during a COLD start)
- Function 90 (This checks to see how many good partions there are, how many spare partitions, etc.)

=COLD STARTs=

<u>NOTE</u>: Also, refer to the "Utilities and Diagnostics" manual (each box has a different manual) pp. 10-3 for proper COLD START procedures and parameters. These manuals are stored in the first floor machine room.

There is a UNIT RESET button inside the 3805 at the left front bottom. RESETS should only be done when the 3805 is physically offline and not in use by the system.

The following situations call for a COLD START.

Initial installation of the 3805

The 3805 has been powered off longer than the lifetime of non-volatile memory (about 96 hours)

The controller fails and inadvertantly alters non-volatile memory

The 3805 is forced into COLD START by the service panel function key

=COLD START PROCEDURES=

COLD START procedures only take a few minutes so if there is any doubt, do a COLD START.

- Take the 3805 physically offline. If the system is up and using the 3805, a complete shutdown will be required.
- press CLEAR

- This should result in a display of FCA 00 which means that FUNCTION CODE entry mode is enabled

• key in B1

• press ENTER

- This should cause a display of DA1 which is a prompt for a channel A device address

• key in nnnn (the device address for channel A) nnnn is different for each box.

• press the CYCLE key

- This should display DA2, a prompt for channel B device address

• key in 0000 (this means that no address for channel B is desired)

• press ENTER

- DA3 is displayed - a prompt for allocation block size

• key in 1000 (hex number of bytes per block)

• press ENTER

- DA4 is displayed - a prompt for the number of blocks in this allocation

Note: The number, 2B00, in the next line isn't always the same, i.e., it depends on the box. Even on the same box, it may have been changed since the last time it was observed. Check with the manual associated with the box for the correct number.

```
• KEY IN 2B00   (HEX NUMBER OF BLOCKS FOR THE DEVICE)
                (2B64 LEAVES 40 SPARES)
                (2BE4 leaves 20 spares)
                (2C5C or any higher number would leave
               2 or fewer spares and is invalid)
```

• press ENTER

• press CLEAR

- This re-enables FUNCTION CODE ENTRY mode

• key in BF (the function code, in hex, for terminating a COLD START)

• press ENTER

- DEFD should be displayed on the panel indicating that the 3805 is defined and a WARM START is permissable

=WARM STARTs=

NOTE: If the 3805 is enabled, a WARM START will automatically be done if the 3805 is powered OFF and back ON.

An INTEL publication lists the following as reasons to procede with a WARM START

- power loss
- hardware reset
- sevice panel RESET or COLD START
- diagnostic microcode is reloaded
- fatal controller error
- WARM START allocation fails to complete
- the 3805 depletes all eligible spare memory

=WARM START procedures=

- Make sure that the 3805 is physically offline.
- PRESS CLEAR
- key in FE (this is the FUNCTION CODE for performing a WARM START)
- press ENTER

- This should result in a display of ALLOC 0n

At the completion of a WARM START, FBA 200 should be displayed. This takes about 4 minutes.

=Function codes 90 and 95=

There are at least two FUNCTION CODES that should be used by the operations and programming staff to check on the configuratio of the INTEL 3805. Two of the FUNCTION CODES are 90 and 95. Unlike COLD START and WARM START, the FUNCTION CODES may be used while the 3805 is online and in use by the system.

=Function code 95=

The VERSION NUMBER message, FBA nnn, should be showing on the 3805 display panel.

- press CLEAR

- This should display FCD 00 (FUNCTION CODE entry mode)

• key in 95

• press ENTER

- Displays DA1

• key in 0000

• press enter

- Displays FCC 0700 (this shows that 7 parameters were returned)

• press CYCLE

- Displays DD1 00D0 (channel A address)

• press CYCLE

- Displays DD2 0000 (there is no channel B address)

• press CYCLE

- Displays DD3 1B64 (the number of blocks assigned during COLD START) (<u>this is machine dependent</u>)

• press CYCLE

- Displays DD4 0000 (the size of each block)

• press CYCLE

- Displays DD5 1000 (number of blocks per partition)

• press CYCLE

- Displays DD6 0004 (the device number entered on input)

• press CYCLE

- Displays DD7 0000 (logical device number)

• press CYCLE

- Displays DD8 0000 (logical device number)

• press CLEAR

- Displays FCD 00 (FUNCTION CODE entry mode)

• key in 0F

- press ENTER

- Displays FBA nnn

=FUNCTION CODE 90=

FUNCTION CODE 90 should be used at least once per week. One of the puposes of this function is to determine if any bad partitions were thrown out by the 3805 and replaced by spares and how many spare partitions remain.

TO USE CODE 90-

- press CLEAR

- Displays FCD 00 (function code entry mode)

- key in 90

- press ENTER

- Displays FCC 0800 (means 8 parameters were returned)

- press CYCLE

- Displays DD1 0004 (number of basic storage modules)

- press CYCLE

- Displays DD2 nnnn (number of good partitions, 4 pages each)

- press CYCLE

- Displays DD3 nnnn (there are nnnn spare partitions remaining- if this number is less than 20, the 3805 should be taken offline at the next reload and INTEL should be called)

- press CYCLE

- Displays DD4 nnnn (hex number of suspect partitions- if this number is greater than 0, INTEL should be told the next time they come in.)

- press CYCLE

- Displays DD5 nnnn (hex number of unservicable partitians- if this number is greater than 0, INTEL should be told the next time that they are in.)

- press CYCLE

- Displays DD6 nnnn (number of sweeps since last WARM START- number varies)

- press CYCLE

- Displays DD7 nnnn (number of big sweeps- number varies)

- press CYCLE

- Displays DD8 nnnn (data transfers since last sweep-number varies)

- press CLEAR

- Displays FCD00 (FUNCTION CODE entry mode)

- key in 0F

- press ENTER

- This returns the panel display to FCA nnn

NOTE: Following a power failure, the 3805 may not automatically power up because it is not connected to the IBM ES3090/300J POWER DISTRIBUTION UNIT (PDU).

If the FAULT light (on the same side as the EPO knob) is on, a F.E. must be called.

INTEL 3825 - Reseting & Warm Start

The INTEL 3825 operates almost the same as the 3805. Instead of a service panel at the rear, a drawer in the front contains an ASP (Alphanumeric Service Processor). The procedure is given in detail in chapter 10 of the 3825 Utilities and Diagnostic manual or Chapter 5 of the FAST 3825 Operators' manual. The 3825 manual mentions "SLOW START" which is the same as "WARM START" for the 3805. Both terms mean the COLD START info is still valid in non-volatile memory. The 3805 has battery backed up CMOS for these tables while the 3825 has E^2PROM.

Since one of the 3825 has no MOTOR ALTERNATORS to smooth over "power spikes", it is more likely to crash following power failures, etc.

Reseting the 3825

If you should need to reset the INTEL 3825, e.g., after a power failure, this is what you should do-

1) Press the white "alarm reset" button on the 3825's front panel (this just turns off the alarm).

2) Write down the time and date and any number displayed on the status indicator on the front of the 3825.

3) Pull open the drawer on the front of the 3825 that holds the keyboard and write down any error message displayed there.

4) Make sure MTS isn't using the 3825 (this isn't normally a problem, because if MTS is using the 3825 when it dies MTS hangs and must be reIPLed).

5) Flip the "enable/disable" switch on the 3825's front panel to the "disabled" position and wait for the green "ready" light to go out.

6) Flip the "reset" toggle switch for controller 0 (the switch is located in the drawer with the keyboard).

7) Open the small door at the back of the 3825 (a dime will to the trick) and press the black "fault reset" button (this clears the status indicator on the front of the 3825).

8) Flip the "reset" toggle switch for controller 0 again.

9) Type "FCA0" at the keyboard and press return. The word DEFINED should appear.

Note: If the above procedure fails to work after two attempts, power the 3825 off and back on.

Warm (or SLOW) Start

10 Type "FCFE" at the keyboard to request a warm start. The words ALLOC BSM 0 should appear.

11) Wait several minutes and when FBA nn.n is displayed flip the enable/disable toggle swtich to the

enabled position. If all has gone well, the green "ready" light will come on.

12) That is all there is to it if MTS isn't running. If MTS is running, you need to online the 3825 and tell the PDP to start using it. From the operator's console:

```
ONLINE FB0x

PGET FB0x
```

13) Pass the information recorded in steps 2 and 3 on to INTEL.

The "PDP" program is restartable at any time if the same devices are used or only when no MTS jobs running if fewer devices are used.

Restarting PDP

1. Do "/t n pdp" to find the task number of pdp.
   where x is the current unit
   number of a fixed-head-file.

   Example: /t n pdp
2. Stop Task # of PDP (if still active)
3. Enter 'pdp'
   or whatever devices desired;
   and you should be back in business.

Example:

```
00003 22:14.01 pdp error ?, 300000 01
40000000FF00 0000004C 07
00003 22:14.09 pdp ff00 not ready
00107 22:24.47 /t d fhf0
00107 22:24.50 00004   PDP 027000   DORMANT  ; D341 FF00 ff08
00108 22:25.57 stop 4
00004 22:26.02 PDP stopped: KILL 35.905 71.582
00109 22:26.08 PDP
00109 22:26.39 PDP PAG101 on D230
```

NOTE: If a PDP device NOT READY message is received while MTS is running, check the lights on the devices and their control units. If there are no error lights on, and the units are "READY":

Restart PDP.

Error Typeout Formats on operators console

1. CHANNEL LOGOUT

   I/O INBOARD ERROR PATH xxxx

2. Direct Access Error

```
dddd mtsnnn error c ffffff xx sssssssssss aaaaaaaa 00

DEV  VOLUME ERROR  PROGRAM SENSE  SENSE     SEEK   LAST OP
NAME MAY BE CODE    FLAGS  FLAG    DATA   ADDRESS  CODE
     ABSENT SEE ATT
            LIST
```

DEV NAME IS UMMPS NAME OF UNIT (DRM1, D237, ETC.).
VOLUME IS AVAILABLE ONLY FOR MTS PACKS.

PROGRAM FLAGS ARE AS FOLLOWS (IN HEX) (ADD BITS IF NECESSARY)

800000:	2321
400000:	2314
200000:	2301
100000:	RETURN ON FATAL ERROR
080000:	CCW LISTS CONSISTS OF A NON-TRIVIAL SEEK
040000:	RETURN ON TRACK OVERFLOW
020000:	RETURN ON END OF CYLINDER
010000:	RETURN ON NO RECORD FOUND
008000:	OPERATION GOT UNIT CHECK ON SIO
004000:	IMMEDIATE OPERATION
002000:	NOT FIRST ENTRY
001000:	BUS-OUT CHECK OCCURRED
000800:	INTERNAL I/O IN PROGRESS
000400:	READING RECORD 0
000200:	RECOVERING FROM TRACK CONDITION CHECK
000100:	REDOING A STAND ALONE SEEK
000080:	EQUIPMENT CHECK HAS OCCURRED
000040:	DOING A SEEK TO A DIFFERENT ADDRESS (2321)
000020:	LOOKING FOR A HOME ADDRESS ERROR ON 2321

SENSE FLAG VALUES:

00	NO SENSE REQUESTED (PROGRAM ERROR)
01	GOOD SENSE DATA
02	UNIT CHECK ON SENSE COMMAND
03	DEVICE ALL FOULED UP - NO SENSE POSSIBLE
04	CHAINING CHECK
08	INTERFACE CONTROL CHECK
10	CHANNEL CONTROL CHECK
20	CHANNEL DATA CHECK
40	PROTECTION CHECK
80	CHANNEL PROGRAM CHECK

OTHER PROGRAM ERROR:

SENSE DATA

SEE SRL, FEMI, ETC.

SEEK ADDRESS

LAST ADDRESS USED IN A SEEK COMMAND, FORMAT IS CCCCHHHH

OP. CODE

OPERATION CODE IN LAST CCW BEFORE THE UNIT CHECK

VIRTUAL MEMORY-PDP

(DOCUMENTATION FOR THE IPLREADER & FRIENDS)

MTS is loaded from an ordinary sequential file when "IPL" occurs. When the system is "IPL"ed, the following message will appear:

"Do you wish to run the current system? (YES or NO)".

If you reply "YES", the system will automatically be loaded.

If you reply "NO", the "IPL" program will read further commands from the console. You should not give a "NO" reply unless specifically instructed to do so.

An explanation of the "IPL" program follows.

RAMROD

A program called RAMROD (Replacement And Maintenance of Resident Object Decks) will be mentioned in parts of the IPL section of this manual. It is a program which runs under MTS. RAMROD replaces the big system object deck of cards which had to be maintained and sometimes loaded by the systems programmers.

The function of RAMROD is to keep track of one or more systems and their components and to allow the systems programmer to manipulate the systems and components with relative ease.

Introduction-(IPLREADER)

The IPLREADER itself is a stand-alone program which when IPLed will load a system (usually MTS) from an MTS sequential file into memory, and then transfer control to the entry point of the loaded system. (By default, it loads the system from the file named "*IPL.0".)

Constructing a sequential file containing a system to be loaded is a feature of RAMROD. The entire IPLREADER process - from the time the LOAD key is pressed to the time the CONFIG initialization message is printed - usually takes less than 5 seconds.

The IPLREADER itself currently has three components:

(1.) The first component, IPLINIT, is the program which writes the IPL data onto the iplarea of a disk in the first place. It is an MTS program which may be viewed as an extension of DASDI.

(2.) The second component, IPLBOOT, is a stand-alone program which is IPLed from the IPLAREA when the LOAD key is pressed, and which then loads or bootstrap loads the third component.

(3.) The third component, IPLREAD, is the stand-alone program which does all the work. It provides the operator with a repertoire of commands for loading and starting systems, displaying and modifying memory, etc.

All three components are designed to run on a /3090 without modification.

IPLBOOT - the IPL Bootstrap

when the LOAD key is pressed, this program is loaded into real memory starting at address X'8000' (to avoid the PSA logout areas). The first thing that this program does is to read the core image of UMLOAD from the iplarea into real memory starting at location X'1000'. Then it calls UMLOAD to load IPLREAD into the top end of real memory. If everything goes OK, IPLBOOT passes control to IPLREAD.

If something goes wrong, error messages may or may not be printed (depending on whether IPLBOOT can get through to the console). After trying to print the error messages, IPLBOOT goes into wait state with the PSW right half containing error status as follows:

```
    00000001 -> unexpected external interrupt
    00000002 -> unexpected SVC interrupt
    00000003 -> unexpected program interrupt
    00000004 -> machine check interrupt
    00000005 -> unexpected I/O interrupt
    00000006 -> RC > 0 from UMLOAD while loading IPLREAD
    00000007 -> I/O error occurred reading disk
    00000008 -> input record > 256 bytes
                (should not happen)
    00000009 -> input record format error
                (should not happen)
    0000000A -> ran out of memory to load IPLREAD
                (a small machine?)
```

Any error message printed by IPLBOOT contains the name "IPLBOOT" in the text of the message.

Typing "rpsw" to ACS on a 3090 at any time while IPLBOOT is executing, or after it has gone into wait state because of some error causes the entire IPLBOOT process to be repeated.

IPLBOOT attempts to write error messages onto the console at address 3A8 by default. If the halfword at real memory location X'FE0' is nonzero, it will use that address instead. X'FE0' is set to zero initially, but not after a PSW RESTART.

IPLREAD - the Iplreader Itself

This component is loaded by IPLBOOT into the top end of real memory. It is a stand-alone program which provides a small repertoire of commands to allow the operator to interactively specify the name of the MTS sequential file containing the system to be loaded, and to display and modify memory locations before or after loading.

This program begins by writing the following message on the operator's console:

```
   "DO YOU WISH TO RUN THE CURRENT SYSTEM?  (YES OR NO)"
```

A "YES" or a null line entered at this point will be the normal response. Any other response (including CANCEL) invokes command mode. The following commands are currently available:

```
COMMAND:  LOAD
SYNTAX:   LOAD [NAME=FILENAME] [ADDRESS=DISKADDRESS]
PURPOSE:  to load a system from the given MTS sequential
          file.  Both keyword parameters are optional.  If
          "NAME=" is not specified, the current system
          filename ( *IPL.0 ) is assumed.  *IPL.1, 2 and 3
          are succeedingly older versions of the system.
          If "ADDRESS=" is not specified, the address of
          the IPL disk is assumed.  The filename should be
          specified in internal format (ie: colon-less ).
          Execution of a system loaded using the LOAD
          command is achieved using the START command.
EXAMPLE:  LOAD NAME=MTS.TESTSYSTEM ADDRESS=260
EXAMPLE:  L N=*IPL.TEST A=267

COMMAND:  START
SYNTAX:   START
PURPOSE:  to start execution of the previously loaded
          system.
          (Note: entering "YES" in response to the initial
          prompt actually causes a LOAD and a START command
          to be implicitly issued)
EXAMPLE:  S
```

```
COMMAND:  DISPLAY
SYNTAX:   DISPLAY ADDRESS [...ADDRESS] [ON ptr]
PURPOSE:  to display the memory locations specified.  If
          only one address is given, then 4 bytes are
          printed starting with the given address.  By
          specifying two addresses, a multiple of 4 bytes
          of memory may be printed.  By default, the output
          device is assumed to be the console device.  The
          output may be directed to a printer by specifying
          the "ON ptr" parameter (eg: "ON 008").  An
          "address" is one of the following:
             1. a real hexadecimal address
             2. the character string "IPLREAD"
                without the quotes
             3. a hexadecimal virtual address
                 (eg: V(100000) )
             4. a symbolic virtual address
                 (eg: V(DUMJOB) )
             5. any expression involving additions
                and/or subtractions of the above
                 (eg: V(MAXPVN)+2 )
          Addresses of types 3. and 4. are only valid when
          a system is loaded.
EXAMPLE:  DISPLAY 1000...1060
EXAMPLE:  DISPLAY 1000...1060 ON 008
EXAMPLE:  DISP 105C+EF...105C+FF-4
EXAMPLE:  D IPLREAD+1D8...1E8+IPLREAD ON 8
EXAMPLE:  D V(MAXPVN)
EXAMPLE:  D V(DSKTAB)...V(DSKTABE)
EXAMPLE:  D V(100000)...V(100020)
EXAMPLE:  D V(OPERDEVS+4)  is illegal
EXAMPLE:  D V(OPERDEVS)+4  is legal

COMMAND:  MODIFY
SYNTAX:   MODIFY ADDRESS HEX-STRING
PURPOSE:  to modify memory locations starting at the
          address specified.
EXAMPLE:  MODIFY 1000 070007000700,0700
EXAMPLE:  M 9D80+BE 4770
EXAMPLE:  MODIFY IPLREAD+DD0 45E0AE4C, 12FF
EXAMPLE:  MOD V(MAXPVN) 00000001
          (Note: if desired the hex-string may be
          segmented -- with blanks and/or a comma
          appearing between the segments.  However,
          each segment must contain an even number
          of digits.)
```

```
COMMAND:  LPSW
SYNTAX:   LPSW PSW-LEFT-HALF PSW-RIGHT-HALF
PURPOSE:  rather than keying in programs using the rather
          awkward facilities provided by the 168 console,
          programs can be entered using the MODIFY command
          - and then started using the LPSW command.  To
          get back to command mode press PSW RESTART.
EXAMPLE:  LPSW 00000000 00001000
          (Note: both the PSW-left-half and the
          PSW-right-half must be exactly 8 digits)

COMMAND:  CHECK
SYNTAX:   CHECK ON or OFF
PURPOSE:  to enable or disable checks for bad PAT bytes and
          bad DSCB type bytes during VTOC searches.
          Initially, this switch is ON and checking is
          done.  Having CHECK ON is obviously an
          undesirable feature if the VTOC is damaged.
EXAMPLE:  CHECK OFF

COMMAND:  CALC
SYNTAX:   CALC ADDRESS-EXPRESSION
PURPOSE:  same as CALC command in MTS (only simpler)
EXAMPLE:  CALC 1EF560-ECD+10
EXAMPLE:  CALC IPLREAD
EXAMPLE:  CALC V(DSKTABE)-V(DSKTAB)
          (Note: the only valid operators are + and -)
          (Note: this command also functions as a SYM
          command (as in SDS))

COMMAND:  REPLACE
SYNTAX:   REPLACE CSECT-NAME FROM TAPE-ADDRESS
PURPOSE:  to replace a deck in a loaded system with another
          deck (sort of like the RAMROD REPLACE command).
          The usual context in which this command will be
          used is for replacing the TABLES csect in a
          loaded system with a DECKGEN'd TABLES csect
          (starter system context).
EXAMPLE:  REP TABLES# FROM 0B0
          (Note: the FROM keyword must be specified)
```

Abbreviations are accepted for both commands and keywords.

CONSOLE DEVICE SELECTION

The console device at the U of M is located at address 3A8. The IPL reader attempts to do I/O to the device at 3A8 - whether it is a 3270 or a 3215.

If you do not have a device which responds either like a 3066, a 3270, or a 1052 at address 3A8, then this program will go into wait state after attempting to do I/O unsuccessfully on 3A8. This situation can be identified by a PSW right half of 00000600. To define a console in this situation, either cause an attention interrupt from the console by pressing "request" if the console is a 1052 or, by pressing "enter" if it is a 3066 or 3270. If for some reason this doesn't work, then do the following:

(1.) enter the 2-byte console address at real memory location X'FE0'
(2.) press PSW RESTART

Alternatively, pressing the INTERRUPT key at any time will put you into wait state with the PSW right half set to 00000100 (signifying an external interrupt) and then you can do 1. and 2. above.

Pressing PSW RESTART is like pressing a break key, and has the effect of putting this program back into command mode. The console device address will be selected from X'FE0' - and, if that is zero (which it is set to initially), 3A8 will be used.

If something goes awry, a message may or may not be printed. In any event, we go into wait state, and the PSW right half is set to one of the following:

```
00000100 -> external interrupt
00000200 -> unexpected SVC interrupt
00000300 -> unexpected program interrupt
00000400 -> machine check
00000500 -> unexpected I/O interrupt
00000600 -> could not find console
00000700 -> could not find console, look at roc
```

Note the third byte (byte 2) of the PSW right half is the error status area for IPLREAD - and the fourth byte is the error status area for IPLBOOT.

WHAT THE IPL SEQUENCE DID

1). IPL - loads main storage, directly from disk (or from cards or tape).

2). System configures itself, i.e., it finds out what hardware is available, initializes some tables, etc., and asks for the time and date and verifies it.

3). Starts an INIT task which echoes time and date, locates the system catalog on the disk packs, and starts the OPERATOR and DMGR jobs.

4). Starts an MTS task in which user INIT is signed on and runs the programs (commands in INIT:INITCMD) which:
   (a) Checks and verifies the time and date given with respect to the time and date of the last record in the file *STATISTICS.
   (b) Records the time and date, and reason for IPL in the system log file.
   (c) Starts the PDP (Paging Drum Processor)
   (d) Prints out the comments for *IPL.O or the operators' console.
   (e) Starts MTS *CLN
   (f) Starts MTS *FSM to complete any work left unfinished because of a system failure while the file save merge program was running.
   (g) Starts MTS *STI to complete work unfinished because the system failed while *STA was running.
   (h) Starts JOBS MAS
   (i) Starts a separate task (MTS *S2L) to load shared segment 2 components such as BASIC, the shared part of the PL/I Library, APL, the EDITOR, etc.
   (j) Destroys any temporary files which were left if the system was not shut down normally previous to this IPL.
   (k) Fixes all i.d.s which need to be "fixed" as a result of a system failure, SNARK or BLASTed job.

Power Fluctuations and Failures

Occasionally, power will be interrupted by storms, worn power lines, by someone working on power equipment, etc. Some of these interruptions will be so short that they won't interfere with the system's performance since the UPS (Uninterruptable Power Supply) system should take over for up to several minutes. Short power spikes should be handled with no interruption in service if the UPS is working porperly. See the section in this manual concerning the UPS. However, some others will cause varying degrees of damage to the system if the UPS failed and power to the system was lost abruptly. If system power was lost abruptly, service to the users could be unavailable for quite a long period of time, as in the case of a disk being partially written over causing files to be damaged, or, other hardware damage.

First, if power was off for just a short time (less than five minutes, you should restore the AC power to the ES3090/300J in order to allow the cooling fans to run.

When you are certain that the AC is functioning, let it run for at least ten minutes so that the hardware has a chance to cool off. If the power repeatedly goes off, then you may have to start the AC each time.

Next, after a power failure which affected the Air Conditioning, be sure to alert the Air Conditioning people, even if the AC seems to be ok and is running. To do this, call 764-8176 from 7:30 a.m. to 3:30 p.m., Monday through Friday. At any other time, call Security at 763-1131 and ask them to get the AC people. The AC people want to come and check the the entire AC system to be sure that everything is really all right.

If power interruptions occur frequently (every few seconds or minutes) it is better to wait until they are over before IPLing the system, if the system is down, although these short power spikes should be handled by the UPS. You may want to call Security (764-3131) or even in some cases, the Edison company to find out what is causing the power problems and how long they are expected to occur.

Recovery from Power Interruptions

1. On rare occasions, a power failure will cause loss of all power including general building power (no lights except the battery powered emergency ones). This would be the case when the power outage lasted longer than the UPS batterys are capable of functioning. In this case, find out from Security how long the power will be down. Of course when no power is available, users cannot be informed

via a recording, etc., how long they are going to have to wait for system availablility. In some cases, after power comes up, the Air Conditioning won't work and the system hardware will be inoperative or the harware could be inoperative for other reasons. In this case, an appropriate recording should be made.

2. After power has been restored, the lights should come on, if they had been affected.

3. Check the little red neon lights above the power panel doors on the first and second floors to see if power is being fed to these breakers. If either set of lights is off, equipment power has to be reset. To do this, go to right of the doors at the north end of the second floor machine room. Turn the key, next to the red EMERGENCY knobs, to the right, a quarter turn (as far as it will go) and back to it's normal position.

4. It is usually necessary to reset the Air Conditioning system. To do this, you will have to enter the electrical rooms. These rooms are just to the left of the elevator on each floor. In each room is a large Air Conditioning power panel and to the left of each of these panels is a gray box, the bottom of which is about five feet form the floor. On the front of these boxes are two rectangular buttons separated by a white rectangular bar. The top button is green and is marked START. Pressing this button will reset the Air Conditioning on that particular floor. The lower, red button is marked STOP.

5. Make sure that the entire system is up (except for some components that may be offline for some reason or other such as maintenance).

6. After the IBM ES3090/300J has been powered up, you must make sure that all of the attached components are up and functioning.

a. Magnetic Tape equipment-- Make sure that all tape units are powered up. Rewind and unload all loaded tapes. If the tape units aren't powered on, go to the rear of the 3800 Contol Units (CUs) open the doors, and power them up using the button located at the bottom. The tape drives also have POWER ON switches at the lower rear of each unit.

b. DISK Storage-- Often, some of the Disk Drives fail to come up after a power interruption.

Amdahl Disks

Before powering up the Amdahl disks for any reason, open the back panels of each disk module and turn off the green tipped power switches for each disk drive, then proceed with the rest of the powering up sequence. All disk activity should have stopped and either the system should be down or the Amdahl disks should be logically offline before using the green tipped switches. Before powering down the Amdahl disk drives, turn these same green tipped switches off before doing anything else but after all disk activity has stopped.

In order to get the Amdahl disks running after a power failure, open the front and back panels of the disk drive modules. Make sure that all of the circuit breakers on on. If these circuit breakers ever go off for some unknown reason, something may be wrong the the Amdahl disk system itself in which case, an Amdahl F.E. should be notified before going any further. Of course, if the C.B.s went off because of a power failure or some other obvious reason, then it's OK for us to reset them. There is one POWER switch and one ATTN switch for each disk drive. These are located inside the back panels of each module. See the stickers on the inside of the back panel of each module. These stickers indicate which switches go with which disk.

LOCAL/REMOTE Switch (Amdahl Disks)

Upon opening the front panel of the Amdahl Control Unit, one should see a black dial type knob (REMOTE/LOCAL) and just below it, four metal toggle switches. Normally, the only one of these toggle switches which should be in the OFF position (while the system is running) is the upper right one marked RERUN.

The Amdahl disks should be in LOCAL mode so they must be powered up manually. First, make sure that green tipped individual power switches inside the back panels of each disk module, are switched off. In the Control Unit, the upper left toggle switch just under the REMOTE/LOCAL dial switch is the power switch.

If the Amdahl disks are in REMOTE, they should come up when the system is powered up, but again, remember to turn off the individual green switches before powering the system up; then turn the green switches on before the system is loaded.

Control Unit Displays

At the end of the CUs, are a series of switches as well as two displays, one for each channel. Normally, these displays should each show 00. If one or the other isn't displaying 00, open the end panel of the CU and lift the appropriate red tipped toggle switch marked IMPL. However, before IMPLing, all disk activity should have stopped and either the sytem should be down or the Amdahl disks should be logically offline. Each side takes about one minute for an IMPL to complete. If these switches are lifted at approximately the same time, one will wait for the other to complete its IMPL before commencing. Before IMPLing, be sure that either the system isn't running or that the disks are logically offline.

Interrupts-Amdahl disks

If an Amdahl disk is in a situation where it needs to be interrupted (comparable to popping the plug on other disk drives) one must open the back panels of the appropriate disk module and and raise the ATTEN switch. Stickers on the inside of the back panel doors should help in deciding which ATTN switch to use.

Emergency Power Switch

The Emergency power switches are located on the ouside of the front panels of each module. If these switches have been turned off for any reason, the panel door must be opened and the rear of the switch must must be lifted with a screw driver or other appropriate implement in order for it to resume its normal (ON) position.

a. MEMOREX 1270-- Go to the MEMOREX 1270 and make sure that it's powered up.

b. IBM & LEE DATA Terminals-- See if the CUs for the IBM and LEE DATA terminals are powered up. When these CUs are powered up, the micro-code is reset also.

c. IBM 2671--Removed from system.

d. Data Concentrators (DCs) & Primary Communications Processors (PCPs)-- If the Data Concentrators or PCPs were affected, go to the rear of these machines and power them up. They should be left in LOCAL. Next, go to the DC Master Console and type HELLO at each one of the MERIT, PCP or DC postions. If a response is printed, usually something like "High Core Monitor", then you can go to the operators' console and type (assuming that MTS is operating) mts *elo, *nlo, etc., depending on the machine type. If no responce appears, then follow the insructions (parts #1 & #2) under the glass in front of Concentrators CB & CC. The instructions for loading the MERIT machine are attached to its front. After these procedures, go to the operators' console and type mts *nlo, *elo, etc.

e. XEROX 9700-- If possible, it is better to let the DPAs take care of restarting the XEROX 9700 printer if a power failure occurs. There are some instruction books under the wrapping table. The first thing that should be attempted is a ROLLOVER. More than two ROLLOVERS in a row will erase the disk pack on which the printout resides. If the ROLLOVER doesn't work, then the 9700 must be BOOTed. The button to do this is in a lower panel on the Control Unit near the Disk Pack.

If a power surge occurs and 9700 power is still present, press STOP and do a ROLLOVER. If the ROLLOVER doesn't work, do a BOOT.

f. MEMOREX 3403-- If the MEMOREX 3403 line printer loses power, open the side panel and follow the instructions. Next, type "t11 ptr2 echo" at the operators' console or using the system ststus oper command. This not only loads the UNIVERSAL CHARACTER SET (UCS) buffer but also "sets" the carriage tape and echos the contents of the UCS buffer on the printer, so the paper must be alligned at the top of the page before the T11 job is used.

If a power surge occurs that stops the 3403 printer but doesn't affect MTS, $drain the 3403 as soon as possible. Reload the UCS buffer using the T11 job. Next, $start the printer. A "power-off" and "on" may be necessary here.

g. IBM 2501-- If the 2501 card reader fails to come up after a power interruption, open the back panel, turn the LOCAL/REMOTE switch to LOCAL and power the reader on. If the LOCAL/REMOTE switch was in REMOTE, set the switch back to that position.

h. IBM 2520-- If the 2520 punch loses power, open the back panel and place the LOCAL/REMOTE switch in LOCAL, use the POWER ON switch and put the punch back in REMOTE.

i. INTEL 3805 & 3825-- If power has been lost to the INTEL 3805 or 3825, and they are in REMOTE status with the ENABLE switch on, a "warm start" will be done automatically. If either of these boxes was in LOCAL mode, then they must be manually powered up and a manual "warm start" should be executed (see the section entitled "Virtual Memory" earlier in this manual).

The following pages contain descriptions of the UMMPS job programs and supervisor commands. In the PROTOTYPE line of each description, the following conventions are used:

a) The first word is the name of the job program. It is always in upper case in the description, but may be typed at the console in either upper or lower case.

b) Other than the exception in a), upper case letters denote parameters which must be typed as shown (they may be in lower case however).

c) Lower case letters denote generic parameters which must be replaced by an appropriate substitute. For example, the sequence <u>nnnnn</u> denotes a number, the sequence <u>device-i.d.</u> denotes the symbolic name of an I/O device (such symbolic names are system dependent) --- an example of a symbolic I/O device-i.d. would be RDRx or PTRx.

(d) {|} denotes a choice of options, e.g., {x|y} means choose either x or y.

(e) [ ] denotes optional characters or parameters.

BLAST

NAME:	BLAST
FUNCTION:	To forcibly kill a job if STOP will not kill it.

PROTOTYPE: BLAST nnnnn

USAGE: This job forcibly removes a task from the system. It should only be used as a last resort, since it does not signoff the user properly (if it is an MTS task that is to be BLASTed). If a task is BLASTed, any termination exits that the task may have enabled are ignored. If the task to be BLASTed is a terminal task, the operator should wait at least three minutes after he has STOPped the task to allow the task to go through the full signoff procedure. Before BLASTing the task, the operator should know the task number of the task to be BLASTed, the i.d. of the user (if it is an MTS task), and the device i.d. of the terminal (if it is a terminal task). If it is a HASP batch task, the operator should also get the HASP job number associated with this task. He can obtain this information by initiating a TASKS job with the job number of the task to be BLASTed as a parameter. The operator should use this information to restart the line later and to run *FIX on that user i.d. (see the section on "OPERATOR-FILES" for a description of *FIX). The task will be terminated with a code of BLST.

<u>WARNING</u>: Use of the BLAST job on an MTS task that is not actually hung up may cause damage to the file system that will keep the system down for several hours. DO NOT USE BLAST except when absolutely required.

PARAMETER: nnnnn - a task number.

NOTE: <u>Make sure</u> you BLAST the task you desire and not the task of another user (who will become furious!) or a HASPLING task.

EXAMPLE: blast 42

NOTE: The SYSTEM STATUS command /blast nnnnn performs the same function and is especially useful if all job tables are in use.

BROADCST

NAME: BROADCST

WARNING:

FUNCTION: This job allows the operator to- 1) send a message to all terminals attached to the system or- 2) request that only certain terminal tasks receive the message.

PROTOTYPE: BROADCST [ALL|nnnnn ...]

USAGE: With the parameter "all", the message will be sent to all users. If any other parameters are present, they are assumed to be task numbers. After giving this command, the BROADCST job-program will read a line from the console --- this is the line that will be BROADCSTed. Just issuing BROADCST will only result in printing a list of task numbers which did not receive the last message. A message will be sent to only those tasks that were indicated in the parameter list.

If the operator cancels or enters a null input line nothing will be broadcast.

The program MTS.:BROADCST can be used to initiate broadcasts from terminals other than the operators' console.

LIMITATIONS: The message will not be sent to a terminal until the current line at the terminal is terminated (i.e., typed). If the terminal is UMnet, the 1270, or a 3270 and is in read status, the message will not be printed until the line is read. If a message is sent to a 3270 display which is being used to edit a file in "visual" mode, the message will not appear until the user leaves "visual" mode. If a BROADCST is given which overrides a previous BROADCST and a terminal user has not yet received the previous BROADCST message, the message

```
LAST MESSAGE NOT RECEIVED BY nnnnn
```

will be typed where nnnnn is a list of one or more task numbers. At this point a null input line will terminate the broadcast request without affecting the previous broadcast.

If one tries to BROADCST specifically to a non-terminal task or to a terminal task which is not in active use, the message

```
  CAN'T BROADCST TO nnnnn     will be typed.
```

NOTE: The operator should be aware that the carriage width of some terminals is normally limited to seventy characters and any message longer than that may become truncated at the carriage width.

PARAMETERS:

a) all --- send message to all active terminals

b) nnnnn --- where nnnnn is a list of one or more task numbers associated with active terminal tasks.

EXAMPLES:

```
broadcst all

shutdown in five minutes


broadcst 40

please call machine room now!


broadcst 401 403 405

please sign off, we will reload.
```

DIS

MAME DIS

FUNCTION: To display the contents of real-memory locations.

PROTOTYPE: DIS xxxxx[+yyyyy ...] [nn]

USAGE: This command allows the operator to display the contents of real-memory locations. There are really two parameter strings. The first is the address string (which can contain intervening plusses "+" so that one can add addresses and displacements to get the wanted address). The second parameter string is numeric and denotes the number of words (starting at the address given) that are to be displayed on the console. If not given, nn defaults to one (1).

PARAMETERS: 1) A hexadecimal expression representing the address and

2) the number of words to be displayed (starting at the address stated by the first parameter).

EXAMPLES:

```
dis 403a
  00403A 90ECD008

dis 403a+1000 2
  00503A 90ECD000 18FF1B00
```

NOTE: This is not really a JOB request, but a SUPERVISOR command. DIS may not be issued from the 3270 operator's console; the /DREAL command should be used instead. See the section on "Systemstatus".

DMGRSTAT

NAME: DMGRSTAT

FUNCTION: Disk Manipulation

USAGE: DMGRSTAT

NOTE: The DMGRSTAT job does not process parameters but rather reads commands from the console.

ADD

Purpose: To add a disk volume to the disk manager.

Prototype: ADD <pvname>

Parameters:

pvname is the MTS public volume name of the disk to be added.

Description: A check will be made to see if the disk manager already has the disk associated with <pvname>. If so an error message will be generated and the request will be ignored. If the disk manager does not have the disk it will attempt to acquire it via DMVOLGET. If the disk is successfully acquired the disk manager holds onto it and creates a DISKLING and IOMNGR subtask for it. If the disk is not successfully gotten an error message will be printed and the call ignored.

Examples: ADD MTS023
the above command will add MTS023 to the set of disks the disk manager has control over.

BLAST

Purpose: To delete a disk volume from the disk manager without attempting to clean things up.

Prototype: BLAST <pvn> [OK]

Parameters:

pvn is the MTS public volume number of the disk to be deleted from the disk manager.

Description: This command does nothing - it once was intended to have been implemented according to the following description, but was ajudjed to be too dangerous.

If the disk manager does not have the drive specified by <pvn>, an error message will be issued and the command ignored. If the disk manage does have the drive, the user will be asked for confirmation unless "OK" has been specified on the command. If the user confirms deletion, a "blast disk" request will be sent ot the disk manager.

WARNING: the use of this command is quite dangerous and can result in lost requests and inconsistent files. Because no attempt is made to post disk requests which are currently being processed for the disk drive requests may be lost. The lost requests can be recovered only by stopping the disk manager job and restarting it. The disk manager will not wait for an I/O channel program to complete before releasing the volume when a blast request is issued. As a result inconsistent disk data may result if an I/O operation is prematurely terminated.

NODMGR

Purpose: To delete a disk drive from the disk manager.

Prototype: NODMGR <pvname> [OK]

Parameters:

pvname is the MTS public volume name of the disk to be deleted from the disk manager.

Description: If the disk manager does not have the drive specified by <pvname> an error message will be issued and the command ignored. If the disk manager does have the drive the user will be asked for confirmation unless OK has been specified on the command. If the user confirms deletion VOLREL will be called and the disk drive removed from the disk manager. Most requests will be retried successfully using non-disk manager methods.

Examples:

```
NODMGR MTS001
```

the above command will delete MTS001 from the disk manager's control if the user confirms that it is to be done.

```
NODMGR MTS007 OK
```

the above command will delete MTS007 from the disk manager's control without prompting the user as to whether that is ok.

DISPLAY

Purpose: To display various information related to the disk manager.

Prototype: a) DISPLAY CONFIGURATION
b) DISPLAY GLOBAL_COUNTS
c) DISPLAY METRIC_COUNTS

Description: If form (a) is specified the disk drives currently being serviced by the disk manager will be displayed. If form (b) is specified, counts for each volume will be displayed. The statistical counts are of the bulk number of channel programs issued, the number of pages transferred, and the number of pages which were retreived froma and placed into the disk cache.

If form (c) is specified, statistical counts are displayed for each volume. The counts contain information for the Metric system-watching program.

Examples: DISPLAY C
the above command will display the current disk drives under the disk manager's control.

FORGET

Purpose: To get the disk manager to release and reacquire a disk drive thus "forgetting" its location momentarily.

Prototype: FORGET <pvname>

Parameters:

pvname is the MTS public volume name of the disk whose location is to be forgotten momentarily.

Description: The disk manager will check to see that it has the drive specified. If not, an error message will be generated and the call ignored. If it does have the drive the disk manager will wait for any current I/O on the drive to be completed and then will call DMVOLREL to release the drive. The disk manager will then immediately call DMVOLGET to reacquire the drive.

Examples: FORGET MTS006
the above command will cause the disk manager to forget the location of MTS006 and then re-acquire the drive.

NOTE: See the section, "Disk Packs - Moving".

HELP

Purpose: To display information about the use of this program.

Prototype: HELP

Description: Presently the only information displayed will be the prototypes for the various commands.

LSTATUS

Purpose: To display the lock status of a disk page, disk volume or task.

Prototype:

a) LSTATUS PAGE <ea>

b) LSTATUS {TASK} <task>

c) LSTATUS VOLUME <pvn>

Parameters:

<ea> is the external address of the disk page whose locking status is to be displayed. <ea> may be either a quotedhexstringor (<volume>,<page>).

<task> is the task number of the task whose status is to be displayed.

<pvn> is the MTS public volume number whose locking status is to be displayed.

Description: If form (a) is specified the current locking task and the task numbers associated with the requests waiting to lock the disk page whose external address is <ea> are displayed.

If form (b) is specified the external address for each page locked by <task> is displayed along with the external addresses of pages <task>'s requests are waiting to lock.

If form (c) is specified the locking task and waiting tasks are displayed for each locked page on MTS public volume <pvn>. The external address is included in the display of each page.

Examples: LSTATUS '01B00005'
LSTATUS (27,5)
both of the above commands will check the lock status of page 5 on volume 27.

REQUEST

Purpose: Generate a request of disk manager via DISKIO supervisor call.

Prototype: REQUEST <opcd> <ea> [<vma>] [OK]

Parameters:

<opcd>is disk manager operation code in character form as defined in DMGR:DM.MACROS in DRCB copy section.

<ea> is the external address of the page the request is for. <ea> may be either a quoted hex string or (<volume>,<page>). Where <volume> and <page> are integers.

<vma> is a number between 1 and 5 specifying which of the 5 page buffers maintained by this program should be read into or written from. <vma> may be ommitted if it is not needed, but an error message will be issued if it is not present for a request which needs it.

Description: A disk manager request control block is constructed with opcode <opcd>. The external address field of the request control block is filled in with <ea>. The virtual memory address field is filled in with the address of the buffer whose number is <vma>. If <vma> is zero the request is checked to make sure it does not perform I/O operations. The virtual memory address field is then filled in with a zero. After the request has been built it is sent to the disk manager via SVC DISKIO. Upon return from the disk manager the return code for the request executed will be printed.

For requests which involve destructive operations (e.g., writes and zaps) the user will be asked for confirmation if OK has not been specified upon the command.

<u>WARNING</u>: the DISKIO supervisor call is synchronous and the use of the REQUEST command may result in the hanging of this program in diskio wait.

Examples:

REQUEST DRWRITE '0010001A' 2 NORELOCATE the above command will write page 26 of volume 1 from vm buffer 2 with relocation disabled. Because OK was not specified on the end of the command, the user will be prompted for confirmation before the operation is performed.

REQUEST DRLKREAD (2,19) 4
the above command will lock and read page 19 of volume 2 into vm buffer number 4.

REQUEST DRZAPTSK 91
the above command will remove task 91's locks and waits from the locking tables. Note that DRZAPTSK is the only request where the <ea> may be given as a single integer rather than a hex string or (n,m) form.

<u>SIGNAL</u>

Purpose: To generate a signal event for a disk manager subtask or subtasks.

Prototype: SIGNAL <event> [<pvn>]

Parameters:

<event> is a character string specifying the event to be generated. See the description below for a list of the possible events.

<pvn> is the MTS public volume number of the disk the subtask is associated with when the signal is to be sent to a DISKLING or IOMNGR disk manager subtask.

Description: The event specified by <event> will be generated.

<event> may be any of the following:
DISP - generate a signal to the

DISPATCH subtask.

DSKL - generate a signal to a

DISKLING subtask; <pvn> must be specified.

IO - generate a signal to an IOMNGR subtask; <pvn> must be specified.

Examples: SIGNAL dskl 19 the above command will generate a dskl event for the DISKLING subtask associated with volume 19.

STATISTICS

Purpose: To display statistics about the operation of the disk manager.

Prototype: a) STATISTICS <sbt> [<pvn>]

b) STATISTICS OVERVIEW

Parameters:

<sbt> is the type of the subtask

statistics are to be displayed about. <sbt> may be any of the following: DISPATCH, LOCKMNGR, DISKLING, IOMNGR, ITSKMNGR.

<pvn> is the MTS public volume number the subtask is associated with when there is more than one subtask of type <sbt>.

Description: If form (a) is specified statistics about the operation of the subtask specified by <sbt> and <pvn> are displayed.

If form (b) is specified statistics about the operation of the disk manager as a unit are displayed.

STATUS

Purpose: To display the status of a disk drive, task, or the subtasks comprising the disk manager.

Prototype:

a) STATUS DISK <pvn>

b) STATUS SUBTASKS

c) STATUS TASK <task>

Parameters:

pvn is the MTS public volume number of the disk whose status is to be displayed.

task is the task number of the task whose requests are to be displayed.

Description: If form (a) is specified the disk manager will check to see that it has the disk drive for <pvn>. If not, an error message will be issued and the command ignored. If the disk manager does have the drive, the current state of the DISKLING and IOMNGR subtasks for that drive as well as the queues of requests currently being processed for the drive will be displayed.

If form (b) is specified the status of all the subtasks which currently comprise the disk manager will be displayed.

If form (c) is specified the status of all the requests for the task specified by <task> will be displayed.

<u>FLUSH</u> This is a HASP command and is obsolete

NAME: FLUSH

FUNCTION: To clear requests from HASP for I/O operations on a device.

PROTOTYPE: FLUSH device-i.d.

USAGE: This job serves as an infinite sink for output from HASP for a given device. It is used when the device has hung up in such a way that does not allow it to be $DRAINed. The usual sequence of commands to clear this hang-up is:

<u>STOP</u> the HASPLING servicing the device

<u>$RESTART</u> the device

<u>$DRAIN</u> the device

<u>FLUSH</u> the device

NOTE: <u>This sequence of commands should never be used with an SAnn device or RJE. One should use the $STOP SAnn command, wait for a "drained" confirmation and a "kill" coded stop and then use $START SAnn.</u>

<u>NOTE</u>: The order of the above commands is very important. FLUSHing a device which has not been $DRAINed can cause loss of output. This command <u>must</u> be used with extreme care. Before using the FLUSHed device again, make sure that the FLUSH job is not executing anymore. To do this, enter

```
TASKS nnnnn
```

where nnnnn is the task number assigned to the FLUSH job. If the system responds with "JOB NOT FOUND", then the FLUSH job has terminated and the associated device is now available for use by HASP. Otherwise, wait a while and try TASKS again until the "JOB NOT FOUND" message is typed out.

PARAMETER: The device-i.d. of the device that seems to be failing.

EXAMPLE:

```
stop 3 (task 3 has device ptrx )

$restart ptrx

$drain ptrx

flush ptrx
```

GOOSE

NAME: GOOSE

FUNCTION: To simulate an ATTENTION INTERRUPT on the operator's console during an MTS session at the console or to interrupt a batch task.

PROTOTYPE: GOOSE nnnnn

USAGE: MTS sessions being conducted at the operators' console can be interrupted. To get the effect of an attention button, one must use the GOOSE command giving as a parameter the job number which is to get an ATTENTION INTERRUPT. It is also possible to interrupt a batch job with the GOOSE command. In this case, after MTS has fielded the attention interrupt (that the GOOSE caused), it will read a line from the console. The operator may enter one of the following:

$ATT - abandon the current user command and go to the next user command. The attention interrupt that was given will be noted in the user's output.

$DUMP - dump user storage as if the MTS subroutine ERROR had been called and then do as for $NEXT.

$NEXT - stop the current batch job and skip all card reader input until the next S-8 card.

There is usually little reason to use GOOSE with a batch job initiated by HASP, since there are HASP commands available to do the same thing.

GOOSE should cause a reaction within half a minute; however, it will wait until all I/O devices attached to the task are ready. Thus, one should check for any "NOT READY" messages before abandoning GOOSE.

GOOSEing a non-MTS task will cause the task to be stopped with code ATTN if that task did not set itself up to intercept attention interrupts.

PARAMETER: The task number of the task that is to receive the ATTENTION INTERRUPT.

EXAMPLE: goose 125

NOTE: The SYSTEM STATUS command /attn nnnnn performs the same function and is especially useful if all job tables are in use.

INIT

NAME: INIT

FUNCTION: To initiate the system at IPL time.

PROTOTYPE: INIT

USAGE: The INIT job is automatically started when the system is IPLed. It is this job that prints out the message:

TIME AND DATE HAVE BEEN SET TO ...

This job also starts an MTS task with parameters '*INI T,,,'. If you have gone through a normal IPL, there should be no reason for the operator to start the INIT job. If he does, he might run the chance of destroying all scratch files in use (that might be the least of his worries, however).

PARAMETERS: None.

IT

NAME: IT

FUNCTION: To obtain a list of all of the intertask Networks in existence and to sense a particular Network.

The IT program does not process parameters, but rather reads commands directly from the console. Currently, there are only three commands; LIST, SENSE and HELP.

LIST

Function: to obtain a list of the names of all networks in existence. The list command does not take parameters.

SENSE net-name <modifiers>

Function: to sense a particular network. The network name must be specified. A summary of the information in the network control block and one line of information per node is printed. The information in the summary is self explanatory. The information in the node descriptors is largely sef explanatory except, perhaps for the information appearing under the column heading STATUS. This information appears as a series of letters which should be interpreted as follows:

```
D -> means node has been deleted
X -> means node has an exit set
S -> means exit is synchronous
Q -> means  node  has a non-empty queue
     of    messages,    receipts    and
     notifications
B -> means node has been blasted
```

The optional modifiers are:

```
NOHEADER -> to suppress the printing
            of  the  summary  of the
            information   from   the
            network control block

NONODES  -> to suppress the printing
            of  the  individual node
            descriptors
```

HELP Prints list of legal commands.

Example: IT

```
               LIST
               SENSE *DMGRCTL
               HELP
```

NOTE: This program may also be run as an MTS program, i.e., $run mts.:it.

JOBDUMP

NAME: JOBDUMP

FUNCTION: To force a dump (on tape) of all of real core and of all tables and virtual memory associated with a single task.

PROTOTYPE: JOBDUMP nnnnn

USAGE: A jobdump is of use only to systems programmers and should be taken only for a good reason, e.g., if an MTS task SNARKs, if a task hangs up for no apparent reason, or if requested to do so by a systems programmer. All processing stops until all of real core is written out onto tape. Before the JOBDUMP request is given, the operator should mount a dump tape on a nine-track drive (and make sure the ring is in). After the command is given, the message:

```
**DUMP** SPECIFY TAPE DRIVE
```

will be printed on the operators' console. The console will be in read status awaiting a reply. If you wish to actually CANCEL the dump (i.e., you really didn't want a dump), type a "c" followed by an ENTER. Otherwise, type the name of the nine-track tape drive on which you have mounted the dump tape, e.g., t903, followed by an ENTER. If you make a mistake in responding, cancel the line and retype the response.

There are certain instances in which a task will internally request a job dump. The operator should comply with all unsolicited requests (If he has enough dump tapes). Some internally requested job dumps will be followed by a message on the console explaining the reason for the job dump --- these messages will (usually) be typed after the job dump has been taken.

A dump tape is a 9-track labeled tape with volume label of DUMP.

PARAMETER: nnnnn - The task number of the task to be dumped.

```
    EXAMPLE:       jobdump 453

                   ** DUMP** SPECIFY TAPE DRIVE..t903
```

After taking a dump, log it in the dump note book together with the the other information requested in the book.

NOTE: When the dump has finished completely, *DMP should be initiated by the operator. *DMP has to do with printing dump tapes. See the section on *DMP appearing later in this manual.

JOBS

NAME: JOBS

FUNCTION: To start a task that (1) gathers statistics about the current system load and does load levelling on the system using these gathered statistics; or (2) display the statistics being gathered.

PROTOTYPE: JOBS xxxx

USAGE: If the xxxx parameter is not specified, the JOBS program will read commands from the console. Otherwise the xxxx is an initial command.

```
JOBS /LOA
```

prints out the system load statistics on the operator's console every 20 seconds (use GOOSE or STOP jobs to get rid of).

NOTE: See the section on System-Status (below) and MTS Vol. I for a complete list of JOBS commands.

This job is not normally used from the 3270 operator's console; the /... SSRTN (SYSTEM STATUS) commands are used instead.

MOD

NAME: MOD

FUNCTION: To modify a location in real-memory.

PROTOTYPE: MOD xxxxxx[+yyyyyy ...] zzzzzz

USAGE: This command is used to modify a real-memory location or a string of contiguous real-memory locations. The first parameter (i.e., xxxxxx[+yyyyyy ...]) is evaluated as a real memory address. It may be a hexadecimal constant or a hexadecimal expression (i.e., hexadecimal constants separated by plus-signs). The real memory address so evaluated will be the location at which modification begins. The zzzzzz parameter is a variable length hexadecimal string that denotes the new contents --- all commas in this string are ignored and this string is terminated by the first blank or end-of-line (which ever comes first).

PARAMETERS: a) xxxxxx[+yyyyyy ...]-the address expression

b) xxxxxx-the hexadecimal modification

EXAMPLES:

```
mod 30fa 90
mod 30fa+1000 90ec,56
```

NOTES: There is no response (or verification) for this command.

There is no MOD job as such since MOD is a supervisor command. MOD is not a valid command from the 3270 operator's console. Use the /MREAL command (which can be found in the "Systemstatus" section of this manual) instead.

MTS

NAME: MTS

FUNCTION: to start an MTS task --- these are the tasks for which this whole system exists. MTS is a re-entrant job program. It provides the capability of loading, executing, and controlling programs from terminals or through batch streams.

PROTOTYPE: MTS [op-par#1] [op-par#2] msource [, msink] [, syspch]

USAGE: The MTS command takes several parameters. It needs at least one parameter, i.e., msource. This mandatory parameter denotes an I/O device or MTS public file from which input is to come. Preceding the mandatory msource parameter, there may be zero, one, or two optional parameters. Following the mandatory msource parameter there may be zero, one, or two optional parameters. Notice that the optional parameters which follow the msource parameter consist of a single comma (delimited by blanks) and the actual parameter itself (either msink or syspch).

The first group of optional parameters (those preceding msource) consists of one or two of the following: NDSK, FULL, BATCH (really ATCH), MORE, and HASP. However, there are limitations as to the combinations of these optional parameters. The optional parameter FULL has meaning only to terminal tasks and denotes whether or not the task is to always have FULL terminal capability (if MTS finds that the system is rather overloaded, it may allow users to use only limited capabilities of MTS --- e.g., EDITing would be allowed but RUNning of programs would be allowed only if the execution time ≤.25 sec.). If a line has been started with the FULL parameter, it gets full service no matter what the system loading happens to be. The HASP parameter is used internally and should not be used by the operator. The next two parameters may be used together, but if used together they must be used in the order BATCH MORE. If just BATCH is used, MTS will read jobs from msource until a hardware end-of-file occurs on msource. When this end-of-file is sensed, MTS makes a note of it, and when it gets ready to read the next job it will STOP itself --- i.e., the MTS task will go away. If MORE is also used, i.e., BATCH MORE, then MTS will read jobs from the msource device until the MTS task is STOPped by the operator or ungracefully dies.

NDSK specifies that the MTS job is to run without using the disk subsystem. This is useful when installing MTS at a new installation and useless thereafter.

NMSG specifies that MTS should not try to access the system message file for SIGNON messages or Bulletins.

STSK is used internally to start an MTS terminal job as a "subtask" of another job.

The group of parameters following the msource designation are optional. The msink parameter specifies the device that is to be allocated to the master sink. This is where, in MTS, output directed to the terminal would go. The second parameter in this group denotes the I/O device where output directed to the MTS logical I/O unit SPUNCH would go (if SPUNCH were defaulted).

Normally, the operator does not have to worry about either of the optional parameters which precede or follow msource. If msink does not appear, then msink defaults to the msource device (unless the msource device is a file, in which case msink defaults to the MTS pseudo-device *DUMMY* [which is the infinite wastebasket]). If syspch does not appear, then calls to the MTS logical I/O unit SPUNCH are illegal if SPUNCH was not assigned in the user's command stream or program.

The parameters msource, msink, and syspch must be a multiple of 4 characters long (use trailing commas to pad things - see example 7 below) and must be broken into groups of 4 characters by blanks.

NOTE: It should be noted here that when running batch with the BATCH parameter (as opposed to letting HASP supervise the batch function) 1) the unit record I/O units used are directly allocated to the MTS task (and therefore run slowly and inefficiently), and 2) there is just one task number for all the batch jobs executed. This is opposed to HASP, where 1) HASP owns the unit record equipment and makes good use of it and 2) each user job is executed with a different MTS task number.

EXAMPLES:

```
(1) mts la71
(2) mts rdrx , ptrx , pchx
(3) mts batch more rdrx , ptrx , pchx
(4) mts batch rdrx , ptrx
(5) mts *fyl
(6) mts *fyl , oper
(7) mts *fil e2,,
```

The following explanations are in order. (1) This starts a task on device LA71 (probably a terminal) with *msource* and *msink* assigned to that device (*syspch* is not available). (2) This starts a task using device RDRx as *msource*, device PTRx as *msink*, and device *pchx* as *syspch*. Note that this is *not* a batch task (BATCH was not given as a parameter). Thus, the input on *msource* should *not* begin with an S-8 card. Also, only *one* MTS job will run (i.e, when the first $SIGNOFF card is read, the task will disappear quietly (no matter if there are more card images on *msource*). (3) This is an invocation of a batch task that will not stop until the operator STOPs it (or it dies ungracefully). The devices are the same as in example (2). However, in this invocation, each $SIGNON card must be preceded by an S-8 card and MTS user jobs may be stacked in this example. (4) In this invocation of MTS, the task will gracefully disappear when the END-OF-FILE is sensed on *msource*. Note that there is no *syspch* unit assigned so all defaulted calls to SPUNCH (in the users' programs) will cause an error (in the user's program. As in (3), each $SIGNON card must be preceded by an S-8 card. (5) This invocation shows how the operator starts an "operator-file." The file *FYL contains the $SIGNON and $SIGNOFF cards (and other commands also). Note that, since *msource* is a file, *msink* will default to the MTS pseudo-device *DUMMY* (the infinite wastebasket).

(6) In this invocation of MTS, we have the same as in example (5), except that any output going to msink will be printed on the operator's console, i.e., OPER. Note also the comma (delimited by blanks) separating the file-name from OPER. The same use of the stand-alone comma occurs in examples (2), (3), and (4). (7) And in this example, we see the invocation of the operator file *FILE2. Note that the name of this file is greater than four characters in length. It is split up into four-character strings (starting from the left) and padded (in the last four-character string), if necessary, with commas.

OFFLINE

NAME: OFFLINE

FUNCTION: To make a specific device, control unit, channel or the path between a channel and a control unit logically offline.

PROTOTYPES:

```
(1)  OFFLINE  devname [ devname ] ...
(2)  OFFLINE  CU  cu# [ cu# ] ...
(3)  OFFLINE  CU  devname  [ devname ] ...
(4)  OFFLINE  CH  chan-adr
(5)  OFFLINE  CH  chan-adr  cu# [ cu# ] ...
(6)  OFFLINE  CH  chan-adr  devname [devname] ...
```

USAGE: If device names are given (1), then the devices are marked as offline and unavailable for further use. If a device is allocated to some job at the time it is taken logically offline, it will remain available to that job until the job releases it.

If the string "CU" followed by by control unit numbers (2) is given, then the control units are marked logically offline. If the string "CU" followed by device names is given (3), then the control units attached to the devices are marked logically offline. Form (3) of the OFFLINE command may only be used if the devices are attached to a single control unit. Control units numbers are not in any way related to the address of a device, but the UNITS command may be used to determine the control unit number for a given unit.

If the string "CH" followed by a channel address is given (4), then the channel is marked logically offline.

If the string "CH" followed by a channel address and control unit numbers is given (5), then the paths between the channel and the control units are marked logically offline. If the string "CH" followed by a channel address and device names is given (6), then the paths between the channel and the control units to which the devices are attached is marked logically offline.

When a control unit, a channel, or the path between a channel and a control unit is taken logically offline devices may be allocated to and released by jobs, but no new I/O will be initiated using the offline units. If there are other online paths to the devices through other channels and control units, then I/O to the devices can continue. If not, any jobs doing I/O to the devices will end up in I/O wait. UNITS will show "SIO PENDING" until a path to the device is made available or the system is reloaded. Even if there are other online paths to a device, a job may still end up in I/O wait if the channel, control unit or path between the units was offlined during error recovery. In this case UNITS will show the device with "SIO PENDING ON SPECIFIC PATH" until the path is made available or the operation is terminated using UNITS ... CLR!.

The UNITS command may be used to check the status of channels, control units and devices. Certain diagnostic programs can use a specific path to a device even if it is marked logically offline. Control units should ALWAYS be marked logically offline before they are taken physically offline.

PARAMETERS AND EXAMPLES:

To offline devices:

```
offline  ptrx
offline  ptrx  rdrx
```

To offline control units:

```
offline  cu  2
offline  cu  ptrx
```

To offline a channel:

```
offline  ch  b
```

To offline paths between a channel and control units:

```
offline  ch  b  2
offline  ch  b  2  3
offline  ch  b  ptrx  rdrx
```

OK

NAME: OK

FUNCTION: To set a switch so as to print or not to print the task termination message on the console when the termination code is O.K.

PROTOTYPE: OK {ON|OFF}

USAGE: Normally, when a job terminates with code O.K., a termination message to that effect is not printed on the console. This job sets a bit so that this message can either be bypassed (not printed) or be printed. Note that if OK was ON, then when any MTS task started by HASP terminates, you will get the O.K. message (which will tax the console just a little bit more).

PARAMETERS: ON or OFF

EXAMPLES:

```
ok on

ok off
```

ONLINE

NAME: ONLINE

FUNCTION: To make a device, control unit, channel, or the path between a channel and a control unit available. This is the inverse of the OFFLINE job.

PROTOTYPES:

```
(1)  online  devname [ devname ] ...
(2)  online  cu  cu# [ cu# ] ...
(3)  online  cu  devname [ devname ] ...
(4)  online  ch  chan-adr
(5)  online  ch  chan-adr  cu# [ cu# ] ...
(6)  online  ch  chan-adr  devname [devname] ...
```

USAGE: If device names are given (1), then the devices are marked as online and available for use.

If the string "CU" followed by by control unit numbers (2) is given, then the control units are marked logically online. If the string "CU" followed by device names is given (3), then the control units attached to the devices are marked logically online. Form (3) of the ONLINE command may only be used if the devices are attached to a single control unit. Control units numbers are not directly related to device addresses, but the UNITS command may be used to determine the control unit number for a given unit.

If the string "CH" followed by a channel address is given (4), then the channel is marked logically online.

If the string "CH" followed by a channel address and control unit numbers is given (5), then the paths between the channel and the control units are marked logically online. If the string "CH" followed by a channel address and device names is given (6), then the paths between the channel and the control units to which the devices are attached is marked logically online.

Devices, control units, channels and the paths between channels and control units may be taken logically offline by the OFFLINE job, they can be marked initially offline in the TABLES assembly, or they will be marked logically offline if they are physically offline when the system is IPLed. The UNITS command may be used to determine the status of channels, control units and their paths.

PARAMETERS AND EXAMPLES:

To online devices:

```
   online  ptrx
   online  ptrx  rdrx
```

To online control units:

```
   online  cu  2
   online  cu  ptrx
```

To online a channel:

```
   online  ch  b
```

Only one channel may be onlined in a single command.

To online paths between a channel and control units:

```
   online  ch  b  2
   online  ch  b  2  3
   online  ch  b  ptrx  rdrx
```

OPERATOR

NAME: OPERATOR

FUNCTION: To start up a 3270 as the operator's console.

PROTOTYPE: OPERATOR [console-device [printer-device]]

USAGE: The OPERATOR job program is the task which controls the 3270 operator's console and echoes console messages to the console printer. This job is normally started automatically after the system is IPLed.

See the sections "DEVICE COMMANDS" and "3066 or 1052 SUPPORT AS BACKUP" for details on the operation of the OPERATOR job, and for instructions on switching between a backup console and a 3270 console.

If the OPERATOR task stops or is stopped, it can be restarted by entering the OPERATOR job request. If the parameters are omitted, it will be restarted using the default 3270 display and printer.

PARAMETERS:

1. console-device specifies a 3270 display to be used as the console. If the parameter is omitted, the usual display will be assumed.
2. printer-device specifies a 3284, 1052, 3211, or 1403 to be used as a console printer. If it is left out, the usual printer is assumed. To start the OPERATOR job without a printer, this parameter may be specified as NONE.

Note that the parameters to OPERATOR are not separated by commas.

EXAMPLES:

```
operator ds02
operator
operator ds02 none
```

The most usual ways that the OPERATOR command will be used at the U of M Computing Center are as follows:

Example 1

- The OPERATOR job must be stopped
- Press the INTERRUPT button on the ES3090/300J console to the backup console
- Type OPERATOR
- Press "enter"

Example 2

If the 3270 in use as the operator's console, won't respond, sign on at any terminal:

```
 • sy
 • - t n operator
   - 0002 OPERATOR 028140    DORMANT    ; DS02 ptrx
 • - o stop 2
 • - t n operator
   - JOB NOT FOUND
 • - o operator
 • - t n operator
   - 1627 OPERATOR 029151    DORMANT    ; DS02 ptrx
 • - s
 • signoff
```

PDP

NAME: PDP [prefix]

FUNCTION: To initiate the Paging Drum Processor. This is the task that does all the system paging. This is an absolute non-reentrant job (so there can be only one active use of the PDP job program at any one time).

PROTOTYPE: PDP

USAGE: The Paging Drum Processor is automatically started in the IPL sequence. If it is restarted it will attempt to regain control of the fixed head file and disks it had when it was first started and also will start using any others that are now available. If the PDP is started without any type of paging device (2301, 2305, 3805, 3825, 4305, or disk), no paging will be done but PAGE CONTROL BLOCKS (PCBs) will still be released. In this case a condition may occur when all the available real core pages are in use and more are requested for allocation --- in such situations, some tasks using virtual memory will have to give up some of it or the system must be re-IPLed.

PARAMETERS: If given a 3 character prefix to be used by the PDP when it searches for paging disks to use. If no prefix is given, the prefix PAG is used.

EXAMPLE: PDP

PDROP

NAME: PDROP

FUNCTION: To remove Paging Devices from PDP

PROTOTYPE: PDROP device-i.d. [device-i.d.] ...

USAGE: To allow paging devices (Solid State Disks or Fixed Head Files (but not paging disks)) to to be romoved form an already running PDP without stopping it.

PARAMETERS: device-i.d. of the equipment, e.g., 3805 or 3825, to be removed from the PDP.

EXAMPLES: pdrop dev dev

pdrop fb00 fb01 fb02 (gets rid of all of our presently owned virtual memory except for the paging disk.)

SYSTEM RESPONSE: DEV HAS BEEN DROPPED BY THE PDP

NOTE: The process may take a while since the all the pages must be removed form the device. This job will not work for paging disks, only for drums and similar devices. One or more devices may be removed at one time, i.e., per line.

See PGET to re-attach VM devices.

PGET

NAME: PGET

FUNCTION: To give additional paging devices to the PDP.

PROTOTYPE: PGET device-i.d.

USAGE: To allow additional paging devices (disks or fixed head files) to be added to an already running PDP without stopping the PDP.

PARAMETERS: device-i.d. of the disk, 2301, 2305, 3805, 3825 or 4305, to be given to the PDP.

Example: pget fb00

SYSTEM RESPONSE:
FB00 ACQUIRED

Note: While it is true that we don't currently have any 2305-type devices, it isn't at all clear that we won't in the future. Almost all of the paging devices on the market today are 2305-type devices and when we benchmark machines we are almost always provided with 2305-type paging boxes.]

Note: Before adding a 2305 or 4305 to the system via PGET, it should be formatted using the program INIT:FMT2305 which is permitted "read" to ccid W035. First, make sure that the hardware itself is online. Next, put the device online using all eight addresses or at least the first and eighth address, e.g., online ff08 ff09 ... ff0f or online ff08 ff0f. The program is invoked with a "$run" command. A device will be asked for. Devices should be entered one per line. Only one device may be added at a time, i.e., per line.

EXAMPLE:

JOB PROGRAMS-PGET

```
   $r init:fmt2305

   EXECUTION BEGINS

   0PFMT: ENTER DEVICE NAME,TYPE

   ff16

   PFMT: FORMATTING FF16

   0SECTOR NUMBERS ARE:  03  3F  7B

   $endfile
```

If a device is unavailable, i.e., offline, then a comment will appear mentioning that fact and the program will ask for an IGNORE or RETRY by the operator. To terminate the program, use an end-of-file.

Fixed-Head-Files are automatically formatted when an IPL occurs.

PN

NAME: PN

FUNCTION: To load the PN, 1403 or 3203 type print chain or train image into the Universal Character Set buffer on the printer controller. Additionally, for 3203 type printers, the forms control buffer (FCB) is loaded for the standard MTS carriage control channel assignments.

PROTOTYPE: PN device-i.d. [ECHO] [FOLD] [ALLOW] [NOMSG]

USAGE: If one wishes to change print chains or trains on a printer, it is necessary to load the UCS buffer in the printer controller with the correct character set. If the UCS buffer gets scrambled for any reason, one would also want to reload it. In addition, at system IPL time, it is wise to reload the UCS buffer. The contents of the UCS buffer may also be echoed (on the associated printer) for verification.

PARAMETERS: 1) The device-i.d.

OPTIONS: The following opetions are recognized:

ECHO causes a print train verification message to be printed after loading the UCSB.

FOLD enable folding when loading the UCSB. This allows the printing of lower case letters with an upper case only print train. This option should not normally be used with MTS.

ALLOW allows data check to be presented when an invalid or an uncomparable character is printed. Normally data checks should be blocked with MTS.

NOMSG supress the verification message normally printed on the operators' console after the buffer load is completed.

EXAMPLES:

```
pn ptrx
pn ptrx echo
pn ptrx nomsg
```

P11

NAME: P11

FUNCTION: To load the P11 3211 type print chain, train, or band image into the UNIVERSAL CHARACTER SET buffer on the printer controller. The buffer is echoed after loading to verify its contents.

PROTOTYPE: P11 device-i.d. [8LPI]

USAGE: If one wishes to change chains, trains or bands on a printer, it is necessary to load the UNIVERSAL CHARACTER SET buffer in the printer control unit with the correct character set. If the UCS buffer gets scrambled for any reason, it should then be re-loaded. In addition, at system IPL time, it is wise to re-load the UCS buffer. The contents of the UCS buffer are echoed on the associated printer for verification. Normally, the printer operates at 6 lines per inch, but this can be changed to 8 lines per inch if the 8LPI parameter is used.

PARAMETERS: 1) The device-i.d. of a printer
2) 8lpi (optional)

EXAMPLES:

```
p11 ptrx
p11 ptrx 8lpi
```

QN

NAME QN

FUNCTION: To load the QN, 1403 or 3203 type print chain or train image into the Universal Character Set buffer on the printer controller. Additionally, for 3203 type printers, the forms control buffer (FCB) is loaded for the standard MTS carriage control channel assignments.

PROTOTYPE: QN device-i.d. [ECHO]

USAGE: If one wishes to change print chains or trains on a printer, it is necessary to load the UCS buffer in the printer controller with the correct character set. If the UCS buffer gets scrambled for any reason, one would also want to reload it. In addition, at system IPL time, it is wise to reload the UCS buffer. The contents of the UCS buffer may also be echoed (on the associated printer) for verification.

PARAMETERS: 1) The device-i.d.

OPTIONS: The following opetions are recognized:

ECHO causes a print train verification message to be printed after loading the UCSB.

FOLD enable folding when loading the UCSB. This allows the printing of lower case letters with an upper case only print train. This option should not normally be used with MTS.

ALLOW allows data check to be presented when an invalid or an uncomparable character is printed. Normally data checks should be blocked with MTS.

NOMSG supress the verification message normally printed on the operators' console after the buffer load is completed.

EXAMPLES:

```
qn ptrx
qn ptrx echo
qn ptrx nomsg
```

SE

NAME: SE

FUNCTION: To print out the first twelve words in the device table for a specified device.

PROTOTYPE: SE device-i.d.

USAGE: This command is used when the operator wants to examine the status of some device, e.g., to see if it is waiting for a device end. The format of the printed information can be found in an Assembler COPY section called COPY:LLMPSEQU.

PARAMETER: A device-i.d.

EXAMPLE: se t905

NOTE: There is no SE job as such since it is a supervisor command. This command can't be used on the 3270 operator's console. Use the /DEV command instead. The /DEV command can be found in the "Systemstatus" section of this manual.

SHUTDOWN

NAME: SHUTDOWN

FUNCTION: To SHUTDOWN the terminal portion of the system with the option that all MTS terminal tasks are to terminate immediately.

PROTOTYPE: SHUTDOWN [ALL|PART]

USAGE: If no parameters are given, then all currently executing non-batch MTS tasks will be allowed to terminate of their own accord. When an MTS terminal user signs off, the associated line will be disabled. Terminal lines which are IDLE will be disabled immediately so that no user can sign on on any of those lines. If the ALL parameter is given, all non-batch MTS tasks are terminated immediately as if the STOP command had been given for each task. Terminal tasks are stopped first followed by *-file and OPER tasks. If the PART parameter is given, all MTS terminal tasks are terminated, but any OPER and *-file jobs continue to execute. MTS batch tasks are not affected by SHUTDOWN.

Note: After SHUTDOWN ALL has been used, a system IPL is necessary, not just a STARTUP.

EXAMPLES:

```
shutdown

shutdown all

shutdown part
```

For the UMnet/MERIT Network PCPs, if *SDM or *SDB cannot be run,

Go to a terminal which can access the network such as the IBM PC near the operatrs' console.

At "which host", type
net hg (or, if HG can't be accessed)

net wu

signon w035

If using HG machine, do the following:

```
        $sou mnop:sdm (and/or)

        $sou mnop:sdb
```

If using WU, do the following:

```
        $sou mnop:sdm (and/or)

        $sou mnop:sdb
```

After you have done the "$sou":

> enter a one-line message as you do when using *SDM or *SDB from the operators' consoles. This procedure is used to enter a one-line UMnet/UMnet message which will be printed for any user attemting to access which ever system is down. It will <u>not disconnect</u> any users currently connected to the UB or UM systems.

There is no "shutdown" facility used here for the PCPs since even though the system may be down at UM, the MERIT network is still in use.

SIGNONM

NAME: SIGNONM

FUNCTION: To print a signon message to all users before they sign on.

PROTOTYPE: SIGNONM

USAGE: This job-program allows the operator to specify that a certain signon message is to be printed immediately before a user signs on. For batch users, it will appear on the first page of his output. For terminal users, it will appear after the brief output that specifies what line he is on and what MTS task is enabled for that line. After the SIGNONM command has been given, the SIGNONM job program will read a line from the operators' console. The operator may then type in a signon message. Pressing the "cncl" button will cause the read to be repeated, and a null line will delete any active signon message.

NOTE: As with the BROADCST command, the operator should keep in mind that the carriage width of some terminal devices is normally limited to seventy characters. If the operator gives a SIGNONM message longer than this, the message will be truncated at the current carriage width.

PARAMETERS: none

EXAMPLE:

```
signonm
SIGNONM..system will be up until 20:00 today
```

NOTE: There can only be one SIGNONM message at a given time and the message must be re-entered following a system reload.

The MTS $Message command provides an alternative method of setting signon messages that avoids these restrictions. See Signon Message using $Message System later in this manual.

SOFTCHK

NAME: SOFTCHK

FUNCTION: To control the recording of soft (recoverable) machine checks.

PROTOTYPE:
```
SOFTCHK OFF
SOFTCHK ON [n]
SOFTCHK
```

USAGE: SOFTCHK OFF disables the recording of soft machine checks. SOFTCHK ON [n] enables the recording of soft machine checks for "n" errors. If the command, SOFTCHK is issued without an integer, information will be printed concerning how many soft errors have occured and how many more may occur before the limit is reached. The limit can be raised by issuing another SOFTCHK n command. When a reload occurs, a default limit of "SOFTCHK 16" is set by the system.

PARAMETERS: The largest value for "n" is 1000. If you use a very large value for "n", you you may get many errors which, if occurring fast enough, could tie up the system; but no damage will be done. The errors that can't be logged will be discarded. Normally, the count ("n") should be omitted since the default of 16 is satisfactory. SOFTCHK OFF is reset each hour and reverts back to the default limit.

EXAMPLE:
```
softchk off

softcjk on 25
```

softchk alone will bring a response similar to:

```
    SOFTCHK 5 SOFT ERRORS HAVE OCCURED.   THE
LIMIT IS 21.
```

STARTUP

Note: If a Shutdown All has been used, a system IPL is necessary, not a STARTUP.

NAME: STARTUP

FUNCTION: To cancel the effect of shutdown on MTS terminal tasks.

PROTOTYPE: STARTUP

USAGE: The STARTUP job will cancel the effect of the SHUTDOWN job on any *currently* enabled lines. This job will *not* re-enable any lines that were disabled by the SHUTDOWN command --- such lines must be re-enabled manually. STARTUP has meaning *only* to MTS terminal tasks.

PARAMETERS: none

EXAMPLE: startup

STAT

NAME:

FUNCTION: To collect statistics of the operating system and to record them on tape.

USAGE: STAT tape1 [tape2]

This job should be run under the supervision of a systems programmer. Data collection proceeds on a tape given as a parameter. Persons wishing to use this facility should refer to the University of Michigan Computer Center Memo entitled "The MTS Data Collection Facility" for further information about this job program.

STATSW

NAME: STATSW

FUNCTION: To dynamically change the parameters that a STAT job is using.

USAGE: STATSW

As for the STAT job, potential users are referred to the University of Michigan Computer Center Memo entitled "The MTS Data Collection Facility." This job program reads commands from the operators' console.

COMMAND SYNTAX:

```
1. JOBS {ALL|MTS|jobnolist} [+|-] ITEMS {ALL|itemlist}

2. - JOBS {ALL|MTS|jobnolist}

3. + JOBS {ALL|MTS|jobnolist} ITEMS {ALL|itemlist}

4. [+|-] INTS {ALL|intlist} [{CHS|CHANS} {ALL|chanlist}

5. {+|-} {CHS|CHANS} {ALL|chanlist}
```

where: joblist is a list of UMMPS job numbers separated by blanks

itemlist is a list of stat job item numbers separated by blanks

chanlist is a list of channel numbers separated by blanks

intlist is one or more of the following keywords separated by blanks: SVC, I/O, IO, EXT, PGNT or SIO.

P may be used in place of +.

M may be used in place of -.

STOP

NAME: STOP

FUNCTION: To forcibly terminate a task.

PROTOTYPE: STOP nnnnn

USAGE: If the task is an MTS task, it will be forced into signoff --- there will be no loose ends as there are with the BLAST job. If the task is not an MTS task, it may have enabled a "trap" for the STOP command --- if it has done so, it depends on what that For a terminal task, if STOP is successful, the job should terminate in a few minutes. This allows a terminal task to go through a *full* signoff procedure. If, after a reasonable time, the task does not terminate (with a code of KILL), then a BLAST may have to be used. The STOP command should *not* be used more than once on the same job. If the task is a terminal task, the operator will have to re-enable the line (with the MTS command) if that line is to be accessible to users again.

As with the BLAST command, make sure the task you are STOPping is indeed the one that you wish to STOP and not that of another (poor) user or a HASPLING task.

PARAMETER: nnnnn - A job number.

EXAMPLE: stop 345

NOTE: The SYSTEM STATUS command /kill nnnnn performs the same function and is especially useful if all job tables are in use.

TABLMOD

NAME: TABLMOD

FUNCTION: To display or modify information in the shared file table.

USAGE: Takes no parameters. Reads input lines until it gets a "Cncl". Commands available are:

STATUS

Prints out:

1. How many files are currently in the shared file table.

2. How many matrix computations have been performed since the last reload,

3. How many (multiple file) deadlocks have been detected since the last reload, and

4. How many jobs are currently waiting on locked files.

| HOGL

| Invoked to list large users of the
| "incore" file table.

VERIFY

Same as STATUS plus checks the consistancy of the shared file table. VERIFY also prints out the number of available entries and elements.

TRACE

Same as VERIFY plus prints out the entire shared file table file by file.

Each file is listed on a separate line in the form

```
   fname -entry1 entry2 ...
where entry is of the form
   jobnum ostatus lstatus wstatus
where jobnum is the job number,
      ostatus is NOTO, OPEN, INVLD
```

lstatus is NOTL, LOCKR, LOCKM, LOCKD,
wstatus is WAITO, WAITR, WAITM, WAITD.
Ex: W045BIGLINE - 0003 OPEN LOCKM

LSTAT FILE fname

prints info as above for single file.

LSTAT JOB jobnum

prints info as above for single job.

LOCATE FILE fname

displays half-word offset of file into ICTAB.

NOTE: Don't use CLEAN, CLEANF, DEQ or DEQF without permission from a senior systems programmer.

CLEAN JOB jobnum

takes the job off all open or locked and waiting lists.

CLEANF JOB jobnum FILE fname

same as CLEAN but for one file only.

DEQ JOB jobnum

takes job off all waiting lists.

DEQF JOB jobnum FILE fname

same as DEQ except for one file only.

TAPE COMMANDS-OPERATOR CONTROLLED

Names: RUN,REW,WTM,FSF,FSR,BSF, BSR, and ERG.

Functions: To manipulate tapes and write tape marks (file marks) on tapes using the operators console

Prototypes:

```
RUN T90X
REW T90X
WTM T90X Y
FSF T90X Z
FSR T90X Z
BSF T90X Z
BSR T90X Z
ERG T90X Z
```

Usages:

REW rewinds tape X to the load point
RUN rewinds and unloads tape X
WTM writes Y number of tape marks (file marks) on tape X
FSF spaces tape X forward Z number of files
FSR spaces tape X forward Z number of records
BSF backspaces tape X, Z number of files
BSR backspaces tape X, Z number of records
ERG erases Z times 3 inches of tape (more or less)

Examples:

```
WTM T902 4
BSR T909 2
```

NOTE: Use the ERG and WTM jobs with caution; if you specify a tape drive other than the one intended, data may be lost.

TASKS

NAME: TASKS

FUNCTION: To determine the status of a particular task.

PROTOTYPE:

```
TASKS M   (mts jobs)
TASKS U   (user-id)
TASKS D   (device-id)
TASKS T   (device type)
TASKS nnnnn   (tasks number)
TASKS B   (batch jobs)
TASKS P   (project number)
TASKS N   [jobname]   (non-mts)
```

NOTE: This job is not normally used from the 3270 operator's console, the /T ... command being used instead.

USAGE: With the M parameter, the status of all active MTS tasks is given.

With the U userid parameter, the status of those tasks on which user userid is signed on will be printed.

With the D device-i.d. parameter, the task that currently "owns" device device-i.d. will have its status printed.

With the T-device-type parameter, the status of all those jobs owning one or more devices of device-type device-type will be printed.

With the nnnnn parameter, the status of task nnnnn will be printed.

With the N parameter alone, the status of all non-MTS tasks will be printed (e.g., HASP, PDP, HASPLINGs, etc.). If the N parameter is followed by a jobname, then the status of all tasks executing with that job-program is printed (note that for jobnames having more than four characters, the name must be split into two parameters the latter of which must contain the last four characters of the job-name).

With the P prjid parameter, the status of those tasks on which a user with project i.d. prjid is active will be printed.

With the B parameter, the status of all currently executing batch tasks will be printed.

With no parameter, the status of all tasks will be printed (this includes all of the above plus those MTS tasks which are IDLE.

The format of the status message is:

```
nnnnn pgs name jobtbl status par1 ...parn; dev
```

where:

nnnnn -- is the task number

pgs -- is the number of virtual pages being used (if it is a relocatable task)

name -- is the name of the command that initiated this task

jobtbl -- is the real core address of the associated job table

status -- a description of the job's current status

par1 ... parn -- are the parameters used to start the task

dev1 ... devn --- are the devices that the task currently has

For MTS terminal tasks, the parameters are user i.d. and project i.d. For MTS batch tasks, the parameters are user i.d., project i.d., and S-8 number. For terminal tasks, the first device parameter will be the terminal line device-i.d.

If the TASKS job program cannot find any job(s) of the specified type, the message JOB NOT FOUND will be typed.

EXAMPLES: tasks 5 -- gives information about task 5.

tasks u w025 -- gives information about the task(s) associated with user i.d. W025.

tasks d d307 -- gives information about the task that <u>currently</u> is using the device with device-i.d. d307.

tasks t 9tp -- gives information about all jobs that are using nine-track tapes (which are of device-type 9TP).

tasks n --- gives information about all non-MTS tasks.

tasks n hasp ling --- gives information about all HASPLING tasks.

TMTS

NAME: TMTS

FUNCTION: To start a task using a test (developmental) version of MTS.

PROTOTYPE: As for MTS.

USAGE: This job request is used in testing out a new version of MTS. The new version must first be loaded using the TMTS loader. It is otherwise similar to MTS.

PARAMETERS: As for MTS.

TN

NAME TN

FUNCTION: To load the TN, 1403 or 3203 print chain or train image into the Universal Character Set buffer on a printer controller. Additionally, for 3203 type printers, the forms control buffer (FCB) is loaded for the standard MTS carriage control channel assignments

PROTOTYPE:

```
TN device-i.d.  [ECHO] [FOLD] [ALLOW] [NOMSG]
```

USAGE: If one wishes to change print chains or trains on a printer, it is necessary to load the UCS buffer in the printer controller with the correct character set. If the UCS buffer gets scrambled for any reason, one would also want to reload it. In addition, at system IPL time, it is wise to reload the UCS buffers. The contents of the UCS buffer on can also be echoed (on the associated printer) for verification.

PARAMETER: device-i.d.

OPTIONS: The following options are recognized:

ECHO causes a print train verification message to be printed after loading the UCSB.

FOLD enables folding when loading the UCSB. This allows the printing of lower case letters with the "upper case only" print train. This option should normally not be used with MTS.

ALLOW allows data check to be presented when an invalid or an uncomparable character is printed. Normally data checks should be blocked with MTS.

NOMSG supress the verification message normally printed on the operators' console after the buffer load is completed.

EXAMPLES:

```
tn ptrx
tn ptrx echo
tn ptrx nomsg
```

T11

NAME: T11

FUNCTION: To load the T11 3211 type print chain, train, or band image into the UNIVERSAL CHARACTER SET buffer on the printer controller. The buffer is echoed after loading to verify its contents.

PROTOTYPE: T11 device-i.d. [8LPI]

USAGE: If one wishes to change chains, trains or bands on a printer, it is necessary to load the UNIVERSAL CHARACTER SET buffer in the printer control unit with the correct character set. If the UCS buffer gets scrambled for any reason, it should then be re-loaded. In addition, at system IPL time, it is wise to re-load the UCS buffer. The contents of the UCS buffer are echoed on the associated printer for verification. Normally, the printer operates at 6 lines per inch, but this can be changed to 8 lines per inch if the 8LPI parameter is used.

PARAMETERS: 1) The device-i.d. of a printer
2) 8lpi (optional)

EXAMPLES:

```
t11 ptrx
t11 ptrx 8lpi
```

UNITS

UNITS output appears as follows.

```
0850 PTR3 Type: 9700 DevNum: 0390 (ON ) LPM: 80 Sch:  0199  Job:
00814
```

Dev Num: A number assigned to a device by a systems programmer. They've been assigned in such a way to give channel and device information

SCH: Subchannel number. This is assigned to a device by the I/O system.

LPM: Logical path mask. This is an 8-bit mask describing how many paths there are to a device.

Example:

80 -> 1 path (path #0) available

C0 -> 2 paths(paths #0, #1) available)

(a disk will have 2 paths available)

UNITS - Old Style (No longer used)

NAME: UNITS

FUNCTION: UNITS is a program to display the configuration of channels, control units, and devices attached to the system, for printing certain aspects of the status of peripheral devices, and for doing a limited amount of error checking of the tables that describe these things.

For operations staff, the most frequent uses of UNITS will probably be to verify and clarify the effect(s) of previous ONLINE and OFFLINE commands, to investigate why a device is "hung", and just to see what is connected to what.

For systems staff, UNITS will also be used to help check new Tables assemblies.

USAGE: UNITS is available as a job program, which in case it is invoked by entering a job request at the console:

```
UNITS  [ units-command [ mod ]]
```

UNITS is also available as an MTS program, in which case it is invoked by entering:

```
$run mta.:units.o [sprint=FDname] PAR=command [mod]
```

-or-

```
$run mta:units.o [SPRINT=FDname]
command  [ mod ]
command  [ mod ]
etc.
end-of-file or MTS
```

Units takes the following commands, each of which may be followed by an optional modifier (see below):

```
D devname     to display a particular device
```

D ALL	to display all units in device table order
CU devname	to display all units on the control unit to which devname is attached.
CU cu#	to display all units on the control unit with the given number
CU cu# ... cu#	to display all units on the CU within the given range
CU ALL	to display all units in control unit order
CH chan-adr	to display all units attached to the channel with the given address
CH chan-adr ... chan-adr	to display all units attached to the given range of channels
CH ALL	to display all units in channel table order
T devtype	to display all units with the given device type
adr	to display the path (see below) with the given hexadecimal I/O address

where:

"devname" is a device name
"cu#" is a control unit number
"chan-adr" is a channel address
"devtype" is a device type
"adr" is a physical I/O address

A null command defaults to "D ALL". MTA.:UNITS.O also accepts the command MTS and can be terminated with an end-of-file.

In addition, there are four modifiers that are selectable by adding a parameter following the command.

Legal modifiers are:

B   Busy: Select only those devices that are allocated to a job.

W   Waiting: Select only those paths for which there is an I/O operation (I/O queue active or START I/O pending).

A   Active: Select the most recently active path(s).

S   Status: Print detailed device status if possible.

O   Offline: Select those paths which are logically offline in some way.

V   Verify: Verify that the tables are internally correct.

CLR!   CLRIO: Does both a FAKE I/O _and_ a CLEAR I/O. (Use only with _extreme_ caution!)

B, W, and A imply S. If no modifier is given, detailed information is not printed.
"devtype" is a device

OUTPUT _from_ _UNITS_

The output from a UNITS request has several parts:

PART 1.   The _path_ _specification_. This part shows how a given device is connected to the system--through which channel and control unit-- and the logical ONLINE/OFFLINE-ness of each part of the path.

PART 2. The address. This part specifies the address associated with the path given in PART 1.

PART 3. The job number. This part appears if a particular task owns the device in question, and gives the task number of the job.

PART 4. The general status information. This part appears if the S, B, W, or A options follow the units-command. It gives selected general facts about the device and about the control unit and channel mentioned in the path specification.

PART 5. The device status information. This part appears only if the S, B, W or A options follow the units-command AND if the device belongs to some task. It gives more specific information about the current state of the device.

PART 6. The error information. This information appears only if the "V" option is in the units-command AND if a possible inconsistency has been discovered in the Supervisor tables.

Subject to the conditions just mentioned, parts one through four appear once for every path to every device that meets the criteria given in the units-command. For instance, if there are two paths to device 'D240', then a units-command of'D D240' will result in two sets of output, one set for each path to D240.

Normal output is one line per path. If the S, B, W, or A options are used, PARTs 4 and 5 appear on additional lines. If the "V" option is used, any error messages appear on extra lines.

The actual output format can best be explained by these four examples.

EXAMPLE 1.

```
   Units D D240

        CH 1 * CU 14 -> DEV D240 ADDR:140
        CH 2 -> CU 14 -> DEV D240 ADDR:240
```

These two lines are interpreted as follows: First, there are two paths to D240. Data can go either through channel 1 and control unit 14 to D240, or through channel 2 and control unit 14 to D240. However, the '*' symbol in the first path means that the path through channel 1 is logically offline. Consequently, the Supervisor won't schedule any I/O on the path through channel 1.

Note that channel 1 might still be used to communicate with some other device and this is the difference between an offline channel and an offline path between a channel and a control unit.

Other things we learn from these two lines are the two hardware I/O addresses of the two paths (140 and 240).

EXAMPLE 2.

```
   Units D DS15 S

        CH 0 -> CU 7 -> DEV DS15 ADDR:015
             JOB(41), CU BUSY, 64 CH OPS, 1 CU OPS
             DEV INITIALIZED, NO ERRORS
        CH 1 * CU 7 -> DEV DS15 ADDR:115
             JOB(41), CU BUSY, 1 CH OPS, 1 CU OPS
```

These lines show the effect of the "S" option. In addition to the information described in Example 1 above, the general status and device status parts appear. The general status (the second line) gives four facts:

1. The device is owned by job (i.e., task) number 41,
2. The control unit is currently busy,
3. There are 64 operations outstanding on

the channel,

4. There is one operation in progress on the control unit.

The device status (the third line) gives two other details:

1. The device has been initialized. This means that the Supervisor has allocated an area of control storage (a Device Table) to record the progress of I/O on the device.

2. The device has "NO ERRORS", that is, there are no current error indications for the device.

Note that although the general status line appears twice-- once for each path--the device status line appears only once. The device status line always appears beneath the description of the path that has been most recently used for an I/O operation.

EXAMPLE 3.

```
Units D ptrx

    CH0 -> CU 9 ->DEV ptrx(OFF) ADDR:082
```

This example shows the notation employed in the path description to describe OFFLINE status. The characters "OFF" appear after any channel, control unit, or device that is logically OFFLINE. A channel/control unit/device may be logically OFFLINE because someone has made it so using the OFFLINE job, because it was initialized as logically OFFLINE when the supervisor tables were assembled or because it was marked logically OFFLINE because it was found to be physically offline when the system was IPLed.

This sort of OFFLINE notation, together with the path OFFLINE notation described in EXAMPLE 1, make the total of four different ways that a device can be OFFLINE. All four of these ways in simultaneous effect would result in a path description like this:

```
CH  0(OFF)  *  CU  9(OFF)  ->  DEV  ptrx(OFF)
ADDR:082
```

To make this device operational, someone would have to ONLINE not only the device itself, but also Control Unit 11, Channel 0, and the path from Channel 0 to Control Unit 11.

EXAMPLE 4.

```
Units D PTRx V

     CH 2 * CU 3 -> DEV ptrx ADDR:208
          *REV PATH POINTS TO PTRx
```

This example shows the kind of error printed by use of the "V" option. These error messages point to possible inconsistencies in the Supervisor Tables.

USING MTA.:UNITS.O

```
$run mta.:units.o

d ds0a
CH 7 -> CU 18 -> DEV DS0A ADDR:70A TYPE:3270 JOB(512)

d d001
CH B -> CU 19 -> DEV D001 ADDR:B41 TYPE:3330 JOB(3)
CH 2 -> CU 1 -> DEV D001 ADDR:241 TYPE:3330 JOB(3)
CH 4 -> CU 2 -> DEV D001 ADDR:441 TYPE:3330 JOB(3)

b41
CH B -> CU 19 -> DEV D001 ADDR:B41 TYPE:3330 JOB(3)

d ds0a s
CH 7 -> CU 18 -> DEV DS0A ADDR:70A TYPE:3270
        JOB(512)
        JOB INITIALIZED, NO ERRORS, GTUNIT QUEUED
        DEVICE STATUS: CHANNEL END, DEVICE END

CU DS0A W
CH 7 -> CU 18 -> DEV PTRx ADDR:70F TYPE: 3284
        JOB(2), 2 CH OPS, 1 CU OPS
        DEV INITIALIZED, NO ERRORS, I/O QUEUE  ACTIVE,  TASK
        WAITING
               I/O OP AWAITING CHANNEL END
```

```
T 3350 A
CH 2 -> CU 1 -> DEV D100 ADDR:250 TYPE:3350
        JOB(3), 1 CH OPS, 1 CU OPS
        DEV INITIALIZED, NO ERRORS
        DEVICE STATUS: CHANNEL END, DEVICE END
CH 2 -> CU 1 -> DEV D101 ADDR:251 TYPE:3350
        JOB(3), 1 CH OPS, 1 CU OPS
        DEVICE INITIALIZED, NO ERRORS
        DEVICE STATUS: CHANNEL END, DEVICE END
CH 4 -> CU 2 -> DEV D102 ADDR:452 TYPE:3350
        JOB(3)
        DEVICE INITIALIZED, NO ERRORS
        DEVICE STATUS: CHANNEL END, DEVICE END
        DEVICE  INITIALIZED,  NO  ERRORS,  I/O QUEUE ACTIVE,
        TASK WAITING
               I/O OP AWAITING CHANNEL END

  .
  .
  .

CH 4 -> CU -> DEV D115 ADDR:45F TYPE:3350
        JOB(3)
        DEVICE INITIALIZED, NO ERRORS
        DEVICE STATUS: CHANNEL END, DEVICE END
```

DESCRITIONS OF THE VARIOUS MESSAGES PRINTTED BY UNITS

CH <ch#> -> CU <cu#> -> DEV <dname> ADDR: <path> TYPE: <dtype>

The string "(OFF)" may appear following the channel number, the control unit number, or the device name to indicate that the channel, control unit or device is logically OFFLINE. The arrow between the channel and the control unit may be replaced with an asterisk (*) to indicate that the path from the channel to the control unit is logically OFFLINE. It isn't possible to logically OFFLINE the path from a control unit to a device.

JOB(<job#>) — Device allocated.

DEV FREE — Device not allocated.

NOT READY — Task waiting for SPUR DEV END (SVC SPWAIT or SVC ATNSU).

IGNORING INITS — On while device is being freed.

SENSE HOLD	No other ops to control unit until sense following unit check is done.
CHAN BUSY	CC=2 to SIO or SIOF.
CU BUSY	Status from a CSW.
FAKE CUE PENDING	Retry every so often even if we don't get a CU end following a CU busy.
<n> CH OPS	<n> I/O operations outstanding for the channel.
<n> CU OPS	<n> I/O operations outstanding for the CU.
*CLRIO FAILED	Path was in use.
DEV INITIALIZED	There is a device table for this device.
NO ERRORS	No UNIT CHK, UNIT EXCEPTION or INOP/CHANERR.
INOP/CHANERROR	CC=3 to SIO or SIOF or channel error in CSW.
UNIT CHK	Status from CSW.
UNIT EXCEPTION	Status from CSW.
I/O QUEUE ACTIVE	A SIOF has been done for the top entry on the active I/O queue.
TASK WAITING	Task has done an SVC WAIT or an SVC ENDEXIT.
I/O OP AWAITING	
CHANNEL END	Subchannel still in use.
DEV END	Subchannel free, but I/O op not complete.
DEVICE BUSY	Status from CSW.
SIO PENDING	Top entry on active I/O queue is waiting for super to do a SIOF.

ON SPECIFIC PATH	Task request, software command chaining, sensing following UC, *DLG.
USING RESTART CAW	Software command chaining.
SENSING	Super doing sense op following unit check.
ERROR EXIT ACTIVE	Top entry on active I/O queue terminated with UC, UE or other error and an SVC SAVE hasn't been done yet.
SPUR DEV END	Power on, popped plug, FAKEIO, ...
WAIT FOR CE NOT DE	SVC CEWAIT.
LAST I/O OP INITIATED BY OLTEP	SVC STIO option.
OLTEP ATTN 1	Ditto.
OLTEP ATTN 2	Ditto.
GTUNIT QUEUED	Another task is waiting for the device.
CU END RECEIVED	CSW status.
FAKE DEVICE END PENDING	Retry operation every so often even if we don't get the device end following a device busy.

DEVICE STATUS:

- ATTENTION
- STATUS MODIFIER
- CONTROL UNIT END
- BUSY
- CHANNEL END
- DEVICE END
- UNIT CHECK
- UNIT EXCEPTION

Busy and Status Modifier is long control unit busy. Busy, Status Modifier and Control Unit end is short control unit busy.

CHANNEL STATUS:

- NOT OPERATIONAL
- PCI
- INCORRECT LENGTH
- PROGRAM CHECK
- PROTECTION CHECK
- CHANNEL DATA CHECK
- CHANNEL CONTROL CHECK
- INTERFACE CONTROL CHECK
- CHAINING CHECK

NOT OPERATIONAL status is reported as channel status X'3F'. The remaining device and channel status bits are defined in the 370 Poo.

Operator Files

Operator files are initiated using the MTS command, and, giving as a parameter, the name of a public file which will contain at least an MTS $SIGNON command. The operator file may $CONTINUE WITH another file, but there _must_ eventually be a $SIGNOFF command to terminate the task. Unless otherwise specified when initiating the MTS task, all output directed to _msink_ will be funnelled into the great infinite wastebasket. It is to your advantage to name your operator files (if you create them) with names of four characters or less (that includes the mandatory '*' at the beginning which denotes the file as being a public one). However, it is possible to use files that have longer names but then you will be fighting the supervisor command language (?) structure of four-characters (at most) per parameter.

*ABS

NAME: *ABS

FUNCTION: starts up ports AB10 - AB3F

USAGE: mts *abs

NOTE: *ABS is normally started by *MNS

*ADS

NAME: *ADS

FUNCTION: starts up ports AD10 - AD3F

USAGE: mts *ads

NOTE: *ADS is normally started by *MNS

*AES

NAME: *AES

FUNCTION: starts up ports AE10 - AE73

USAGE: mts *aes

NOTE: *AES is normally started by *MNS

*AFS

NAME: *AFS

FUNCTION: Starts up ports AF10-AF3F

USAGE: mts *afs

NOTE: *AFS is normally started by *MNS

*ANS

NAME: *ANS

FUNCTION: starts up ports AN10 - AN17

USAGE: mts *ans

NOTE: *ANS is normally started by *MNS

*AUS

NAME: *AUS

FUNCTION: starts up ports AU10 - AU6F

USAGE: mts *aus

NOTE: *AUS is normally started by *MNS

*AVS

NAME:	*AVS
FUNCTION:	starts up ports AV10 - AV6F
USAGE:	mts *avs
NOTE:	*AVS is normally started by *MNS

*AWS

NAME: *AWS

FUNCTION: starts up ports AW10 - AW6F

USAGE: mts *aws

NOTE: *AWS is normally started by *MNS

*AXS

NAME: *AXS

FUNCTION: starts up ports AX10 - AX6F

USAGE: mts *axs

NOTE: *AXS is normally started by *MNS

*AYS

NAME: *AYS

FUNCTION: starts up ports AY10 - AY6F

USAGE: mts *ays

NOTE: *AYS is normally started by *MNS

*AZS

NAME: *AZS

FUNCTION: starts up ports AZ10 - AZ6F

USAGE: mts *azs

NOTE: *AZS is normally started by *MNS

*BBS

NAME: *BBS

FUNCTION: To start up the "babysit" program which watches for an i.d. This is used when it is suspected that someone is fraudulently using an i.d, and notification is wanted if that i.d. signs on.

USAGE: mts *bbs

```
       MTS  ENTER USER ID'S ONE PER LINE.
       MTS..0ab2
       MTS..
```

An end-of-file condition was entered on the last line above to terminate the list. When someone signs on with the i.d.(s) being watched for, the following is printed on the operator's console:

```
              MTS


       ******
       ****** USER 0AB2 SIGNED-ON
       ******
```

To get rid of a watchdog task, use the STOP job.

*CCD

NAME: *CCD

FUNCTION: The program causes the deferment of long or difficult plots so that they may be plotted at a more convenient time.

USAGE:

```
mts *ccd
```

You will be promted for a ccid. Enter all the ccids for which plots are to be deferred, one per line. Terminate the list with an end-of-file ("CNCL"). The deferred plots were queued by the user in the normal way, but placed on a separate plot tape.

To defer a users plot, a program is run to tell *CCP that all plots for a particular signon id are to be deferred. After this is done, *CCP will ignore that users plots when doing the scheduled plot runs (the "deferred plots will be left in the queue). The deferred plots can be put on a tape, at the operator's convenience by running *CCP using a special plot tape. The system ensures that only deferred plots are placed on this tape. When it is no longer necessary to defer the user's plots, a program is run to tell *CCP to process the plots normally.

EXAMPLE:

```
mts *ccd
ENTER SIGNON ID:
sys.
PLOTS FOR SYS.  WILL BE DEFERRED
mts.
PLOTS FOR MTS.  WILL BE DEFERRED
$endfile
**CANCELLED**
```

This program writes information into SYS.:CCPPRIO telling *CCP which ccids to defer plots for. _All_ plots for the ccid will be deferred until *CCD is instructed otherwise (see below).

Running Regular Plots

Run *CCP in the normal way. Deferred plots will be left in the queue, and their receipt numbers will be listed on the plot sheet under "requeued".

Running Deferred Plots

Run *CCP in the normal way, _except_ use the plot tape marked "deferred plots". This tape is in the rack with the other plot tapes. The deferred plots will be added to the logical end of the tape, i.e., after the plots already there. The deferred plots will remain on the deferred tape until the tape is reinitialized (see below). All normal plots will remain on the queue, and their receipt numbers will appear in the list of requeued plots on the plot sheet.

Plotting Deferred Plots

Run PLT in the usual way. Note that the deferred tape is added to, so that you may not want to plot all of the plots. The "WHICH TAPE" message from PLT will say that the deferred tape is being plotted:

EXAMPLE:

```
MTS  PLOTS FROM DEFERRED TAPE;  TAPE WRITTEN 13:46.15 03-11-78.

            MTS  1=BLAK/4     2=BLAK/3     3=BLAK/6

            MTS..ok
```

To Stop Deferring User Plots

This should be done to allow a users' plots to be processed in the normal way after they have been deferred.

Start the MTS job in the file *CCD. You should be promted for a ccid. Enter all ccids for which you wish plots to no longer be deferred, one per line. Terminate the list with an end-of-file ("CNCL").

EXAMPLE:

```
mts *ccd
ENTER SIGNON ID:
```

```
mts.
PLOTS FOR MTS. WILL BE PROCESSED NORMALLY.
sys.
PLOTS FOR SYS. WILL BE PLOTTED NORMALLY.
**CANCELLED**
```

This is, of course, exactly the program which defers plots. It just changes the status of any id you give it, as:

```
mts *ccd
ENTER SIGNON ID:
sys.
PLOTS FOR SYS. WILL BE DEFERRED
sys.
PLOTS FOR SYS. WILL BE PROCESSED NORMALLY
**CANCELLED**
```

*CCL

NAME: *CCL

FUNCTION: This program initializes plot tapes for use by the program *CCP.

USAGE: mts *ccl

The program will print-

```
ENTER DRIVE DENSITY NAME
```

Type in the name of the drive, the density, and the name of the plot tape separated by blanks. The program will label the tape with the name PLOT and write a file used by *CCP to identify the tape. The same information may be entered for another plot tape.

Use an "end-of-file" to terminate the program.

The name of the plot tape must appear in exactly the right format, e.g.,

```
MON 1000
```

but not

```
MON  1000
```

The weekday names used are-

```
MON   TUE   WED

THU   FRI   SAT
```

and the times are-

```
1000   1400   1800   2200   0200
```

REMARKS: The plot tape names WEEKEND and DEFERRED are also recognized.

EXAMPLES:

```
t903 1600 fri 2200

t905 1400 SUN
```

t914 1400 SAT

*CCP

NAME: *CCP

FUNCTION: To produce a tape for the CALCOMP plotter from the plot requests that users have queued.
*CCP should be run by the Operators at the appropriately scheduled times.

USAGE: mts *ccp

System replies with a receipt number in the following manner:

"*PRINT* ASSIGNED RECEIPT NUMBER 610418"

Where is a plot tape? (system asks)

Operator mounts appropriate (check sticker for day and time) tape on a 9 track unit and responds with:

T90X, X being the number of the unit to be used.

System responds with:

```
        USER W175(W175) FILE *CCPFILE
        ????
*CCP is done.  approximately 26 min.  plotting; Blue used.
```

NOTE: If colors other than black are to be used, this will be mentioned in above line.

A tape has now been prepared for plotting.

NOTE: *CCP is usually started automatically by *CLK

CCT

NAME: *CCT

FUNCTION: To let the operators know how much time the plotting session would take, and how many user tapes would be used if the plots were dumped on tape and run at the time *CCT is run.

USAGE: mts *cct

Response on operator's console might be:

```
01721 PLOTTING TO DO; 27 MIN.  REQUIRING NO USER TAPES.
```

*CHK

NAME: *CHK

FUNCTION: Displays list of OFFLINE equipment on OPER.

USAGE: mts *chk

REMARKS: *CHK is atumatically started fifteen minutes after the beginning of each shift.

*CHK should always be run following a reload.

*CLB

NAME: *CLB

FUNCTION: This program retrieves the label from a tape and also places the label in the catalog.

USAGE: mts *clb

REMARKS:A user may hit "enter" when asked for a tape label during the running of *Tapesubmit in which case, the tape label doesn't get entered. Using *CLB will remedy this.

When run, the program will ask for the drive, receipt and tape id.

*CLK

NAME: *CLK

FUNCTION: Starts the JOB SCHEDULING program. Program automatically schedules initiations of other jobs such as *CCP.

USAGE: mts *clk

REMARKS: Normally started automatically by *LAS. This job should not be started manually. Use *CLR instead.

*CLN

FUNCTION: Cleans out old information from the files used to hold machine check and unit check information; then starts *CMD.

USAGE: mts *cln (automatically started by *INIT)

REMARKS: Normally started automatically by *INIT.

*CLR

FUNCTION: Restarts JOB SCHEDULING program (*CLK)

USAGE: mts *clr

REMARKS: *CLR is used if JOB SCHEDULING program must be restarted at a time other than reload.

*CMB

NAME: *CMB

FUNCTION: Program to combine statistics tapes into tapes of "format MTA". Tapes to be combined are produced by *STA. This program produces tapes which are blocked in a rather none-standard manner. However, most system programs which read *STATISTICS data are capable of deblocking these tapes.

INPUT TAPES: Blocked or unblocked tapes which may be unlabeled or labeled STAT, STATM, or STATC. Labeled tapes with other volume names are not acceptable.

OUTPUT TAPES: Blocked by this program into blocks not longer than 4352 bytes. Tape format is U(4352). Tapes may be unlabeled or labeled STATC.

USAGE: mts *cmb

Strip about 40 feet from the front of the new output tape.

The program will ask for the first output tape (must be mounted with ring in). Reply with the tape unit name on which it is mounted - for example, T901. The program will then ask for successive input tapes.

Input tapes will be requested in the following manner:

```
ENTER INPUT TAPE NAME (AND VOLUME NAME) OR "DONE"
```

To end the combining process, enter DONE in place of the tape unit name.

NOTE: Generally, this program is used on the first or second day of the month. When an output tape is asked for, a new tape (remember to strip about 40 feet from the beginning of the new tape) labeled (txxx 6250 statc umcmpctr), should be used. When input tapes are asked for, use the output tape from the last *CMB run if it isn't full, then use the monthly statistics tape from the preceding month. When an output tape has been filled, a new output tape should be mounted. Beginning and ending dates will be printed on the operators' console. These dates should be written on the stickers on the ouput tapes. If a second ouput tape is needed and is not filled, then this should be used as the input tape for the next *CMB run. Filled output tapes should have a copy made of them. The original (copy 1) should be stored at the "print building" and the copy (copy 2) in the racks at the Computing Center.

The format for monthly statistics tapes is-
      VB(16379,4000)

The format for the combined statistics tape is-
      U(4352,32767)

Older Version

NOTE: Generally, this program is used on the first or second day of the month. When an output tape is asked for, a new tape, labeled (txxx 1600 statc umcmpctr), should be used. When input tapes are asked for, the first one should be the monthly (STATM) statistics tape for the preceding month, e.g., if it is Oct. 2, use the September monthly statistics tape as the first input tape. When a second input tape is asked for, use the daily statistics tape which carried over from September to October. The third input tape would be the Filecharge tape for September.

*CMD

NAME: *CMD

FUNCTION: Program to write command statistics and unit-check messages into a file. This program is automatically started by *CLN.

USAGE: Is normally started up automatically. Requires only "mts *cmd". Takes no parameters.

NOTE: See the CMDSTAT writeup for details and explanatiions.

Problems: If the message,

"THE CMDSTAT PICKUP PROGRAM (*CMD) ISN'T RUNNING OR IT IS HUNG"

is printed on the console, it means that an MTS job has been unable to log a *STATISTICS record by calling CMDSTAT. The first time the message is printed for a given job it means that the call to CMDSTAT has failed 60 times (one call a second). The message will be repeated every two minutes (every 120 calls) until a call succeeds. While the job is doing this the /T command will usually show the job in TIME WAIT. When this is going on, the job has the file *STATISTICS locked for modification so any other jobs that try to signoff will end up in a FILE WAIT for *STATISTICS (including *CMD if it is trying to sign off).

When you see this message you should do the following:

(1) Check to see if *CMD is running. The simplest way to do this is to say "/T U SYS" on the console and look for *CMD.

(2) If *CMD isn't running, you should restart it (MTS *CMD) and then look back at the console to see if you can figure out why it stopped.

(3) If *CMD is running, check its status several times using the "/T nnn" command. If it shows CMDSTAT WAIT, the problem has probably cleared itself up.

(4) If *CMD is in a FILE WAIT, use TABLMOD to see which file

*CMD is waiting for. If *CMD is waiting for *STATISTICS, you should start a new *CMD job (MTS *CMD).

(5) If *CMD is waiting for a file other than *STATISTICS, use the output from TABLMOD to figure out which job has the file locked. Then use the /T command several times to check the status of that job. Usually you will find that *CMD is waiting to lock INIT:UC3330LOG, INIT:MCFILE, SYS.:CMDSTATn, or SYS.:HASPLOGn and that this file is in turn locked by a staff member. If this is the case try to find the staff member and ask them to unlock the file. If the staff member isn't working from a terminal in the Computing Center you can try BROADCSTing to them and asking them to unlock the file. If you can't find the staff member in question or a BROADCST doesn't work, after waiting several minutes, you should go ahead and stop the staff member's job.

(6) If *CMD is in a FILE WAIT, but for some file other than the ones listed above or if the job that has the file that *CMD is waiting for locked doesn't belong to a staff member, you should wait a few minutes and check again. If *CMD is still waiting for the same file and that file is still locked by the same job as before, then you should job dump the job that has the file locked and then stop it. If a new job has the file locked then you should wait a few more minutes and check once again.

(7) If *CMD isn't in a FILE WAIT, use the "/J n" command to see if *CMD is completely stopped or if it is using some CPU time. If it is using some CPU time, wait several minutes and check again. If it is completely stopped try to see why. If you can't figure it out, STOP *CMD and start a new version (MTS *CMD).

(8) If *CMD shows as LOCK 9 ICTAB, it means that it is waiting to use the incore file table. The command "/DIS LOCK 9" will show you which job currently has the incore file table lock set and the "/J n" command will in turn show the status of that job. If that job seems to be using CPU time, wait several minutes and check again. If the job isn't using any CPU time, take a job dump and stop the job.

(9) If *CMD is doing DISK I/O, then it is waiting for the disk manager. Check to make sure that the disk manager is running using the command /T N DMGR. If the DMGR isn't, running get help from a systems programmer or if no systems programmer is available try restarting the DMGR. If it is running use the

DMGRSTAT job to check its status.

(10) In all cases check each job several times using the "/T n" or "/J n" commands before STOPping them. If the situation changes start the checking over from the top. If *CMD goes into CMDSTAT WAIT the problem has probably cleared itself up. If you need to stop a job, always try to STOP a job first and only if that fails after waiting several minutes should you consider BLASTing the job.

If you are working from an MTS terminal rather than the operator's console the Systemstatus CLS can be used to issue the /... commands, the $LOCKSTATUS command can be used in place of TABLMOD, and MTA:UNITS.O in place of the UNITS job. If you do work from a terminal and you find it necessary to STOP a job, make sure that there isn't another person working on the same problem from the console or another terminal.

There is almost never a need to dump the job that is printing the message about CMDSTAT being hung on the console!

*CPW

NAME: *CPW

FUNCTION: To change the password for a given account to something unknown.

USAGE: MTS *CPW

Note: User Advocate requests for password changes should be honored.

*CPW may be used to suspend the use of an MTS ID to prevent unauthorized use of the CCID. The program would normally be used at the request of the CCID's owner when the Computing Center business offlice is closed, the owner suspects that the ID is being used in an unauthorized fashion, and the owner is unable to change the password. Ater *CPW has been run, no one, including the true owner of the ID, may use the ID until a new password has been assigned by the business office.

Under certain, extreme circumstances, it may be necessary for the operator to use *CPW to suspend an account to prevent unauthorized use of an ID without a request from the owner of the ID. When this is necessary, the ID involved and the reason for the suspension should be reported to operations supervisors and to the business office. *CPW should never be used to prevent inappropriate use of an ID, instead, inappropriate use should be reported to the business office and they in turn will notify the project director for that ID. Unauthorized use is the use of a ccid and password by someone other than the owner of the ccid without the owner's permission. Inappropriate use would be when someone uses their ccid to send unwelcome messages, run jobs which print out lots of blank pages, etc.

Requests to suspend an ID must be made in person to the Machine Room operator at the Computing Center on North Campus. The individual must present some sort of identification that includes a picture, e.g., a drivers' license. Telephone requests to suspend an ID must be rejected.

Running *CPW will result in the following prompts at the operators' console:

```
Operators' Name?

Requestors' Name?
```

Address?

Telephone?

Signon ID?

Re-enter ID for verification?

The operator should respond with his or her name; the name, address and phone number of the person making the request, and the ID of the account for which the password is to be changed. The ID must be entered again to reduce the possibility that a typing mistake will cause the wrong ID's PW to be changed. If possible, the address and phone number of the person requesting the change should be taken directly from the identification presented.

*CTD

NAME: *CTD

FUNCTION: To check out the disk drives after preventive maintenance has been performed or after a hard crash, for example, when a storm takes the system down.

USAGE: MTS *CTD

REMARKS: This will do some I/O to each of the MTS disk drives. If a disk error is encountered, no output will be printed. The procedure may get "hung up", or if you're lucky, it will generate an error message.

Note: *CTD shouldn't be run while users are on the sytem since it really hogs the channel.

CT3

NAME: *CT3

FUNCTION: Validates file system integrity and does a cursory check of the message system data structures of each mailbox.

USAGE: mts *ct3

REMARKS: *CT3 is usually used after a "hard" crash. It is started after "peek" completes. *CT3 runs "validatefile" (which itself, validates file system integrity) on *Userdir.Master as well as all of the files that comprise the messagesystem data base. If files are in good working order, the routine starts the message-file mamangers and does a cursory check of the message system data structures of each mailbox. If everything is satisfactory, the routine enables the message-system. If the routine finds something wrong, it leaves the message-system disabled and prints error messages on the console, alerting the operations staff.

*CWM

NAME: *CWM

FUNCTION: Message file purge and validation.

USAGE:mts *cwm

REMARKS: Started by "clockwatcher" program.

If error occurs during running of *CWM, a message with information on making calls to call appropriate persons appears on operators' console.

*DLG

NAME: *DLG

FUNCTION: Program which logs disk errors. This program is started automatically.

USAGE: Is normally started up automatically. Requires only "mts *dlg". Takes no parameters.

*DMP

NAME: *DMP

FUNCTION Initiates deferred *batch* jobs which print dump tapes

USAGE:

```
mts *dmp

ENTER DUMP TAPE # (OR EOF OR STOP):

<operator enters one of these>

ENTER REASON FOR THE DUMP(ONLY ONE LINE):

<operator does this>
```

REMARKS: Whenever a dump finishes, whether it was initiated by an operator or by the system, the program, *DMP should be started.

*DRN

NAME: *DRN

FUNCTION: To issue HASP commands to drain SA00 through SA02 and SA04 through SA19. This is identical to entering the equivalent $DRAIN SAnn commands one at a time. *DRN also drains MNT.RD1, .RD2, and .RD3 as well as MNT.MS, .UM, .WS, .WU, .WM, and .UB.

USAGE: mts *drn

*DSD

NAME: *DSD

FUNCTION: Starts amd runs DASDI

USAGE: mts *dsd

NOTE: See DASDI elsewhere in this manual

*DSK

NOTE: See the section concerning the Disk Manager (DMGR) in this manual for additional disk pack manipulation information.

NAME: *DSK

FUNCTION: To manipulate the table of disks volumes. Disks may be dynamicallyadded and removed, as well as allowing the drive address to be forgotten.

NOTE: This program only handles MTS public volumes. It has no knowledge of SPOOL packs or paging packs.

USAGE: Enter mts *dsk on the operators' console or run sys.:dsk.

The commands are-

ADD MTS xxx
MOUNT MTSxxx

to add MTSxxx to the system. This should be used to add a new MTS volume to a running system when it becomes necessary to expand the available disks space.

CATALOG MTSxxx
NOCATALOG MTSxxx

these commands effect whether or not the file system catalog is permitted to expand onto the designated volume.

DELETE MTSxxx
DISMOUNT
REMOVE MTSxxx

to make the volume "MTSxxx" unavailable to MTS. Doing this a running system will cause all jobs which reference the pack to receive "Hardware error or software inconsistency" messages. This should only be used on a running system in cases of extreme emergency.

DMGR MTSxxx
NODMGR MTSxxx

these commands control whether the Disk Manager is to be used to manage the specified volume. These commands should NOT be used unless the Disk Manager is installed in your version of MTS. Also, if you are using the version of the PDP which pages through the Disk Manager, you should not remove the Disk Manager from any volume which contains a paging extent.

EXPLICIT MTSxxx
NOEXPLICIT MTSxxx

these commands affect whether files will be created on the designated volume. If a volume is designated EXPLICIT, nothing will be created on the volume unless explicitly specified via the VOLUME= parameter on the MTS $CREATE command. NOEXPLICIT removes this designation.

FORGET MTSxxx

to forget the device address of an MTS volume. This causes the file routines to re-initialize the disk table entry for the specified volume on the next reference to it, which is useful if a program such as VAMREC has been used to change information in the volume's PAT which the file routines need to know about (e.g., the pack's relocation entries).

LIST
DISPLAY

to list all currently defined volumes in the table. Useful for determining which volumes are on which drives. This will also summarize the total amount of space and the total amount of free space in the system.

LIST MTSxxx
DISPLAY MTSxxx

to list the DSKTAB information for the volume

"MTSxxx".

POLR MTSxxx
NOPOLR MTSxxx

these commands control whether the volume should be considered one of last resort for the creation of files. If it is, no files will be created on it unless insufficient space is available on the other MTS volumes. NOPOLR resets this designation.

SPACE
AVAILABLE

to summarize the total amount of space and the total amount of free space in the system.

STOP
END
DONE

to end the run. An end-of-file may also be used.

A single command may be specified in the PAR field of the $RUN command.

For those commands that accept a volume name, a list of volume names may also be given, eg.

```
  LIST MTS001 MTS002
```

EXAMPLES:

```
    POLR MTS009
    ADD MTS099
    NODMGR MTS008
```

*DSN

NAME: *DSN

FUNCTION: Starts the user HASP status display at NUBS. This display is a CONRAC monitor driven from the video output of an ANN ARBOR TERMINAL CONTROLLER which is hardwired to an SCP.

USAGE: Is normally started automatically by *MNS. Requires only "mts *dsn". Takes no parameters.

REMARKS: If the user screen seems not to be functioning, e.g., the cursor isn't moving and the time displayed on the screen is behind, the following actions may remedy the situation:

Check and see if there is a job on CB7D. If there is none, start one by typing "mts *dsn". If there is a job present, check with a systems person.

*DSO

NAME: *DSO

FUNCTION: This program prints the total number of disk pages, the number of free disk pages and the percent of free disk space available. You also may initiate *DSK or run sys.:dsk using a LIST.

*DWB

NAME: *DWB

FUNCTION: Program which copies command statistics and unit-check messages from a file to a tape. This program is started automatically by the pickup program when a file fills up.

USAGE: Is normally started up automatically. Requires only "mts *dwb". Takes no parameters.

NOTE: See the CMDSTAT writeup for details and explanations.

*FIX

NAME: *FIX

FUNCTION: Update the accounting record of a user to be correct with regard to files, and to unlock the user's accounting record. To be used if the user had to be BLASTed, or if his task SNARKed. (This is automatically done for all users when the system is initialized after IPL, so any users cut off in mid-air will automatically be fixed.)

USAGE:

```
mts *fix


MTS  SIGNON CSIG  ODP  NDP  ODF  NDF

MTS..0ab2

MTS   0AB2     0   47   54   14   15

MTS..
```

where the last request for input above was terminated by pressing the "cncl" button. For each i.d. fixed it prints the signon id, the number of current signons, the old disk pages, new disk pages, old disk files, and new disk files.

*FLB

Name: *FLB

Function: To determine whether or not a tape is labeled. If labeled, displays label, owner id, and density. If not labeled, message confirms.

Usage: mts *flb

Examples:

```
mts *flb
ENTER TAPE NAME
t910
TAPE NOT LABELED
"eof" if no additional tapes


mts *flb
ENTER TAPE NAME
t908
T908 NOT READY
VOL=CHG186 OWNER=TECUMSEH DENS=1600
ENTER TAPE NAME
"eof" if no additional tapes
```

NOTE:

The present policy permits a staff member to give out tape label information under the following conditions:

- The tape receipt is brought to the Computing Center by the person requesting label information.

- The user on the phone is able to give the name, phone number, and tape id, and be reached at the number given on the rack sticker. After running *FLB, we call the person back (only at the phone number on the sticker) with the label information. If a user other than the owner of the tape is requesting label information and cannot meet the requirments mentioned, e.g., can't be reached at the phone number on the rack sticker, he or she must have the owner call us to request the information. The owner, of course, must be reachable at the phone number on the sticker.

*FSC

```
    Name:          *FSC


    Function:      *FSC is a program designed to make copies  of
                   original filesave tapes.


    Usage:         mts *fsc

    Operator and System interaction:


             IS PRINTED LISTING OF FILES DESIRED? (YES/NO) (SYST.

            "yes" or "no"               (oper)

(If "yes" response is given, system will ask if no tape copy  is
to be made but only a listing)

            IS COPYING OF TAPES DESIRED (YES/NO)

            "yes" or "no"               (oper)

            TYPE STREAM NAME: "MTS00N, "FILES"., OR "ONLINE"

            online       for *SVW              (oper)

            OR

            files.  (When DEADFILES is being run)      (oper)

            Type starting input tape number (syst.)

            "2"  (meaning  mts001,tape  "2")  or  in the case of
            DEADFILES, "1" for the first "original" tape and "2"
            for the second "original" tape, etc.

(The following questions are asked only if a  digit  other  than
"1" was typed in the above command.)

            ARE WE: A) STARTING WITH A NEW OUTPUT TAPE?  (SYST.)

                 OR:  B)  CONTINUING  IN THE MIDDLE OF AN OUTPUT
                        TAPE?  (SYST.)

            ENTER "A" OR "B" FROM ABOVE SECTION           (OPER)
            Type starting output tape number (syst.)

            "1"                            (oper)

            Where is input tape  number  "2"  for  mts001  (ring
            out)   (syst.)

            T90X                       (oper)

            Where  is  output tape number 1 for mts001 (ring in)
            (syst.)

            T90X                       (oper)
```

*FSC is normally terminated with an end-of-file.

Copying of weekly filesave tapes (which produces a list) is usually done in two parts producing two, consecutinve listings. This "six week" filesave copy produces only one continuous list. See "Filesave Schedule and Tape Storage".

Comments: If an error occurs on the output tape, the operator starts the job over and the tapes will space out to where the error occurred for a retry. If an error occurs on an input tape record, *FSC will mention this and copy the error as is. No operator action is necessary.

*FSC will copy labeled input tapes to non-labeled output tapes. If the output tape is labeled incorrectly, the NOCHECK option (T90X NOCHECK) option must be used if one actually does want to change the label. It is not necessary to strip copy tapes before using them since *FSC will back space the output tape if it cannot complete the writing of a record on it and will ask for a new output tape.

NOTE: The original filesave tapes should be kept at the Computing Center for use as "restore" tapes. The copy tapes do not always agree exactly with the listings.

*FSM

NAME: *FSM

FUNCTION: Merge File Save

Usage: mts *fsm

NOTE: *FSM is normally started by *SAV, *SVW or at IPL(*INIT).

*HGS

NAME: *HGS

FUNCTION: Program to run MTA.:HOGS. MTA.:HOGS will look at all jobs running on the ES3090/300J for about thirty seconds to determine which of these jobs are using the most CPU time, VIRTUAL MEMORY, Disk I/O, DRUM READS, REAL pages and which jobs need the most REAL pages. Ten rows of job information will appear under each of the six headings CPU, DSK I/O, VM, DR, and RPN. This job information is the Supervisor#, the CCID or job type i.e., PDP and an amount pertinent to the heading. Under CPU, the first column contains ten supervisor numbers, the second, user ID's or job type, and the third contains amounts of CPU TIME used in the last thirty seconds by each job. Under VM, first column equals supervisor number, second column equals user ID, and third column equals number of virtual pages being used by each job. Under DR, first column equals supervisor number, second column equals user ID, and third column equals number of DRUM READS.

Before and after this list, a line of general system load information will be printed.

The program will print out names of any non-MTS jobs if any appear in its list.

USAGE: mts *hgs

Response (example) on operators console is:

```
----CPU TIME-----  ----DISK I/O-----  --------VM-------
    3    DMGR 3.3    356  IECN   632    536  SQR7   692
  293  K9SK   0.6    313  SPL9   269    583  ID2R---354
Etc.
Etc.
Etc.
Etc.
Etc.
Etc.
Etc.
Etc.


----DRUM READS---  ----REAL PAGES---  -------RPN-------
   76  1ZBK   292   2349  DBMS    93    130  SPDL   107
  536  SQR7   267     54  1N2A    92    216  SPD5    96
Etc.
Etc.
Etc.
Etc.
Etc.
Etc.
Etc.
Etc.
```

*HLB

NAME: *HLB

FUNCTION: Forces all access to the UB host through the "Host Locator". Also can be used to restrict access to the host.

USAGE: MTS *HLB

REMARKS: This program is used for testing by systems people.

The user will be promted as follow:

For Host UB:

Enter YES to force Host Locator access or NO to release system to users.

After the operator enters the appropriate response, a confirmation message will be given.

*HLG

NAME:	*HLG
FUNCTION:	Starts HASP log collection program
USAGE:	mts *hlg
REMARKS:	Normally started automatically by *HSP

*HLM

NAME: *HLM

FUNCTION: Forces all access to the UM host through the "Host Locator". Also can be used to restrict access to the host.

USAGE: MTS *HLM

REMARKS: This program is used for testing by systems people.

The user will be promted as follow:

For Host UM:

Enter YES to force Host Locator access or NO to release system to users.

After the operator enters the appropriate response, a confirmation message will be given.

*HPS

NAME: *HPS

FUNCTION: To display the status of HASP. This program is usually used when it is suspected that HASP is in some sort of trouble. No input is required after the program has been initiated.

USAGE: mts *hps

*HUH

NAME: *HUH

FUNCTION: To determine the current state of all PCPs.

USAGE: mts *HUH

The operator file *HUH runs the file SYS.:HUH. HUH checks each Data Concentrator or UMnet PCP in turn and one of the following messages will be printed:

Message: <name> seems to be running normally.

Meaning: Everything looks good.

Message: <name> is busy - Try again later.

Meaning: This message will appear if the xxOP device is busy when HUH is run. Try *HUH again after a short wait or /t d xxop to see what job is using the device.

Message: <name> is logically offline.

Meaning: Either <name> was not "up" when MTS was IPLed or it has been offlined using the OFFLINE job. In either case to online <name> sign on to the CCID STRT and $SOURCE STRT:ONLINExx, where xx is AB, AD, AE, AF, AN, AU, AV, AW, AX, AY or AZ. Once this is done rerun *HUH.

Message: <name> is physically offline.

Meaning: The online/offline switch is in the offline position. Switch it to online and rerun *HUH.

Message: <name> does not respond - Check and reload if necessary.

Meaning: Either <name> is powered off or a serious error has occurred. If the power is off, it should be turned on and *HUH rerun. If the power is on, then the PCP has probably crashed in such a way that the High Core monitor is not running. To see if this is the case, select <name> at the Super Console terminasl and type HELLO. If there is no response, the machine has crashed and must be restarted from the front panel of the PCP. If it says "<name> monitor", then high-core is running and *xDL should be run. If it says "name> ... PCP MINOS version xxxx" or " ... PCP MCS", then the operating system is running but the channel interface is not operating properly. If the latter happens, do a UNITS D <name> OP S on MTS, and try to cantact a UMnet/UMnet Firefighter. If none are available, restart from the front panel and run *xDL.

Another somewhat unlikely possibility when this message appears is that the control unit, the path from the control unit to the channel, the channel itself, or some combination is logically offline. The UNITS command

```
UNITS  D  xxOP
```

where xx is AA, AB, AD, AE, AN or CC should provide the information necessary to see if this is the case. If it is, the ONLINE job can be used to place things online.

Message: <name> Hicore system is running - Start *xDL.

Meaning: <name> is running but must be dumped and reloaded using *ADL, *BDL, *CDL, or *NDL. Once this is done rerun *HUH.

Message: Unexpected status from <name>, show this output to ....

Meaning: This message should never appear, but if it does it isn't clear just what the problem is or how to fix it. Rerunning *HUH might result in a better message the second time. In any case please show the output to a Communications Group member.

HUH may also be run from an MTS terminal with the following command:

```
$run sys.:huh
```

Output is written on SPRINT and/or SERCOM.

*IG.SHUFFLE

NAME: *IG.SHUFFLE

FUNCTION: This program sources the file GRAF:IG.SHUFFLE and causes subsequent summarization of *IG usage.

USAGE: Normally started by the CLOCKWATCHER program. If it were to be started manually, the procedure would be "mts *ig. shuf fle,".

REMARKS: When this program is run, the *IG statistics are dumped onto tape and the file, GRAF:IG.SHUFFLE is emptied.

*INIT

FUNCTION: When the system is IPLed, after the time and date have been set, the MTS task using *INIT as master source is automatically started by the INIT job program. See the description of the IPL procedure earlier in this manual.

USAGE: Is normally started automatically. If, for some reason, it has to be restarted, enter "mts *ini t,,,". Must NEVER be run once users have been allowed to sign on, as it destroys all scratch files.

*ITX

NAME: *ITX

FUNCTION Starts five additional *-file jobs to exercise and test intertask communication between UMMPS jobs. It is normally used by systems programmers.

*LAR

NAME: *LAR

FUNCTION: Program to reset the line adapters on the transmission controller (Memorex 1270).

USAGE: mts *lar

*LAS

NAME: *LAS

FUNCTION: Program to start up the line adapters and miscellaneous terminals and logging functions.

The results of running *LAS appear on the operators' console as follows:

```
        Line configuration (20 April 1987) - Jobs on:
                Memorex: None started by request;
    Line configuration (25 January 1992) Jobs on:
    Lee Data: DS00, DS02, DS09 & DS0B;
    IBM 3274-1B: DS20 thru DS2E;
    IBM 3274-41D: DS40 thru DS5F;
    IBM 3274-41D: DS80 thru DS8F;
           Other: *MGR, *CLK, *MWT, *TPR,
                  *ORC, *HIM, *MIH, *DSO

                        Other: *CLK, *TPR
```

This is the same as if the operator had typed in the job requests "mts la00", "mts la01",...

USAGE: mts *las

*LBH

Note: *LBH is a program to label tapes that are not to be processed by *LBL (the labelling program for cataloged tapes). Users may use this program by running *Label.

FUNCTION: To label a tape so it can be used by MTS. Any completely blank tape or any tape with an unreadable first record MUST have something written on it by this program before it is attempted to be used in MTS.

USAGE: mts *lbh (The program will prompt for information)

```
    Enter tape name, density, volume name, owner


    t90x 1600 jake icpsr (operator response)
```

The above action wold label the tape on t90x at 1600 bpi using the label 'jake' and the optional owner-id ICPSR.

```
    t90x 6250
```

This action would write a few 6250 bpi file marks on the tape but would not label it.

The operator response for labelling an ANSI tape would be-

```
    t90x 6250 mack engin lbltype=ansi
```

This last example shows-

t90x=tapeunit, 6250=density, mack=label, engin=ownerid, and lbltype=ansi causes the tape to be written in ANSI.

*LBL

NAME: *LBL

FUNCTION: To label a tape so it can be used by MTS. Any completely blank tape or any tape with an unreadable first record MUST have something written on it by this program before it is attempted to be used in MTS.

USAGE: Enter "mts *lbl". The program will prompt for a tape drive and code number.

Drive Name is the tape drive name, eg. T901

Code Name is in the form 123-MIKE-345 (Example)

Tapes to be initialized will have a green sticker with the letter "I" attached to the tape. This sticker should be removed after initialization.

When users submit tapes to be initialized, they should have inserted labelling information into a file. When *LBL is run and the tape and code number are typed in by the operator, the tape is initialized with the information that the user entered into the file mentioned above.

The prompt for input from *LBL is-

```
        ENTER TAPE DRIVE AND RECEIPT CODE.
```

Enter the drive name and the receipt code-

Example-

```
        t904 123-MIKE-456
```

At this point, the tape will be initialized. If the process fails, an error message will be printed on the operators' console.

The output on the console will appear as in the example below-

```
IBM Labelled DENSITY=1600 VOLUME=VOL_2 OWNER=JANE/
```

Operator Errors

In case the tape is mis-labelled (the wrong tape is mounted) the tape must be re-initialized. Tapes cannot be re-initialized using the same code number because once this code has been used, a bit is set in the system to keep the information associated with code number from being re-used.

If the wrong tape was labelled, the information associated with the mis-labelled tape is printed on the console (shown in the example above) as is true with correctly labelled tapes. This information can be used to correct the situation.

To re-lable a mistakenly initialized tape, do the following-

Mount the correct tape and-

```
$signon sys.

$run *label
```

Enter the tape information from the console just as it was printed by *LBL. In the example this was-

```
IBM Labelled DENSITY=1600 VOLUME=VOL_2 OWNER=JANE/
```

so when you run *LABEL, you type in-

```
t90x 1600 vol_2 jane/

$endfile
```

The tape which was mis-labelled can be relabelled using *LBL since the information associated with its code number hasn't been used yet.

ANSI Labelled Tapes

```
t90x 1600 vol_2 Jane/ lbltype=ansi
```

*LIM

NAME: *LIM

FUNCTION: To display (in minutes and hundredths of minutes) the CPU time remaining to a batch job.

USAGE: Enter "mts *lim" at the console. When it prompts for input, enter either the MTS task number of the batch job in question or the user id. for that batch job. Output takes the form:

```
     JOB 00302 (W035):   3.23
```

which means that job 302 (user id. W035) has 3.23 _minutes_ of CPU time left before his next limit expires. Note that this will be a local limit if he has one set, i.e., he will not necessarily get "global time limit exceeded" at the end of that time.

If the job has no limit set (e.g. a terminal job), you will get the message:

```
     JOB 00427 (MTA.): NOLIMIT
```

If you supply an invalid task number:

```
     JOB 99999 (    ): ?????
```

If you supply an invalid user id:

```
     JOB 00000 (BARF): ?????
```

*LIM will come back and prompt for another line of input until terminated by an end-of-file.

*LMT

Name: *LMT

Function: Same as *MSS except more limited. Lists no more than last four instances of tape usage.

Usage: mts *lmt

NOTE: Before using, see *MSS.

*LRL

Name: *LRL

Function: Gives information about last IPL, i.e., time in hours, minutes and seconds since last IPL as well as the reason, date and time for the reload.

USAGE: mts *lrl

*MNS

NAME: *MNS

FUNCTION: To start-up UMnet tasks on the AA, AB, AD, AE, and AN PCPs after IPL.

USAGE: mts *mns

Response on operator's console is :

```
MTS  UMnet (AZ) Configuration (6 April 1992)
MTS  Jobs on: AZ10 through AZ6F
MTS  UMnet (AY) Configuration (6 April 1992)
MTS  Jobs on: AY10 through AY6F
MTS  UMnet (AX) Configuration (6 April 1992)
MTS  Jobs on: AX10 through AX6F
MTS  UMnet (AW) Configuration (6 April 1992)
MTS  Jobs on: AW10 through AW6F
MTS  UMnet (AV) Configuration (6 April 1992)
MTS  Jobs on: AV10 through AV6F
MTS  UMnet (AU) Configuration (6 April 1992)
MTS  Jobs on: AU10 through AU6F
MTS  UMnet (AB) Configuration (6 April 1992)
MTS  Jobs on: AB10 through AB3F
MTS  UMnet (AD) Configuration (6 April 1992)
MTS  Jobs on: AD10 through AD3F
MTS  UMnet (AE) Configuration (6 April 1992)
MTS  Jobs on: AE10 through AE3F
MTS  UMnet (AF) Configuration (6 April 1992)
MTS  Jobs on: AF10 through AF3F
MTS  UMnet (AN) Configuration (22 April 1992)
MTS  Jobs on: AN10 through AN17
```

(or current equivalent).

*MSD

NAME: *MSD

FUNCTION: To set the time and date in all PCPs on the UMnet network.

USAGE: mts *msd

NOTE: This job is normally run automatically by *CLK at midnight.

*MSG

NAME: *MSG

FUNCTION: Cleans up $Message System, particularly, *MESSAGES.

USAGE: mts *msg (See REMARKS)

REMARKS: This program is started up each morning at 2am by the "clockwatcher" program. It should never be started manually. A log is kept in the file MAIL:MSG.LOG.

MSS

Name: *MSS

Function: Provides lists of, and important information about, tape mount instances.

Usage: mts *mss , oper

NOTE: In place of OPER, a printer (>PTRx) may be used, x being the number of an available printer.

Example:

```
    ENTER: "R" RACK NO.  STARTING DATE (AND ENDING DATE)
       OR: "U" USERID STARTING DATE (AND ENDING DATE)
       OR: "ALL"

R C1234 02-05-76 03-10-76  (operator)
U WSG.  03-20-76
U W035
R P10#



RACK    DATE     TIME   USER PROJ  JOB# TYPE UNIT MODE STAT FLAGS

C1234 02-23-76 13:45:32 K1F0          9TP T903        OK  LP
DUMP 03-01-
76 21:32:76 W035            9TP  T902       OK  RING,LP
DUMP 03-03-
76 17:31:39 W035            9TP  T900       OK  RING,LP
DUMP 03-08-
76 18:15:47 W035            9TP  T903       OK  RING,LP
DUMP 03-17-76 17:23:51 W035 WOPR 01519 9TP T900 160 OK RING,LP
DUMP 03-18-76 19:38:32 W035 WOPR 01870 9TP T901 160 OK RING,LP

STAT 03-01-76 07:53:41 W035          9TP T901      OK  RING,LP
```

Continuation of Examples

```
S0011 08-30-73 18:30:06 W035         9TP  T909       OK  NL

S011 09-17-73 21:51:32               9TP  T908       OK  NL

S11 01-13-75 21:56:54 W035           9TP  T900       OK  NL

S188A 04-13-72 22:01:02 W035         7TP  T700     OK  NL,RING

S188B 04-13-72 22:06:58 W035        7TP  T700     OK  NL,RING

S417 02-23-73 19:58:33 W035        9TP  T900      OK  NL,RING

YYY  02-12-76 18:14:10 W035        9TP  T900      OK  NL,RING

P100  08-18-75 16:00:20 SDDZ             PTPR PTPR   OK

P101  08-18-75 16:02:22 SDDZ             PTPR PTPR   OK

P102  09-02-75 23:19:02 K2LE             PTPR PTPR   OK

P103  01-10-76 21:01:20 SBLM             PTPR PTPR   OK
# EXECUTION TERMINATED
```

NOTE: Up to six "#"s can be used in the rack number section. Where there is a "#" there is a digit or letter being ignored. All instances of that particular position will be printed. For instance, typing six "#"s would be the same as typing "ALL". Typing c12## would give a list of all tapes beginning with "C12".

Heading Explanations:

RACK=Rack Number DATE=Date of Mount

TIME=Time of Mount USER=User Id

PROJ=Project Number JOB#=Supervisor Number

TYPE=Type of device tape was mounted on

UNIT=The particular device that the tape was mounted on

MODE=Bits-Per-Inch STAT=Operator Response

FLAGS=Description of Tape, e.g. whether or not the tape was labeled, had a ring in it, etc.

MSSCN From a Terminal

Msscn can be run from a terminal as well as from the operators' console. Its proper name is sys.:msscn and scards is usually sys.:mountstat although it can be a tape, either s0820 or s1091.

If scards a tape, which at this writing would be either s0820 or s1091, you will want to position the tape according to the dates that you are concerned with. Firs, mount the tape and then "$cont *j* posn=*eot*". After you have done this, then "$cont *j* bsr 9999". You will get an error indication the first time that you back-space but just do it again. After you backed up the tape as shown above a cuople of times, you'll want to see where you are on the tape. To do this-

```
    $run msscn scards=*j*(1,3)

    all
```

The above action will print out three lines containing the date and time (as well as other information). This will help you to decide whether to re-position the tape before running msscn.

Example Msscn Run

```
    $run sys.:msscn scards=sys.:mountstat sprint=-o
```

The program will then wait for you to enter rack numbers, CCIDs, project numbers or "all" followed by an $endfile.

u abcd — (this will list all tapes mounted by CCID abcd, that is, the last mount and the last mount with the ring in)

r n1234a — (this will print the last time(s) that this tape was mounted as well as the last time with the ring in)

r c30### — The # acts as a wild card, that is, it will look for and print all numbers and letters that are in its position, in this case (c30###) it will do from c3000 through c30999z. The # can be used in any or all positions, e.g., n#2#4# or ######. Using ###### would be the same as the "all" parameter.

Example:

If one wanted to pick up all tapes in the range c3650 to c4300 and all tapes used by CCID scd4, they should do the following:

```
    $run sys.:msscn scards=sys.:mountstat sprint=-o
    r c365##
    r c37###
    r c38###
    r c39###
    r c40###
    r c41###
    r c42###
    r c4300#
    u scd4
    $endfile
```

<u>or</u>,

you could:

```
    r c3####
    r c4####
```

and then delete the lines in -ooo (after you run the sort which follows) that you didn't need.

After running sys.:msscn, if you expect more than a few output lines, you should run a program which cleans up the file, that is, removes the extra title lines and leaves only the last mount statistic for each tape in the file.
This "clean-up" program must be compiled before it can be used. The programs as it is, is set up to take care of "C" tapes only.

```
$run *ftn scards=w225:w32.tpst(,57)
                                        (This compiles the
                                        program and puts the
                                        output (spunch) in the
                                        file -LOAD)
$run -load 5=-o 6=-oo
                                        (This actually cleans up
                                        the file and puts the
                                        output in file -oo)
```

The followint "sort" run will sort the tapes by rack number and then by date.

<u>Note</u>: In order for the sort program to run most efficiently, the number of records to be sorted should be determined and then placed in the record field at the end of the sort run. To determine the number of records to be sorted in this case-

```
$f -oo lines
```

if the number of lines is, for example, 12,154, then the record field should reflect this. (See followint example.)

```
$run *sort par=sort=ch,a,2,6,ch,a,15,2,ch,a,9,6,ch,a,18,8
                   i=-oo o=-ooo r=12,154
```

If you want the tapes sorted by date only, which show you which tapes haven't been mounted recently, do the following-

```
$r *sort par=sort=ch,a,15,2,ch,a,9,6 i=-oo o=-ooo r=12,154
```

Following is a break down of the sort command:

"ch" = a character sort, "a" = ascending order, 2,6 = begin in col. 2 and sort 6 columns in from there, in this case, the rack number. If you look in the file -o after running msscn, you can see what is sorted by counting in from col. 1. Columns 15,2 and 9,6 get the date and 18,8, the time. i=-oo means -oo is the input file and o=-ooo means that -ooo is the output file and r= is the number of records to be sorted.

Since you will probably want a double spaced listing of your output, do the following:

```
  $edit -ooo

  col 1 1                        (this will confine any edit
                                 commands to column 1)

  a@a@nv@opl /f ' '0'           (this changes any blanks in
                                 col.  1 to "0"s.  "0"s in col.
                                 1 will cause the lines to be
                                 double spaced)
```

NOTE: The above example will only list "C" tapes since the "clean-up" program was set up that way. To get a listing with just "N" tapes or with "N" and "C" tapes, one must edit w225:w32.tpst and change line 2 from INTEGER CEE/'C'/,OK/'OK'/ to INTEGER CEE/'N'/,OK/'OK'/ and then compiled specifying an spunch, for example-

```
$copy w225:w32.tpst(,57) -w32

$edit -w32

a 2 .'C'.'N'.

stop  (gets you out of the editor)

    $run *ftn scards=-w32 spunch=-load.1
    $run -load.1 5=-o 6=-oo(last+1)
```

The "(last+1)" adds the "N" tapes statistcs to the end of file -oo which contains the "C" tape statistics.

Now you can "$run sort" as shown above.

Next, to print out your listing. To do this:

```
    $run *pagepr scards=-ooo par=format=landscape,onesided
```

or, if you want a line printer listing,

```
    $cont *print* printer=line
    $copy -ooo *print*
    $release *print*
```

Below, is an example of a procedure for compiling a list of "last mount" statistics for all "N" and "C" tapes on both the UM and UB systems. This list would be used to determine which tapes should be pulled (due to their lack of usage) and sent to storage elsewhere.

```
    $sig w035 (UM System)
    $run sys.:msscn scards=sys.:mountstat sprint=-o
      Enter: R rackno etc.
       or    U userid etc.
       or    P projno etc.
       or    ALL
    r c#####       (gets all "C" tapes)
    r n#####       (gets all "N" tapes)
    $endfile
    $run *ftn scards=w225:w32.tpst(,57)
                                 (This ($run *ftn, etc.)
                                 compiles the "cleanup" program
                                 for "C" tapes.  The "cleanup"
                                 program extracts the last mount
                                 statistic for each tape from
                                 the output of msscn which is in
                                 -o, and gets rid of extra
                                 titles, etc.  The object from
                                 this fortran run automatically
                                 goes into -LOAD.)
    $run -load 5=-o 6=-oo
```

Now, you need to do the same thing for the "N" tapes

```
    $copy w225:w32.tpst(,57) -w32
                                 (This copies the first 57 lines
                                 of w225:w32.tpst to file -w32)
    $edit -w32
    p 1 3          (prints 1st 3 lines of -w32)
```

Line 2 should appear as-

```
    INTEGER CEE /'C'/,OK/'OK'/
```

To change this line so that the "cleanup" program looks for "N" tapes-

```
    a 2 :'C':'N':
    p 2
```

Now, line 2 of -w32 should appear as-

```
    INTEGER CEE /'N'/,OK/'OK'/

    stop           (gets you out of the editor)
```

To compile this program-

```
    $run *ftn scards=-w32 spunch=-load.1
                                 (In the example above, spunch
                                 is specified and has a
                                 different file name; otherwise
                                 the object code would
                                 automatically go to -load which
                                 aleady contains the program
                                 looking for "C" tapes.)
    $run -load.1 5=-o 6=-oo(last+1)
                                 (This runs the "cleanup" job
                                 for the "N" tapes and places
                                 the ouput records at the end of
                                 -oo)
```

Now, the same procedure must be done on the UB system.

```
    net ub
    $sig w035
    $run sys.:msscn scards=sys.:mountstat sprint=-o.b
```

```
  r c#####
  r n#####
$endfile
$run *ftn scards=w225:w32.tpst(,57)
$run -load 5=-o.b 6=-oo.b
$copy w225:w32.tpst(,57) -w32
$edit -w32
a 2 :'C':'N':
stop
$run *ftn scards=-w32 spunch=-load.1
$run -load.1 5=-o.b 6=-oo.b(last+1)
```

Now, to combine the UB file (-oo.b) with the UM (-oo) file, the UB file must be copied to the end of the file on the UM system.

```
.copy rmt -oo.b lcl -oo(last+1)
```

This procedure should take a minute or so.

To get back to the UM system-

```
.mts
```

At this point, the "C" and "N" tapes from both systems should be in the file -oo. The next thing to do is to sort the tape list. The following example sorts the tapes by date.

```
$run *sort par=sort=ch,a,15,2,ch,a,9,6 i=-oo o=-ooo
```

Finally, to get a double spaced listing, all of the blanks in column 1 must be changed to zeros, zero being the carriage control character for skipping a line.

```
$edit -ooo
col 1 1
                             (this confines editing to
                             column 1)
a@a@nv@opl /f ' '0'
                             (nv means "no verification" and
                             opl means "once per line")
stop
```

To print the list on the line printer-

```
$cont *print* printer=line
$copy -ooo *print*
$release *print*
```

To print the list on a page printer-

```
$run *pagepr scards=-ooo
```

or for onesided pages-

```
$run *pagepr scards=-ooo par=onesided
```

*MUP

NAME: *MUP

FUNCTION: To determine the current state of the UMnet Computer Network.

USAGE: mts *mup

The operator file *MUP runs the file MNOP:MUP. MUP first checks the file MNET:AUTH and if problems are found one of the following messages is printed:

Message: Outbound MTS access to UMnet is restricted from HH:MM to HH:MM.

Meaning: MTS users are not able to MOUNT outbound network connections during the period specified. This would normally be the case when either hardware or software testing on the UMnet PCP is necessary. Running *MNS will remove this restriction.

Message: Outbound MTS access to UMnet will be restricted from HH:MM to HH:MM.

Meaning: MTS users will not be able to MOUNT outbound network connections after the time specified. This would normally be the case when hardware or software testing on the UMnet PCP is scheduled. Running *MNS will remove this restriction.

Message: Error while accessing MNET:AUTH.

Meaning: A call to GETFD failed or a return code greater than 4 was returned by READ while accessing MNET:AUTH. If this message appears, show it to a member of the UMnet staff.

Message: Unable to lock MNET:AUTH.

Meaning: The file MNET:AUTH is locked for MODIFY or DESTROY. Rerun MUP in a few minutes and if the file is still locked try to determine what job has it locked using the MTS $LOCKSTATUS command or the TABLMOD job and BROADCST a message to this job.

Message: MNET:AUTH is not permitted read others.

Meaning: MNET:AUTH is not correctly permitted, let a member of the Communications Group know.

Message: MNET:AUTH is incorrectly formatted.

Meaning: One or both of the first two lines in MNET:AUTH is missing, running *MNS should fix it up.

Next MUP checks the state of the UMnet PCP and one of the following messages will be printed:

Message: <name> seems to be running normally.

Meaning: Everything looks good.

Message: <name> is busy - Try again later.

Meaning: This message will appear if the xxOP device is busy when HUH is run. Try *HUH again after a short wait.

Message: <name> is logically offline.

Meaning: Either <name> was not "up" when MTS was IPLed or it has been offlined using the OFFLINE job. In either case to online <name> sign on to the CCID

STRT and $SOURCE STRT:ONLINExx, where xx is AA, AB, AD, AE, AN, or CC. Once this is done rerun *HUH.

Message: <name> is physically offline.

Meaning: The online/offline switch is in the offline position. Switch it to online and rerun *HUH.

Message: <name> does not respond - Check and reload if necessary.

Meaning: Either <name> is powered off or a serious error has occurred. If the power is off, it should be turned on and *HUH rerun. If the power is on, then the PCP has probably crashed in such a way that the High Core monitor is not running. To see if this is the case, select <name> at the Super Console terminals and type HELLO. If there is no response, the machine has crashed and must be restarted from the front panel of the PCP. If it says "<name> monitor", then high-core is running and *xDL should be run. If it says "name> ... PCP MINOS version xxxx" or " ... PCP MCS", then the operating system is running but the channel interface is not operating properly. If the latter happens, do a UNITS D <name> OP S on MTS, and try to cantact a UMnet/UMnet Firefighter. If none are available, restart from the front panel and run *xDL.

Another somewhat unlikely possibility when this message appears is that the control unit, the path from the control unit to the channel, the channel itself, or some combination is logically offline. The UNITS command

```
UNITS  D  xxOP
```

where xx is CA, CB, CC or MN, should provide the information necessary to see if this is the case. If it is, the ONLINE job can be used to place things online.

Message: <name> Hicore system is running - Start *xDL.

Meaning: <name> is running but must be dumped and reloaded using *ADL, *BDL, *CDL, or *NDL. Once this is done rerun *HUH.

Message: Unexpected status from <name>, show this output to ....

Meaning: This message should never appear, but if it does it isn't clear just what the problem is or how to fix it. Rerunning *HUH might result in a better message the second time. In any case please show the output to a Communications Group member.

Next, if the UMnet PCP was operating normally, MUP will count the number of UMMPS jobs using the network and print the following message:

```
aa Idle Inbound, bb Active Inbound, cc Active Outbound, dd
Hasplings
```

where: aa is the number of free MTS jobs serving the network.

bb is the number of MTS jobs serving inbound network users.

cc is the number of outbound connections mounted from MTS jobs.

dd is the number of HASPLINGS acting as network batch transmitters and receivers.

Next one of the following messages will be printed if anything out of the ordinary appears in the totals:

Message: There are no inbound MTS jobs serving the UMnet network.

Meaning: Either *MNS was not run, it was run but the UMnet PCP was not "available" or SHUTDOWN has been run. Running *MNS should start the necessary MTS jobs.

Message: All inbound MTS jobs serving the UMnet network are in use.

Meaning: This is not an error, but any additional users who try to access UofM through UMnet will get the all interactive ports busy message.

Finally, if the PCP is operating normally and outbound access to the network is currently restricted by MNET:AUTH (see above), then MUP will check the state of each host on the network and print one or two explanatory messages if they are <u>not</u> operating normally.

MUP may be run from any MTS terminal using the following command:

```
    $run mnop:mup
```

Ouput is written to SPRINT and/or SERCOM.

PAR=WAIT may be given or the RUN command to request a fifteen second wait before MUP starts its checking. This is useful in *xDL just after the PCP has been reloaded, but before it's had a chance to respond properly.

PAR=SHORT may be given and MUP will only check MNET:AUTH and the current status of the PCP. It won't count the UMnet jobs or attempt to open connections to each UMnet host. The string PAR=MED may be given which will cause MUP to check MNET:AUTH, the status of the PCP, and to count the UMnet jobs but won't attempt to open connections to each UMnet host.

The strings PAR=WAIT,SHORT, or PAR=WAIT,MED may also be used.

MSV

Name: *MSV

There is a new (07/08/92) operator file, *MSV (Messagesystem Save) that will be run from *MDP at the end of the UM full file save.

Function: To back up to tape the *MESSAGES.OLD? files created by *MDP. To do this, it uses the Deadfiles DEADSAVE program. *MSV will be automatically started when *MDP completes. It will announce which series of tapes it expects, and then issue prompts.

Usage: What tape drive can I use?

Respond with a four-character cartridge tape drive (e.g. TC04). Load up the tapes into the autoloader. After you tell it which drive to use, it will ask you to mount the first tape in the series:

Please mount FSxxxx on TCxx.

Assuming that the tape has been mounted, just press the Enter key. If the tape has never been used, *MSV will prompt:

OK to label tape on TCxx? (Y/N)

Assuming that you've got the correct tape there (and not some poor user's unlabeled tape), type Y and press the Enter key. If the tape has been used previously, *MSV will prompt:

OK to re-use tape on TCxx? (Y/N) to which you should respond by typing Y and pressing the Enter key.

The "Please mount ..." and "OK to ..." sequence will be repeated for each tape that *MSV needs to use (currently three cartridges, but this will increase). In each case, you should be able to just press the Enter key in response to the "Please mount ..." prompt, and type Y and press the Enter key in response to the "OK to ..." prompt.

Remarks:The tapes used by *MSV come in four series which will be cycled through. The tape series are numbered 9000-9099, 9100-9199, 9200-9299, and 9300-9399. There is an initial set of five tapes for each series in the tape racks with the UM Dump tapes (on the left side of the isle, near the top). They're marked as the mail backup tapes.

If you have problems with this, please get in touch with either Steve Rothwell (3-4902) or Steve Burling (3-4905).

NST

NAME: *NST

FUNCTION: Test program, manipultes "Name Address Spaces"

USAGE: mts *nst

REMARKS: The purpose of *NST is to allow the NAS test program to be run from the console. The NAS test program allows manipulation of Named Address Spaces, into which commonly used programs are loaded. Using the NAS test program, one can obtain information about the system's usage of NASes and modify certain aspects of NAS processing.

Note: This program is intended to be a debugging tool for use by systems programmers.

*NDL

NAME: *NDL

FUNCTION: Dumps and reloads Primary Communications Processor AN (PCP)

USAGE: mts *ndl

NOTES: The High-Core monitor must be running in Primary Communications Processor (PCP) AN before using *NDL. To start the monitor, follow instructions on the front panel of PCP AN (PCP).

After successfully starting the PCP system, the following question will appear on the console:

```
ENTER REASON FOR RELOADING PCP:
```

The operators' initials are typed in followed by a space and a brief explanation.

If problems are experinced during the day, contact the Computer Communications Group or the MERIT office, or if at night, call one of the current UMnet system firefighters.

*NLO

NAME: *NLO

FUNCTION: Loads Primary Communications Processor AN

USAGE: mts *nlo

NOTE: No dump occurs when this is run.

<u>*OOT</u>

NAME *OOT

USAGE: mts *oot

FUNCTION: Automatically starts NTM and offlines all tape drives on the UB system at 10 am on weekdays. At 4pm, *OOT removes NTM and onlines t906 and t907 on the UB system and offlines them tho the UM system. One can online more drives on the UB or UM system if necessary.

*PAG

NAME: *PAG

FUNCTION: Runs the program MTA.:PAGLOAD which creates an artificial paging load on the system. This program is useful for testing the PDP and its paging devices.

USAGE: mts *pag

REMARKS: This job will continue to run until GOOSEd.

*PLB

NAME: *PLB

FUNCTION: This program prints identifying information about plot tapes, i.e., the name, e.g., "TUE 1000", when it was last written, and the minimum time to re-use.

USAGE: To run from the Operators' console, enter:

```
       mts *plb
```

The program will print

```
       ENTER TAPE NAME:
```

Respond with the name of a drive on which a plot tape is mounted, e.g.,

```
       T911
```

The program will print several lines of tape information and then prompt for another tape name. Enter an end-of-file (CNCL) to terminate the program.

EXAMPLE:

```
mts *plb
ENTER TAPE NAME:
t911
TAPE "MON 2000" LAST USED 11-06-78 22:00:11
MIN REUSE TIME: 6 DAYS 0 HRS 0MIN.
NEW PLOTS REPLACE OLD.
ENTER TAPE NAME:
```

When running the program from somewhere other than at the Operators' console:

```
       $run sys.:pident
```

When promted for a tape name, enter either the name of the drive, e.g., t901 or, a pseudo-device name, e.g., *tape*. The program reads tape names from GUSER and prints its output on SERCOM.

ERROR MESSAGES:

"NOT A TAPE"

The tape name entered was not the name of a drive (or proper pseudo-device name) or, the drive was busy.

"NOT A PLOT TAPE"

either the volume name was not "plot" or, the name of the first data set was not "IDENT".

"FORMAT ERROR ON PLOT TAPE"

The tape was a plot tape but some error was found in the identifying information in the first file on the tape. The tape should be re-initialized before being written by *CCP.

*PLQ

NAME: *PLQ

FUNCTION: Runs PLOTQUIK

USAGE: mts *plq

REMARKS: This program plots which have been queued by the user with PAR=ASAP. The program was designed to let people get their plots out sooner than they would by queueing up for the normal plot times. Currently the QUICK plots are done in ball point. For information on how to use this program, see the PLOTTING section later in this manual.

*PWC

NAME: *PWC

FUNCTION: Prints WAIT COUNTS for the system

USAGE: mts *pwc

COMMENTS: *PWC runs MTS.:PWC and directs the output to the operators' console. The job will stop automatically or it may be stopped via STOP or GOOSE.

OUTPUT:

```
Totals__ Ready______ Page Wait__ DASD wait__ 3270 wt_ Intertask__
- Other______ MISC___________________  MON MAY 25/81
To Sk ?W To Nu Pr |34P To Nu Pr |34P To Io Gu Fi To Io Gu To Ry By Wt To Io
- Ti ?? To Do Js Pk Gu Lk Op  Hs
52 48  4  .  .  .  .  .  .  .  .  2  2  .  . 10  9  1  .  .  .  . 33 29
- 4  .  3  .  2  1  .  .  .  .  15:20:02
52 48  4  .  .  .  .  .  .  .  .  1  1  .  . 10  9  1  .  .  .  . 34 30
- 4  .  3  .  2  1  .  .  .  .  15:20:05
  e 52 48  4  .  .  .  .  .  .  .  .  2  2  .  . 10  9  1  .  .  .  . 34 30
- 4  .  3  .  2  1  .  .  .  .  15:20:05
  m 52 48  4  .  .  .  .  .  .  .  .  2  2  .  . 10  9  1  .  .  .  . 34 30
```

Lines that start with "e" are the extremes, with "m", the means, and with a blank, the currnet count.

```
Totals__

To Sk ?W
|  |  |------State unknown
|  |---------State known
|------------Grand Total

Ready______
To Nu Pr |34P
|  |  |   |--Non-Priv. jobs ready for a CPU
|  |  |------Priv. jobs ready for a CPU
|  |---------Neutral jobs ready for a CPU
|------------Total jobs ready for a CPU

Page wait__
To Nu Pr |34P
|  |  |  |---Non-priv. jobs waiting for a page-in
|  |  |------Priv. jobs waiting for a page-in
|  |---------Neutral jobs waiting for a page-in
|------------Total jobs waiting for a page-in

DASD wait__
To Io Gu Fi
|  |  |  |---Jobs waiting to lock a file
|  |  |------Jobs waiting to obtain a disk unit (SVC GTUNIT)
|  |---------Jobs waiting for disk I/O (with or without DMGR)
|------------Total jobs waiting for disk or files

3270 wt_
To Io Gu
|  |  |------3270 jobs waiting to obtain the device(GRAB/FLIP)
|  |---------Jobs waiting for I/O to a 3270 display
|------------Total jobs waiting on 3270s

Intertask__
To Ry By Wt
|  |  |  |---Information on the use of UBC intertask communication
             (at UM only the DMGRSTAT job uses this currently).

Other______
To Io Ti ??
|  |  |  |---Unclassified jobs (PWC doesn't know about new TASKSTAT
- codes
|  |  |------Jobs in time wait
|  |---------Jobs waiting for I/O to something other than a 3270 or a
- disk
|------------Total jobs in the "other" group

MISC____________________
To Do Js Pk Gu Lk Op Hs
|  |  |  |  |  |  |  |--Jobs waiting for HASP
|  |  |  |  |  |  |-----Jobs waiting to read or writ the console
- (SVCs read and write)
|  |  |  |  |  |--------Jobs waiting to obtain a supervisor lock (SVC LOCK)
|  |  |  |  |-----------Jobs waiting to obtain a unit other than a disk
- or 3270 (SVC GTUNIT)
|  |  |  |--------------Jobs in CMDSTAT wait
|  |  |-----------------Jobs waiting for the systemstatus master
                                                   job - (SSRTN)
|  |--------------------Dormant jobs (waiting for an interrupt)
|-----------------------Total jobs in the "MISC." group
```

*P16

NAME: *P16

FUCTION Starts 16 *PAG jobs

USAGE: mts *p16

*RCP

NAME: *RCP

FUNCTION: To produce MTS S8 cards.

FEATURES: Produces receipt and deck or just deck S8 cards. Start and end number set at run time.

LIMITATIONS: Punch must be drained.

USAGE:

1. Make sure you have a free punch. E.g., use a free punch or '$drain pchx'
2. Put blank S8 cards in hopper and ready punch
3. At console type 'MTS *RCP'
4. Program will ask for:
   (a) DO YOU WANT RECEIPT CARD=YES;DECK CARDS=NO
   (b) THE PUNCH TO USE \\ i.e., pchx
   (c) A SIX DIGIT STARTING NUMBER i.e. 110001
   (d) A SIX DIGIT ENDING NUMBER \\ i.e. 110500

If you ask for 500 cards you will get:

1. 500 deck cards in ascending order.
2. 500 each of receipt and deck cards in ascending order.

ERROR MESSAGES:

PUNCH IS BUSY, NOT OPERATIONAL, OR NOT A PUNCH.
RESPONSE WAS NOT YES OR NO.
NUMBER JUST ENTERED CONTAINS ILLEGAL CHARACTERS OR IS NOT SIX DIGITS LONG.
FIRST NUMBER IS LARGER THAN SECOND.

EXAMPLE:

```
$drain pchx
mts *rcp
MTS MTS RECEIPT CARD PROGRAM
MTS ENTER THE NAME OF THE PUNCH
MTS..>pchx
MTS ENTER YES IF RECEIPT CARDS WANTED, NO IF
ONLY DECK CARDS WANTED
MTS..no
MTS ENTER SIX DIGIT STARTING RECEIPT NUMBER
MTS..300000
MTS ENTER SIX DIGIT ENDING RECEIPT NUMBER
MTS..300010
```

```
MTS ALL RECEIPT CARDS HAVE BEEN PUNCHED
```

*RES

NAME: *RES

Purpose: To display the "restore status" i.e., the dates and times the file was saved, and/or to actually restore the file(s).

Note: Users can display and restore files using the *RESTORE program. However, if a file is to be restored from one ID to another, *RES must be run by an Operator.

Usage: The program is invoked by issuing "mts *res". When execution begins, *RES expects the operator to enter a command that indicates which of several services is desired. For example, the DISPLAY command may be used to determine which versions of a file are available for restoration. The RESTORE command may be used to indicate the name of a file to be restored. Several successive RESTORE commands may be entered; the file names are queued and then the files are restored after the STOP command is given. The queue may be emptied with the CANCEL command.

After the "stop" command is entered, the program automatically requests that an operator mount the proper tape(s) and searches the tapes for the proper files. If a tape drive is unavailable or if the desired tape is in use, a message will indicate which files were not restored. Since tape mounting and searching are somewhat time consuming, the resore operation may take several minutes. An attention interrupt (goose) aborts the running of the program.

Comments: *RES does not check to determine whether files queued for restoration will fit in the disk space allocated to the user's CCID. Restore operations are aborted if the user's file space limit is exceeded while a file is being restored.

The save tapes used by *RES only contain files which have existed during the most recent past (about six weeks). However, the Computing Center maintains a separate set of tapes to provide file restoring service for older files that are not available through *RES. These older tapes are used with the program *RST.

Commands: The following commands are available for *RES. The underlined portion of each command may be used as an abbreviation.

```
RESTORE usid:fname [mm/dd/yy] [hh:mm:ss] [NEWNAME=usid:newfname]
    or
RESTORE usid:? [mm/dd/yy] [hh:mm:ss] [NEWNAME=usid:?]
    or
RESTORE any filename pattern, e.g., usid:c3m?, usid:?.old, usid:?x?
```

In the above examples, "usid:filename" would be specified as w000:file, for example. If only the file name is specified, then the most recent version available of that file is queued for restoration. The file will be restored with the same name unless a NEWNAME is given. If "res usid:?" is specified, all files under this USER ID will be restored. If "res usid:? newname=usid:?" is used, all files from the first USER ID will be restored to the second. If a patterned filename such as "res w035:c3m? is used, all files owned by CCID w035 and beginning with the characters c3m will be restored.

If a date (or date and time) is specified, then the most recent version saved on or before the date/time specified is queued for restoration. It should be remembered that since daily filesave only saves files that have been changed since the last filesave, specifiying a date actually amounts to requesting that a file be restored to its status as of that date.

```
DISPLAY {any filename pattern } [mm/dd/yy] [hh:mm:ss]
[HISTORY={ALL | x}]
```

If usid:fname is specified, then the most recent version of "usid:fname" available for restoring is listed

If only the date and time are given on the display command, then only versions on or before that date are listed.

If a patterned file name is specified such as "usid:xxx?" is given, the latest version of each file available for restoring is listed along with the comment:

"There are x older versions of this file". If the ? is not used then the complete history is printed. If the complete history of each file is wanted and when an unknown (?) is used, then "history=all" or "history=x" (x=digit for number of history lines) must be appended to the display command.

If, for example, Display usid:c3m? is used, then all files under "usid" beginning with the characters "c3m" will be displayed.

QDISPLAY will list all of the files queued to be restored, along with their new names if applicable. (This is useful for determining what *RES thinks it's going to do if there is some uncertainty.

QPROCESS Restores all files queued for restoration.

STOP Terminates command input, restores all files queued for restoration, and then terminates execution of the program.

<end of file> (or CNCL key on Operators' console)
Same as STOP command.

<zero length line>
Ignored.

CANCEL Removes all files queued for restoration by any previous RESTORE commands.

<attention>=(goose)
If STOP has not been issued yet, an attention interrupt terminates the current command and requests another. If STOP has been issued (while the program is in the process of restoring files), no more files are restored, a list of unrestored files is printed, and the program terminates.

The following is a sample session using *RES (Operator input shown in lower case):

```
mts *res
Type Command
dis w035:video
1 w035:video saved tue 08/09/77 04:55:44 on tape 1308 ver. 3
2     "      saved sun 07/31/77 10:43:31 on tape 1208 ver. 3
3     "      saved sun 07/24/77 09:07:18 on tape 1105 ver. 2
4     "      saved sun 07/17/77 08:40:42 on tape 1008 ver. 2
```

```
  5     "      saved sun 07/10/77 10:48:07 on tape 1505 ver. 1
  6     "      saved sun 07/03/77 10:30:24 on tape 1408 ver. 2
  Type Command
 res w035:video 07/10/77 newname=w000:video1
  1 W035:VIDEO SAVED SUN 07/10/77 10:48:07 ON TAPE 1505 VER. 1
  W035:VIDEO has been queued for restoration
  Type Command
  stop
  It will take some time to mount and scan the tape.
  Please be patient.  Use ATTN (goose) to abort.
  (Operator mounts tape(s) requested by *RES)
  W000:VIDEO1 10:48:07 07/10/77 SUNDAY
  COMPLETED

Type command.

  dis w000:d?

   1 DG saved TUE 10/30/84 03:17:01

   1 DIRECTORY000 saved SUN 10/28/84 09:26:27
     (There are 5 older versions of this file.)

   1 DISASTER saved SUN 10/28/84 10:56:36
     (There are 5 older versions of this file.)

   1 DISIG saved 10/28/84 10:14:22
     There are 7 older versions of htis file.)

Type command.

  dis w000:d?s?  histroy=5

   1 DISASTOR saved SUN 10/28/84 10:56:36
   2    "     saved SUN 10/21/84 10:06:44
   3    "     saved SUN 10/14/84 10:12:13
   4    "     saved SUN 10/07/84 10:21:37
   5    "     saved MON 10/01/84 01:41:28
    (There is 1 older version of this file.)

  1 DISIG saved SUN 10/28/84 10:14:22
   2    "     saved SUN 10/21/84 09:20:31
   3    "     saved SUN 10/14/84 09:30:50
   4    "     saved FRI 10/12/84 03:03:43
   5    "     saved WED 10/10/84 02:40:21
    (There are 3 older versions of this file.)

Type command.
 *?stop
```

*RIP

NAME: *RIP

FUNCTION: Produce a ripple pattern on a printer with a PN train (or chain) mounted.

USAGE: Drain the printer.
Enter "mts *rip , ptrn", where ptrn is the name of the printer.
To stop the rippling, use the GOOSE or STOP jobs on the task.

RHB

NAME: *RHB

FUNCTION: To Hold & Release HASP Rate Classes

USAGE: Started automatically by *AUTOSTART & *HSP

*RMC

NAME: *RMC

FUNCTION: Lists RM Processors and permits some manipulating of same.

USAGE: mts *rmc

"p" prints all processors, "p xxxx" prints a particular processor and "q" shows the number of queued jobs and their locations. The information produced by *rmc may be used to determine which processor(s) need to be drained, if need be such as when a processor name is highlighted on the operators' console.

*RMD

NAME: *RMD

FUNCTION: Shuts down (drains) Resource Manager.

USAGE: mts *rmd

| *RMP*

| NAME: *RMP

| FUNCTION: Starts Resource Manager Processors

| USAGE: mts *rmp

*RMS

NAME: *RMS

FUNCTION: Starts Resource Manager

USAGE: mts *rms

*RST

NAME: *RST

FUNCTION: Restore user files from filesave or deadfiles tapes.

USAGE:

1. Enter 'mts *rst' on console.
2. *RST will ask 'WHERE IS A PRINTER'
3. When *RST asks 'WHAT FILES ARE TO BE RESTORED' do one of the following:

   (a) To restore one or more files enter filenames, one per line. Terminate the list with an end of file ('cncl').
   (b) To restore all files except some particular one(s) type 'EXCEPT', then filenames and end of file.
   (c) To restore entire volume, type 'ALL'. Program will ask for volume names to restore.

   NOTE: As used in restore a "FILENAME" will have one of the following formats:

   (a) (User1,User2) will cause all files of UserID1 to be restored under UserID2. The copies of the files under UserID1 will be left unchanged. If UserID2 is omitted, all files of UserID1 will be restored under that UserID. Both User1 and User2 must be 4 character UserIDs.
   (b) (UserID1,UserID2)Filename will cause the file given by UserID1Filename to be restored as UserID2Filename.
   (c) Filename where Filename is the full disk type name, ie. UserID concatenated with UserFilename or System File Name. If this format entry is given, only that particular file will be restored. Up to 250 file restore requests can be processed at one time.

(d) Userid1filename,userid2newfilename - similar to (b)

4. *RST will ask 'WHERE IS THE RESTORE TAPE'. Reply with the physical device name. (Tape to be used will be indicated on the form).
5. When the file is restored:
   The CSID followed by the file restored and 'COMPLETED' will be written on the operators console, the former when it begins restoring the file, the latter when it finishes.

EXAMPLE:

```
00419 15:28.34 mts *rst
00419 15:28.44 MTS WHERE IS THE PRINTER
00419 15:28.47 MTS..oper
00419 15:28.53 MTS WHAT FILES ARE TO BE
RESTORED
00419 15:28.57 MTS..(0ab2,k2t4)
00419 15:29.30 MTS..  (cncl)
00419 15:29.35 MTS WHERE IS THE RESTORE
TAPE?
00419 15:29.38 MTS..t901
00419 15:35.55 MTS K2T4ALFAR8
00419 15:35.59 MTS COMPLETED
00419 15:36.28 MTS WHERE IS THE RESTORE
TAPE?
00419 15:36.31 MTS..t908
etc, etc.
```

*RSV

NAME: *RSV

FUNCTION: Program to issue reserve commands to a group of disk drives when a single control unit and its drives are to be used by both halves of a split configuration with each half having exclusive use of certain drives. This program is only suitable for systems with more than one processor.

USAGE: Right after IPL, use the OFFLINE job to offline the drives that should not be used on that half of the system, then enter "mts *rsv" and when it asks for input enter the volume names of the disks on that controller it should be using, one per line. Use "cncl" to terminate.

*SAV

MAME: *SAV

FUNCTION: To save files which have been changed since the last file save.

USAGE: mts *sav

The program *SAV, will ask for a particular tape, e.g., FS0023, and, when the tape is mounted, procede to save files on it. *SAV can be used without ridding the system of users. Usually, only one save-tape is called for, but occasionally, a second may be requested.

If the tape requested by *SAV is in use (a user is restoring some files from it) the operator should type, "next", and a different tape will be requested by the program.

If a tape error occurs, the program will ask if it should continue to write on that tape. The operator should answer "no", and a new tape will be requested by *SAV.

If a new FSnnnn tape is requested, it should be initialized by writing a tape mark on it first.

*SAV aoutomatically starts the directory merge program which does the same thing as *FSM. No operator intervention is required by *FSM. *FSM is also started at every system reload in case *FSM or *SAV were interrupted by a crash. If *SAV generates any error messages of the form **** FILE SAVE ERROR, they should be reported to a systems programmer.

Note: The program, *SVW, will continue to be used for the large, weekly file save.

*SDB

NOTE: *SDB should be run after the SHUTDOWN ALL COMPLETE appears. If UB has been taken down before *SDB has been run, *SDB can be run from the UM system.
Also, see *SDM

NAME: *SDB

NOTE: If *SDB cannot be run for some reason, use a terminal which connects to the network and at "which host", type HG. Then, signon and--

```
$sou mnop:sdb
```

If HG isn't available:

```
net wu

sig w035

sou mnop:sdb
```

FUNCTION: To place a shutdown message in PCP(s) attached to the UB system. This procedure is used to enter a one-line UMnet/UMnet message which will be printed for any user attemting to access which ever system is down. It will not disconnect any users currently connected to the UB or UM systems. One message line is read from the console (terminated by a null line or an end-of-file) and placed in the PCP(s). The shutdown facility is then activated. Any previous message is replaced by the new message.

USAGE: mts *sdb

REMARKS: This procedure should normally be run after SHUTDOWN ALL has been entered at the console and you have checked to see that all PCP jobs have stopped. Use /t t mnet to verify this.

*SDB will skip any processors that are logically "offline", but may hang if any is "online" but not running.

*SDB (shutdown message) should normally be run after SHUTDOWN ALL or SHUTDOWN PART has been entered and all UMnet/MERIT jobs have stopped. Use /t t mnet to check.

Note: *SDB may be run on the UM console if necessary.

*SDM

NOTE: *SDM should be run after the SHUTDOWN ALL COMPLETE appears. *SDM can be run on the UB system if the UM system is already down.

Also, see *SDB

NAME: *SDM

NOTE: If *SDM cannot be run for some reason, go to a network terminal and at "which host" type HG.

```
signon ccid

sou mnop:sdm
```

If the HG machine isn't available:

```
        net wu

        sig w035

        $sou mnop:sdm
```

FUNCTION: To place a shutdown message in the UMnet/UMnret PCPs. One message line is read from the console This procedure is used to enter a one-line UMnet/UMnet message which will be printed for any user attemting to access which ever system is down. It will not disconnect any users currently connected to the UB or UM systems. (terminated by a null line or end-of-file) and placed in the PCP(s).

USAGE: mts *sdm

REMARKS: This procedure should normally be run after SHUTDOWN ALL has been entered at the console and you have checked to see that all MERIT/UMnet jobs have stopped. Use /t t mnet to verify this.

*SDM will skip any processors that are logically "offline", but may hang if any is "online" but not running.

REMARKS: *SDM (shutdown message) should normally be run after SHUTDOWN ALL or SHUTDOWN PART has been entered and all UMnet/MERIT jobs have stopped. Use and /t t mnet to check.

*SFT

FUNCTION: Turns on SOFTCHK which enables reporting of soft machine checks.

USAGE: Started automatically by *AUTOSTART

*STA

Note: *STA may require a jobdump if it fails (use mts *srk). Notify Charles Engle if this happens.

NAME: *STA

FUNCTION: To save the billing information (*STATISTICS) on magnetic tape.

USAGE: The STA program normally should be initiated whenever the MTS file (*STATISTCS) is nearly full. MTS will print the message

```
*** STATISTICS IS FULL ***
```

on the operator's console whenever a user signs off if there are more than 7500 statistics records in the file (one record is added to the file every time a job signs off). The program may be initiated by issuing MTS *STA. This should be done as soon as it is convenient. If the statistics file is allowed to completely fill up, SNARKS will begin to occur as jobs sign off; if this happens, *STA should be initiated at once.

The *STA program initially copies the file *STATISTICS into the internal work file SYS.:STATISTICS, then, immediately empties *STATISTICS. During the initial copy operation, the system will prevent users from signing off. Therefore, it may be preferable not to run the program if the system is *extremely* overloaded. After the copying is completed, the program will request that the operator mount a daily statistics tape, i.e., a mount message will appear requesting a particular tape. After statistics are saved on the daily tape, the program will request a monthly statistics tape, again causing a mount request to appear on the console. After statistics are saved on the monthly tape, the message

```
*** STATISTICS SAVE SUCCESSFULLY COMPLETED
```

will be printed on the console. At this time, the work file will be emptied indicating that the operation was successful.

The tapes used for the daily and monthly statistics are maintained in the directory file SYS.:DIRECTORY The STA program automatically maintains this directory and selects the correct tape. The exact range of dates over which the statistics were accumulated are automatically recorded in the directory by the program; therefore, it is no longer necessary to write this information on the tape cannister.

System Crashes and Recovery Operations during *STA

If the system should crash during the saving of the statistics file, recovery operations will automatically be initiated when the system is reloaded. Each time that the system is reloaded, the STA program is automatically run to determine whether the system crashed during the statistics file save. This is indicated by the statistics work file SYS.:STATISTICS still being full, as described above, the first thing that the program does is to copy *STATISTICS into the work file and the last thing done is to empty the work file. If the work file is still full at system reload, the previous file save did not run to completion. The reocovery operations consist of re-saving the work file on both the daily and monthly save tapes. When this occurs, the message

```
        *** LAST SAVE OF STATISTICS UNSUCCESSFUL -
            REPAIR OPERATIONS COMMENCING
```

is printed on the console.
The operator may also manually check for a system crash and perform repair operations by issuing the command

```
        $run sys.:newsta+sys.:dirlib par=init
```

If the work file is empty, no save operations will be performed.

If the STA program is run normally when the work file is already full, the operator will be notified that the file is currently full and inquire as to whether recovery operations should be commenced. This may be necessary, for example, when the previous run was interrupted by an attention interrupt.

Errors in running STA

If an error occurrs while *STA is running, the program should be rerun, i.e., MTS *STA. The program will put forth questions which you must answer such as:

```
DO YOU WANT TO REPAIR?

yes
```

You should answer YES whether the daily or monthly tape is being used. Once the repair has been completed the system will ask:

```
DO YOU WANT TO SAVE THE CURRENT *STATISTICS FILE?

no    (if the repair was successful)
```

If the repair was not successful, try using a new tape. If this doesn't work, some help from a systems programmer is needed.

When an error is detected while running *STA, the program will save the data in *STATISTICS in a file with a name of the of the form STATmmmdd, e.g.,STATAUG08. When *STA is rerun, this file is used for the repair operation. The file will remain under the ID SYS. until manually destroyed by a systems programmer.

*STI

NAME: *STI

FUNCTION: This program checks to see whether the statistics "work" file is proper order. If the system crashed during the running of *STA, some repairs may have to be made. These repairs are done automatically.

USAGE: mts *sti

NOTE: This program (*STI) is automatically run at IPL time and should almost never have to be initiated manually.

*STM

NAME: *STM

FUNCTION: To monitor the number of lines in *STATISTICS during unattended mode and to empty *STATISTICS when it nears the number of lines that would cause it to be considered full by MTS.

USAGE: This procedure may be invoked when placing the system in unattended mode. It will continue to run until the task is stopped or until the system is reloaded. However, allowing it to run when the system is not in unattended mode should not cause a problem since it will simply wake up once an hour, determine that the system is not in unattended mode, and then go back to sleep.

To invoke the procedure, enter

```
MTS *STM
```

on the operators' console. This signs on under the ID SYS and sources to the file SYS:STATMONITOR, which executes the macro that performs the procedure.

At the time the procedure is invoked, and once an hour thereafter, the procedure will check to see if the system is in unattended mode. If the system is in unattended mode, the procedure will determine if there are more than 6500 lines in *STATISTICS. If there are more than 6500 lines, the procedure will expand SYS:STATISTICS, if necessary, to a size sufficient for adding the contents of *STATISTICS, copy *STATISTICS to the end of SYS:STATISTICS, and empty *STATISTICS.

Once this procedure has saved *STATISTICS, the next time *STA is invoked, it will issue the message:

```
   *** LAST SAVE OF *STATISTICS APPEARS UNSUCCESSFUL OR
*STATISTICS WAS

   *** DUMPED DURING UNATTENDED MODE

   *** SHOULD THIS BE REPAIRED AND COPIED TO TAPE?  ENTER "Y" OR
"N"
```

This is because the result of saving *STATISTICS with this procedure appears to *STA to be the same as does an unsuccessful save. The above message should be answered with Y. This will cause a "repair" operation to be performed, and when it is successfully completed, the message

```
*** DO YOU WISH TO SAVE CURRENT *STATISTICS FILE?  ENTER "Y" OR
"N"
```

will appear. Assuming that the original reason for this invocation of *STA was to save the current *STATISTICS file, this should also be answered with Y. If it is not, or no longer, necessary to save the current *STATISTICS file, this should be answered with N.

If there is a problem with the *STM procedure, e.g., if SYS:STATISTICS cannot be expanded sufficiently, SYS:STATCOPY will be run, saving *STATISTICS in a file with a name of the form SYS:STATYmmmddi (e.g., STATYAUG17 or STATYAUG17A). Once a file belonging to the ID SYS that has a name beginning with "STATY" exists, whether as a result of this procedure or of SYS:STATCOPY having been run previously, all subsequent emptying of *STATISTICS by this procedure will be done by running SYS:STATCOPY. Furthermore, *STA will not save the current *STATISTICS file while there is a file belonging to SYS with a name starting with "STATY", and a systems programmer will have to manually make repairs.

*SVS

NAME: *SUM

FUNCTION: To remove the network message installed by use of *SDM.

USAGE: mts *sum

*SVS

See *SVW for *SVS and *SVW differences and similarities.

SVW

NAME: *SVW

FUNCTION: Weekly file-save. Saves a copy of all disk files on tape.

USAGE: mts *svw

Before running *SVW, there are several things you must do.

- A BROADCST message must be sent to the users telling them to signoff.

- The phone message must be changed.

- Execution must be held.

- The "SHUTDOWN PART" command must be issued.

- The system should be taken out of unattended mode (/mod uam 0).

- Make sure that no users are signed on. It is all right to leave the public files with user id SYS. (*CMD etc.) running.

Behavior and output are pretty much the same as the daily file-save (see example under *SAV) except that each stream fills up more tapes, and (as you might therefore have guessed) it takes a good deal longer.

NOTE: If one of the streams fails, call Steve Burling and don't restart the individual stream, but instead, stop the individual streams and restart *SVW - otherwise, the wrong tapes will be called for.

Use the program, *FSC to make copies of the filesave tapes. See the write-up in this manual.

NOTE: When the comment IT IS OK TO LET THE LINES OUT, NOW appears, do the following:

- Type STARTUP

- If there are no clerks or DPAs on the first floor, offline ptrx, ptrx and ptrx as well as pchx.

- Start *RMS, *RMP, *LAS and *MNS

```
$release ex
```

Do not shut the system down for any reason before the "IT'S OK TO TAKE THE SYSTEM DOWN, NOW" message is printed on the operators' console. If the system crashes after the lines have been let out and before the above message appears, contact one of the FIREFIGHTERS. It is not necessary to restart the weekly filesave.

If there is trouble concerning the running of *SVW, call Steve Burling.

If *SVW is started on the wrong date, the system will ask if you really meant to run it. This is to guard against *SVW being run accidentally. If *SVW suspects that it is being run at the wrong time, it will ask if it is ok to proceed. Responding NO to the question will terminate *SVW with no other bad effects.

If one wants to run a special, large, weekly type filesave, *SVS may be used. *SVS is just about the same as *SVW except that when it begins, it will ask what series of filesave tapes to use. After *SVS prompts you, type and enter the series number, e.g., 1100 followed by a null line.

NOTE: See *TSN

NOTE: If the message-

```
   FILE CCID:FILENAME DSCB-E NOT FOUND AT XXXXXXXXX
```

appears, notify a firefighter (but don't wake anybody up to do it).

*TNR

NAME: *TNR

FUNCTION: Produce a ripple pattern on a printer with a TN train (or chain) mounted.

USAGE: Drain the printer.
Enter "mts *tnr , ptrn", where ptrn is the name of the printer.
To stop the rippling, use the GOOSE or STOP jobs on the task.

*TPD

NAME: *TPD

FUNCTION: By default, TPD tests 9-track drives in the following way. First, it tests the drive at 800 BPI by writing a tape mark, followed by a random length block consisting of all possible bit combinations. It then does a backspace file (backing over the block and tape mark), followed by a forward space file (moving past the tape mark), and reads and checks the random length data block. If the data checks correctly, TPD repeats the entire sequence (without rewinding the tape) 300 times (by default) with the MTS read data check recovery disabled (RETRY=0), i.e., any data check will show as a "permanent" error.

If no errors are found after the 300 repeats of this sequence, the tape is rewound and tested at 1600 BPI and then again at 6250 BPI. The test will be skipped (with a comment) for any density not available on the drive being tested.

For 7-track tapes, the test is similar, except that each repeat in the sequence of 300 is done at a different one of the 15 7-track "modes" so that all densities, both parities, the translator, and the data converter are all tested. Because of this, only one 300-repeat sequence is done for 7-track tapes.

USAGE: mts *tpd

The program will ask for a tape unit name and parameters. The unit name may be any 7- or 9-track tape drive. The optional parameters may be entered in any order after the unit name, each item separated by at least one blank.

EXAMPLE:

The program will ask for a tape

```
mts *tpd

***ENTER TAPE UNIT NAME AND PARAMETERS

t900 1600
```

(The above example would cause drive T900 to be tested at 1600 BPI only.)

The parameters which may be used to modify, somewhat, the default behavior of TPD follow:

800 For 9-track tapes, the tape drive is tested only at 800 BPI. For 7-track tapes, this program is ignored.

1600 For 9-track tapes, the tape drive is tested only at 1600 BPI. For 7-track tapes, this parameter is ignored.

6250 For 9-track tapes, the tape drive is tested only at 6250 BPI. For 7-track tapes, this parameter is ignored.

ALL This parameter causes each of the tests to be repeated continuously until the end-of-tape marker is reached (instead of just 300 tries).

*TPS

NAME: *TPS

FUNCTION: A file to run UNSP:TAPEDRIVES. This program informs the operator of the number of available tape units. If a tape drive is OFFLINE or in use, the program will consider it unusable by the operator. The program can distinguish between 800/1600 and 1600/6250 drives.

USAGE: mts *tps

EXAMPLE:

```
-At This moment there are

 AT MOST

  1 7-Track and

  2 9-Track

    1 6250/1600-BPI
    1  800/1600-BPI

drives free.
```

*TSH (Obsolete)

NAME: *TSH

FUNCTION: Adds tapes to, or deletes tapes from UNSP:TAPESHARE

USAGE: mts *tsh

EXAMPLE:

```
mts *tsh

addr <rackno> <owner CCID>

EOF to terminate
```

If an error was made while entering the tape, e.g., the wrong CCID was used, it can be removed and re-entered-

```
delr <rackno> <owner CCID>   (then, re-add it using
ADDR)
```

REMARKS:

When a tape has been entered in TAPESHARE via addr or addrack, the owners may permit the tape any way they like by running UNSP:TAPESHARE.

NOTE: To find out if a tape is in TAPESHARE and which CCID owns it, type-

```
t <rackno>
```

Example: t s1091

*TSN

NAME:	*TSN
FUNCTION:	The weekly (full) file-save slave stream. *SVW will start 6 to 8 of these. Not normally used directly.

*UDP |CONT('*UDP')

```
   NAME:          *UDP
$Sig UD$.
*
*  22 Nov 88 : sgr
*  I added the mail:udp.semaphore file to force *UDP to wait unt
*  *MDP has finished it's LOCK/UNLOCK cycle on *MSG. It is possi
*  for *MDP to exclude *UDP if they both signed on simultaneousl
*  So now *MDP locks udp.semaphore, then Locks & Unlock *msg and
*  then unlocks udp.semaphore, thus releasing *UDP to do its Loc
*  Unlock cycle on *Msg. This is the only reason I am able to so
*  up with for *UDP not running for a long time. Now we get to s
*  if I'm right ...
*
*  23 Nov 88 : lbt
*  renamed ID W494 to UD$. on UM-MTS
*
*  31 May 93 : lbt
*  added a $LOCKSTATUS on *MSG
*
$lock mail:udp.semaphore mod wait
$Lockstatus *Msg
$Lock *msg mod nowait quit
$Unlock *msg
$unlock mail:udp.semaphore
$Source UD$.:UdmasterDup
```

*UNU (Not presently used)

NAME: *UNU

FUNCTION: Enables all the terminals on the NUBS & UNYN SCPs and starts the job status monitors at NUBS and UNYN.

REMARKS: If the automatic NUBS and UNYN restart procedure at 8:00 a.m. doesn't work or is missed because MTS was down, then this file can be used to bring the terminals at NUBS and UNYN back up.

ABNORMAL SITUATIONS SUMMARY

Abnormal Situation and Error Conditions

This section of the manual outlines abnormal error conditions and situations which the operator is expected to detect and correct.

Error situations which arise and which are not covered by this section of the manual should in all cases be reported to a Systems Programmer or to a senior member of the Operations Staff. Subsequently, new problems will be diagnosed and included in this section of the Operators Manual.

The following sections assume familiarity with the preceding portion of the manual and make free use of terms defined therein.

Indications

1. Various software detected errors will appear as error messages on the operators' console. Such errors appearing in the form of messages, from the disk support routines, the tape handling routines, or possibly from individual processors such as HASP.

2. Errors or malfunctions in devices may occur which are not immediately apparent from the behavior of the operators console. In these cases the operator is expected to recognize and diagnose the error condition and take corrective steps.

3. Channel and control unit errors may also occur as above (1).

4. A few central processor error conditions are also included in this section of the manual.

5. A rather large number of error situations may originate from the HASP processor and these are detailed below under separate sections.

CONSOLE ERROR CONDITIONS

Snarks

This problem situation is indicated by the following lines on the operators console:

```
00023 10:40.13 MTS

********************

HELP--SNARK IN MTS !

********************
```

do an mts *srk at operators' console

this message indicates that the task whose number is specified on the first line (23 in sample above) has encountered an irrecoverable system error. In short, the job has folded up and is going out feet first. A SNARK indicates that a job is terminating because of a failure in MTS. It is necessary to take special steps to remove a SNARK'ed task from the system. These steps are outlined here.

Step 1

*********TAKE A JOBDUMP*********

Step 2

Determine whether the task is associated with a terminal job or a HASP job. To do this, initiate the TASKS job program with the job number of the job which is SNARK'ed as an argument. This will type out the standard tasks message format. By looking at the devices associated with the job you will be able to determine for terminal tasks the line adaptor it had and for batch tasks the S8 number.

Step 3

The next step after having identified the type of job is to initiate the STOP job program giving it as an argument the task number of the job which has SNARK'ed. If 'stop' fails to terminate job, see STEP 4. However, should STOP work OK and should this be a terminal job which has SNARK'ed proceed to STEP 6 below, then you are finished.

Step 4

This step is only to be undertaken if 'stop' failed. This is done for both terminal and batch tasks. First read the dire warnings under BLAST earlier in this manual. Then invoke the BLAST job program with the SNARK'ed job's task number. The BLAST job program will irrevokably remove the SNARK'ed task from this system. Irrespective of the results obtained from the indication of the BLAST job program you may proceed to STEP 5.

Step 5

When a job has been removed from the system using the BLAST job program or if a message is returned from 'stop' the accounting records have been somewhat disturbed. And in most cases it will be impossible for the user with a CSID of the job which SNARK'ed to start another task. To permit this user to continue running, it is necessary for the operator to use the MTS library file *FIX to restore the user's account record. This program is invoked by the command form MTS *FIX. The use of this program is described elsewhere in this manual.

Step 6

For terminal jobs the successful reinitiation of the terminal device requires that the operator enable the line. This is similar to the operation performed by the MTS *LAS (or *DCS,...) at IPL time. However, the form here is to issue a command to the system of the form MTS xxxx where xxxx is the device name of the terminal in question, as determined from the TASKS output obtained above.
For example, MTS LA23, MTS CA11, MTS DS12

HARDWARE

CPU

Most CPU machine errors will invoke the "Machine Check Handler" package and possible error messages, operator actions, etc., are outlined in the following pages. Some CPU problems prevent anything from running including the machine check handler. But, occasionally the CPU will just stop with no indication as to why. There may may be no response to a request, etc. The attempt to take a dump or run "Serep" will usually indicate whether hang was due to Hardware or Software.

Actions Required of Operator When Machine Checks Occur

Machine checks will be indicated by the appearance of one of two messages on the console, followed by 0 or more lines of comment.

The following is a brief list of the messages that can be expected to come from the machine check handling code.

```
 **MCHK**  INCOMPLETE LOGOUT
     Insufficient information logged by the machine to recover
     from the machine check
 **MCHK**  TIMING FACILITY DAMAGED
     Failure in the system clock or other hardware
 **MCHK**  SYSTEM DAMAGE
     Serious machine check that precludes further processing.
     Printed when the system damage bit is on in the machine
     check interrupt code.
 **MCHK**  EXTERNAL DAMAGE
     Hardware failure during channel or other operation outside
     the CPU itself.  Printed when the external damage bit is on
     in the machine check interrupt code.
 **MCHK**  RECOVERED MACHINE CHECK
     Machine failure that was corrected by the hardware.
     Printed when the system recovery bit is on in the machine
     check interrupt code.
 **MCHK**  SOFT RECORDING DISABLED
     No more recovered machine checks will be reported.  The
     SOFTCHK operator command can be used to turn the reporting
     of recovered machine checks on or off.
 **MCHK**  MACHINE CHECK IN WAIT STATE
     An otherwise fatal machine check occurred in WAIT STATE,
     the system will try to ignore it.
 **MCHK**  SUPERVISOR DAMAGE
     Some information crucial to the supervisor was damaged by
     the machine check.  RE-IPL required.
 **MCHK**  RECOVERED STORAGE PROTECT FAILURE
     Failure in the storage protect mechanism that has been
     recovered from by reloading the damaged keys from the
     relevant PCBs.  The system will attempt to continue.
 **MCHK**  STORAGE PROTECT FAILURE
     Solid failure in the storage protect mechanism.  Printed
     when the key error bit is on in the machine check in the
     machine check interrupt code.
 **MCHK**  STORAGE DATA FAILURE
     Uncorrectable error in main storage.  Printed when the
     storage data failure bit is on in the machine check
     interrupt code.
 **MCHK**  MACHINE CHECK IN M/C HANDLER
     Second machine check while trying to recover from a machine
     check.
 **MCHK**  DEGRADATION REPORTED
     The hardware has moved some part of the machine from
     service because it is malfunctioning.  Printed when the
     degradation bit is on in the machine check interrupt code.
 **MCHK**  NO SUBCLASS BIT ON
     Machine check occurred for a reason that cannot be
     determined.
 **MCHK**  INSTRUCTION PROCESSING DAMAGE
```

Failure in the instruction processing part of the machine. Printed when the processing damage bit in the machine check interrupt code is on and both key error and storage data failure are off.

MCHK VM REFRESH RETRY

Uncorrectable storage error on a page for which we have good copy paged out. The sytem will attempt to recover the page from the paging device.

MCHK INTERVAL TIMER ERROR

Interval timer (location 50) failure.

MCHK ENVIRONMENTAL WARNING

Power, air conditioning, or other similar failure. Printed when the warning bit in the machine check interrupt code is on.

MCHK PROGRAM INTERRUPT IN MACHINE CHECK HANDLER

Program interrupt while processing a machine check interupt.

I/O ERROR. PATH=pppp CSW=cccccccc cccccccc LCL=llllllll

Interface control check, channel control check, or channel data check on path "pppp". The CSW (CHANNEL STATUS WORD) and LCL (LIMITED CHANNEL LOGOUT) formats are described on the system 370 reference card.

SOFTWARE FAILURE

A SUPERDUMP just occurred.

NO LOG AREA AVAILABLE

The machine check or channel error being processed will not be logged since too many logs are already pending. Recovery from the error will not be effected.

##RE-IPL##

The system is down.

1. Punch Errors

PCHn: Machine ERROR CARD PUNCH STATUS UUIIFFSS
Program

UU: Program flag, values are (in hex):

01:\\Return on punch check after message
02:\\Return immediately on punch check

II: Program flag, values are:

80:\\Initial rejection of CCW
40:\\Bus-out check occurred
20:\\Repunching card after punch check
10:\\EOF (unit exception) on retry

FF: Sense flag - See description above (should be 01)

SS: Sense data

2. Printer Errors

PTRn: Machine ERROR PRINTER STATUS UUIIFFSS
Program

UU: Program flags:

10:\\Doing UCS buffer load
08:\\Return if channel 12 detected
04:\\Return if channel 9 detected
02:\\Return on print check
01:\\Expect invalid character check

II: Program flags:

80:\\Initial rejection of CCW
40:\\Bus-out check occurred
20:\\Equipment check occurred
10:\\Bad character printed
08:\\Parity check occurred
04:\\Immediate operation
02:\\Reprinting last line after print check
01:\\Return after reprint

FF: Sense flag as above (should be 01)

SS: Sense data

3. Card Reader Errors

Machine ERROR CARD READER STATUS UUIIFFS
Program

UU: Program flags:

40:\\Return on fatal error

20:\\Ignore validity check if col. 1 has a 7-9 punch

10:\\Return after validity check after message

08:\\Return immediately on validity check

04:\\Unusual command sequence is OK

02:\\Command was column binary

01:\\Return in read check

II: Program flags:

40:\\Overrun has occurred

20:\\Got unit check on dummy read after run in

10:\\Must do dummy read-no feed after run in

08:\\Bus-out check occurred

04:\\Command was initially rejected

02:\\Command is read no feed

01:\\Command is feed and stack, no read

FF: Sense flag as above (should be 01)

SS: Sense data

MEMOREX 1270 LINE I/O ERRORS

The following error messages are printed on the HASP log.

```
I/O ERROR ON LINEm cc,ssss,iirr,ttee
```

where:

cc	-Channel command code being executed
ssss	-Channel Status Code
ii	-Unit Sense Information
rr	-BSC-Terminal Response
ltt	-Internal Sequence and Command Code
ee	-Expected Response

DISK UNIT CHECKS

If the program which dumps error records from VM into the files falls behind and it is not possible to log an error a message is written on the operator's console. This message is

DXXX ERROR C

where "c" is the error code from the above list. All other information about the error is lost in this case.

To print Disk Unit Checks, See the programs EREP, PM2 or MTA.:PUC mentioned elsewhere in this manual.

DISK, SOLID STATE DISK AND FIXED HEAD FILE UNIT CHECKS

Another, similar program for printing disk and fixed-head-file errors is also available. This program can print output in the same format as mta.:printuc3330 or can summarize the printout as well.

```
$signon W---

$run mta.:puc sprint=*sink* (if you wish the output on your
terminal) or, sprint=*print*.

EXECUTION BEGINS
WHICH DEVICE TYPE?  (2305, 3330, OR BOTH)

   (operator types option)

ENTER STARTING TIME (E.G.  8:20 JAN 13)

1 am jan 17 (EXAMPLE)

ENTER ENDING TIME

3 am jan 17   (EXAMPLE)

SUMMARIZE?  (YES OR NO)

yes

*PRINT* ASSIGNED RECEIPT NUMBER
EXECUTION TERMINATED
```

If the option "yes" was given in the above example, the output will be sorted by module and format 6 type information will be accumulated. If the option "no" had been given, then the output would be the same as if mta.:printuc3330 had been run.

```
$run mta.:puc sprint=*print* par=stc
```

(The "STC" parameter says to assume the disks are STC disks and interpret the physical drive ID accordingly)

```
EXECUTION BEGINS
*PRINT ASSIGNED RECEIPT NUMBER
EXECUTION TERMINATED
$signoff
```

EREP-(SYS.)

LOGOUTS-CPU AND CHANNEL ERRORS--PRINTING OF USING IBM'S EREP

Note: Presently (16Jul90) use SYS.:ERP.2 rather than EREP, e.g., $sou erp.2 after altering the dates or any other material in the file SYS.:ERP.2 using the MTS Editor.

This is a complete rewrite of the obsolete version of EREP that we have been using and contains many new features. The IBM manual GC28-0772-3 contains a writeup on the output formats and parameters for controlling this version of EREP. There are some suggestions for useful combinations below.

FUNCTION: This program is used to print some or all of the machine check, I/O outboard errors, etc., which are recorded on system disk files.

SYS.:EREP contains a driver program that runs this EREP under VSS. It uses the following logical I/O units:

SCARDS CMDSTAT type 4, 10, 25 records
Default: INIT:UC3330LOG(*L-40)+INIT:MCFILE
however, we are presently using the following:

```
INIT:MCFILE+INIT:UC3330LOG(*LAST-500)+INIT:MCFILE.UB+INIT:UC3330L
```

Note: The use of such a large number, *L-500 is for the purpose of getting a "history" that goes back far enough.

SPRINT EREP error report printed output
Default: *PRINT*

SERCOM EREP messages and prompting messages
Default: *MSINK*

GUSER Responses to prompts
Default: *MSOURCE*

SPUNCH Error records in MVS history tape format
Default: -##ACCIN

0 Source of additional VSS commands to be executed immediately before calling EREP
Default: No additional commands are read

If SCARDS is assigned to *DUMMY* and SPUNCH is assigned EXPLICITLY to the location of MVS history tape data, then the CNVTEREP step will be skipped and the history tape data will be read. SPUNCH must not be allowed to default in this case even if the default is correct since the file will be emptied as part of the defaulting process.

EREP uses a machine carriage control on its printed output, so remember to use MCC if you copy a file containing EREP output to *PRINT*.

SYS.:EREP will prompt on SERCOM and GUSER for EREP parameters after CNVTEREP has been run (if necessary) to convert CMDSTAT to MVS history tape format. CNVTEREP still prompts for a starting and ending time, but this isn't really necessary any more since EREP now accepts a time range as well as a date range. Some useful parameter combinations you can use at this prompt are:

```
     Tape errors:         THRESHOLD=(1,1)

     Disk errors,
     machine checks
     and channel checks:  TYPE=MCOT,DEV=(33XX)
```

Example: (Use upper case)

```
$run erep sercom=-ser  (you don't usually need  to  see  SERCOM)
          [sprint=-x]   (use  sprint  to  direct  output  to  a
          temporary file, etc., if you don't want it  to  go  to
          *PRINT*).


8:00 JUN 20

8:00 JUN 23

ok

TYPE=MC      (picks up machine checks and channel errors)

  or

TYPE=MC,PRINT=PS       (if a more detailed printout is desired)

  or

TYPE=OT,THRESHOLD=(1,1)       (one   way   of  picking  up  tape
errors)

  or
```

NOTE: For the time being, whenever EREP is used to pick up disk errors, use EREP.OLD.

```
TYPE=OT,DEV=(33XX)       (picks up disk errors)

  or

TYPE=OT,CUA=(BXX,FXX)       (picks up errors on channels B & F)
```

or

TYPE=OT,VOLID=(MTS001,MTS004) (picks up errors on two disks mentioned)

or

TYPE=OT,VOLID=(MTS001,MTS004),PRINT=PS (same as last parameter but produces expanded printout)

$ENDFILE

Remember that SPRINT defaults to *PRINT*.

If you must go back more than a couple of weeks or so, SCARDS=INIT:MCFILE or INIT:UC3330LOG must be included in the $RUN command. INIT:MCFILE is for machine and channel checks and INIT:UC3330LOG is for disk, tape, etc. errors.

Note the parantheses around the device type, e.g., dev=(33xx). These are required even if only one device type is given as in the example above. Each of these must be a separate run, EREP is not capable of making more than one type of report per run. You can assign SCARDS to *DUMMY* and SPUNCH to -##ACCIN as described above to save time on the second and subsequent runs. See the IBM manual mentioned above for more possibilities and examples of each of these reports. Most of the parameters that worked with the old EREP also work with this one, but often, there is a better way to do things.

Example:

```
$run ifcacs01

     $run erep sercom=-ser sprint=-x
     8:00 jun 20
     8:00 jun 23

     ok
     type=mc   (or type=mc,print=ps)
               "print=ps" provides a much much more detailed
               printout of the error condition(s) but can be
               quite voluminous so is used infrequently.
     $endfile
```

If you have sent the output to a temporary file, e.g., -x, and you'd like to print it, you can:

EREP-(SYS.)

```
$run *pagepr scards=-x
```

or

```
$cont *print* printer=line

$copy -x *print*

$rel *print*
```

MEMOREX: LINE "HUNG"

Occasionally, the channel to which the Memorex Control Unit is attached may be out of synchronization due to a missed interrupt. The "Hangup" can usually be cleared by generating an interrupt for the appropriate device address. An Operator may examine and probably solve the problem in the following manner:

<u>Procedure to present CHANNEL-END and DEVICE-END to the Channel from the 1270 CE panel.</u>

Simple CEDE

A. Set TEST switch OFF (in the bottom row of switches).

B. Set TEST ADDRESS switches to desired address. <u>NEVER</u> change address switches with TEST switch on. Remember that Memorex Address "0" is equal to Channel Address X'C0'.

C. Press the gray CMD switch (in the TEST FUNCTION switch group), leave it in.

D. Set roller to panel "A".

E. Set the COMMAND BIT switches (at the right end of roller A) to 011111. (This is the 1270 internal code for a DISABLE command). For a zero bit, the switch should be DOWN; for a one bit, it should be UP.

F. Set TEST switch ON.

G. Push SINGLE CYCLE (in the bottom row of switches) down momentarily; it will spring return to the normal position.

H. Set TEST switch OFF.

I. Repeat the procedure for other addresses as needed.

CEDE with UNIT CHECK and TIMEOUT status.

Follow the procedure given above for the simple CEDE, but set the COMMAND BIT switches in step E. to 00010. This is the 1270 internal code for a READ command. After it is entered, the READ command will "timeout" in 28 seconds or less and present ending status to the Channel.

')

Some Memorex 1270 Information

Displaying line information

```
Display address using TEST ADDRESS  with  TEST  switch
off.
 A Roller
  1=transmit
  2=receive
  6=prepare
  7=enable
  F=disable
  0=dead or thinking

 B Roller
  Buffers

 C Roller
  If  CS,  CD,  or  DSR  lights  are  out, line is not
connected

 Examples:

Set roller to "A"

To disable a line
     Set  TEST  ADDR  switches  to   correct   address
     (0101=SA05)
     Press CMD button
     Set TEST switch on
     Set switches in COMMAND REGISTER (F=Disable)
     Flip SINGLE CYCLE switch
     Set TEST switch off
     CMD off

To enable a line

Set roller to "A"
     Set address switches at TEST ADDRESS
     Press CMD button
     Set TEST switch on
     Set switches in COMMAND REGISTER (7=enable)
     Flip SINGLE CYCLE switch
     Set TEST switch off
     CMD off
```

CARRIAGE CONTROL TAPE FOR MTS

The 1403 printers use 11-inch paper printing 6 lines per inch (normally). Thus there are 66 lines per page. Printing is usually done in units of a logical page. A logical page is defined as the 60 middle lines (of the full 66 lines) --- thus the first and last three physical lines on the page are not part of the logical page. In either machine carriage control or logical carriage control (i.e., USASI or ASA carriage control) a request to skip to the top of the next page will be interpreted to be a request to skip to the fourth physical line on the next page (this is where Channel 1 is defined to be).

LOGICAL CARRIAGE CONTROL

The printer has a twelve channel carriage control tape which moves synchronously with the paper. For each line on the page, there is a corresponding line on the carriage control tape which may have a hole punched in one or more of the twelve channels. Thus, a printer command such as "skip to channel 1" has the effect of moving the tape (and thus the paper also) until a hole is sensed in channel 1 of the carriage control tape. For the tape used by MTS, this has the effect of postioning the paper at the fourth physical line on the page.

MACHINE CARRIAGE CONTROL

Machine carriage control acts at a lower level than logical carriage control and, in fact, logical carriage control characters must be converted to machine carriage control characters (not a straight translation mind you) by system routines. The first byte (i.e., character) of the output record to be printed is interpreted as the command portion of a CCW (channel command word) for the printer. The reader is directed to IBM form X20-1703 (the "green" card) for a description of the legal machine carriage control codes. Unlike logical carriage control, machine carriage control <u>always</u> spaces after printing (rather than before printing).

CARRIAGE CONTROL TAPE FOR HASP

Col	Punch	Col	Punch	
3	3	69	3	
4	11	70	11	
7	1	73	1	
8	5	74	5	
11	10	77	10	
17	6	83	6	
22	4	88	4	
27	7	93	7	
37	2	103	2	
47	6	113	6	
52	4	118	4	
57	7	123	7	
66	8	132	8	cut

(con't. next column)

CMDSTAT

This writeup describes the CMDSTAT buffer-to-file (CMDPIKUP) and file-to-tape (CMDTAPE) programs. The first part covers what an operator needs to know for normal operations. The second part describes the programs and explains the possible error conditions and recovery actions. Everything covered in the first part is also covered in the second.

OPERATIONAL INSTRUCTIONS

The CMDSTAT file-to-tape program (*DWB) has been changed in an attempt to make it more reliable. Hopefully, when things start to go wrong it will ask either the operator or the firefighter for help, rather than just letting things get worse as in the past.

If it is only suspicious about something, it will send a $MESSage to the firefighter (using the names in SEG2:MONITORS). If it thinks something is really wrong, it will ask the operator to get some help before going any further.

There is also a change to procedures: the program will now ask the operator to create new M tapes, rather than waiting for some forgetful programmer to remember to do it.

New Tapes

When *DWB needs a new tape, it will print on the console

```
      *DWB: MOUNT NEW CMDSTAT TAPE RING IN WITH RACK Mnnn:
```

and wait for an operator response. Get a new (blank) tape, put on the appropriate sticky labels for Mnnn, hang the tape, and enter the drive name to the program. *DWB will label the tape and release it. Then it will issue a normal mount request for the new tape. *DWB will ask for a new tape when it gets to the end of the current tape, or when it gets tape errors on the current tape. If you think the program is starting new tapes too frequently, tell the firefighter (*DWB is probably also sending messages to the firefighter about the problem, but do it anyway). If *DWB mounts a tape and then shortly thereafter releases it and mounts it again, check to see if the message

```
       *DWB: CMDSTAT TAPE ERROR; PLEASE TRY ANOTHER DRIVE
```

appears on the console. If it does, *DWB has gotten an error writing the tape (this error may or may not show up on the console). Mount the tape on a different drive this time. If the message does not appear, something is probably wrong -- tell the firefighter.

If *DWB prints the message

```
             *DWB: IS ANOTHER COPY OF *DWB RUNNING?
```

Do a '/t u sys' to see if there is another copy. Respond 'Y' if there is, and 'N' if there isn't. If another copy isn't running, something is wrong, and *DWB will probably print another message shortly.

If *DWB prints

```
 *DWB: NO ROOM IN CMDFDIR AFTER nnnn; CALL SYSTEMS PROGRAMMER!
```

Call someone. Don't enter any response to the program; just let it be until the programmer can look at it. It has successfully written the tape, but is unable to update the directory.

If *DWB has trouble reading or writing some file, it will print

```
       *DWB: ERROR ON fdname.  CALL A SYSTEMS PROGRAMMER!
```

Call someone. Don't enter any response to the program; just let it be until the programmer can look at it. If "fdname" is CMDDIR, *CMD will probably become ill within the next hour or so unless someone fixes the problem.

If *DWB prints the message

```
         *DWB ERROR: JOBDUMP AND/OR CALL SYSTEMS PGMR!
```

it is confused in some way. The best thing to do is call the firefighter; if you can't get some help, jobdump it and then STOP it.

DESCRIPTION AND FIREFIGHTERS' INSTRUCTIONS

Overview

CMDTAPE writes files of CMDSTAT data to tape. These files are filled with data from the CMDSTAT buffers by CMDPIKUP. When CMDPIKUP fills a file, it makes note of that fact in SYS:CMDDIR, and STRTJBs *DWB. CMDTAPE copies the file to tape and then empties the file so that CMDPIKUP can use it again.

Files SYS:CMDSTAT1 and CMDSTAT2 are normally used for this purpose. If CMDPIKUP fills both of them and needs more space, it will create a "temporary" file named CMDSTATc (where c is some character). After CMDTAPE copies a "temporary" file to tape, it will destroy the file.

In addition to writing the CMDSTAT records on tape, both CMDPIKUP and CMDTAPE write records of certain types to other files. (When CMDPIKUP is hung up, the problem is often that one of these files is locked and CMDPIKUP is waiting for it.) CMDTAPE will wait a while for any file it writes to, then give up, send the firefighter a message, and skip that copying operation. (If could send a message to the operator if someone thinks that would be useful.) CMDPIKUP and CMDTAPE communicate through the file CMDDIR (under SYS at UM).

CMDDIR has the format:

```
1   Tsfffttttxxxxxxxxxxxxxxxxcc:cc:cccc-cc-cc
2   FsCMDSTAT1
3   FsCMDSTAT2
4+  FsCMDSTATc
```

The first character in each line says whether the line describes a tape or a file. The second character, s, is the status of the file or tape, and has values:

"available": the file is empty and available for use by CMDPIKUP

A "active": the file currently being written in

A "available": tape is not currently mounted

B "busy": CMDTAPE is using it right now

F "full": the file is full, and CMDTAPE should write it to tape

Line 1 describes the tape currently in use: "fff" is the next file on the tape to use, Mttt is the rack number, xxxxxxxxxxxxxxxx is the BINTIME the tape was created, and cc-c... is the printing form of the date. (Note, this is binary data -- be careful when editing this file.) The tape line must be line 1; the lines describing files may be in any order thereafter. If something is marked 'B' and CMDTAPE is not running, CMDTAPE died or the system crashed. When CMDPIKUP starts up, it changes a "B" on the tape line (line 1) to "A", and any "B" on a file line to "F". Since CMDPIKUP is started when the system comes up, this covers the system crash situation, but if CMDPIKUP is started at some other time, un-busying things may not be the appropriate action (but it probably won't cause a catastrophe). CMDFDIR (under SYS at UM) contains a line for each tape file. That line contains the beginning and ending times for the file in binary and printing forms, the mount request for the Mmmm tape the file is on, and the posn of the file on the tape. (On the tape, the file DSN contains the current time, and the last character of the name of the file the data was copied from.) The lines are ordered by beginning times. CMDTAPE writes these lines when it copies files. (It puts each line in its proper place, so it does not always write at the end.) These lines contain binary data, so handle with care. Programs which read CMDSTAT data from the tapes use this file for an index. CMDPIKUP does not use it. CMDTAPE (and soon, CMDPIKUP) prints comments on the console and sends messages to the firefighter when it has problems. The next section describes these messages.

CMDTAPE Messages

All messages printed/sent by CMDTAPE are labeled "*DWB". A message is sent to the firefighter for a problem which needs fixing, but will not prevent CMDTAPE from continuing. A message is printed on the console for more serious things. Console messages require some response (mostly to intensify the message). If the message does not ask a question, a null line is an adequate response.

FILE ffffffff IS UNAVAILABLE (or LOCKED)

File ffffffff could not be used for copying when CMDTAPE copied a CMDSTAT file to tape. CMDTAPE continued anyway. The data now missing from ffffffff can be sifted from the tape file. The file should be checked to see what the problem is ("locked" means just that, and "unavailable" means anything else). NOTE: CMDTAPE does several minutes of waiting before declaring a file to be locked.

RC n FROM WRITE ON LIOUNIT n of Mmmm OF FILE p LINE nnn; CONTINUING WITHOUT IT.

A non-zero return from write occurred while copying to the copy file attached to logical unit n. mmm, p, and nnn give the current CMDSTAT tape position (use this information to sift records off the tape). The file should be checked to see what the problem is.

TAPE Mmmm HAD A TAPE ERROR. NOW USING NEXT TAPE.

If CMDTAPE gets a tape error while writing or positioning a tape, it will try again on a different drive. If that fails, it will abandon the tape and send this message. Check the tape and make any repairs necessary to make it readable. Also check the tape drive(s) -- this problem is often caused by drives having difficulty with tape marks.

TAPE Mmmm FILLED IN < 5 DAYS. PLEASE CHECK.

This may or may not be an error. It does mean than data is being generated at a much faster than normal rate. Check the data for too much of some record type (some program going crazy generating it), or for something wrong with CMDPIKUP or CMDTAPE.

CMDSTATc WAS MARKED FULL IN CMDDIR BUT WAS EMPTY.

Presumably, CMDSTATc was marked F by CMDPIKUP. So, either CMDPIKUP was confused, someone marked the file F in CMDDIR by hand, or the file really wasn't marked F and CMDTAPE was confused. The only thing to do is to tell the person who maintains this stuff about it.

LINE llll IN CMDFDIR DUPLICATED WRITING CMDSTATc

The data in CMDSTATc had exactly the same time range as some file already on the tape. This is unlikely to occur without a mistake by a person fixing something. Check whether the two tape files are really duplicates, and if so, delete one. (Delete the one which is not described in llll.)

NEW FILE FROM time OVERLAPS FILE IN CMDFDIR LINE llll

This is similar to the problem above, except that the two files are not duplicates. This message is generated only if the overlap is more that a minute (shorter overlaps sometimes occur because of the way the buffers are filled and emptied). If this occurs just once, ignore it, but if it occurs several times over a couple of weeks, tell the person who maintains this code about it.

NO ROOM IN CMDFDIR AFTER llll; CALL SYSTEMS PROGRAMMER!

This message goes to the console. Spread out the lines around llll in CMDFDIR (it is safe to renumber the file), then enter a null line at the console. CMDTAPE will make a new attempt to write out the directory line. This problem is unlikely to occur; the probable cause is that files were not written to tape for quite a while, and when they were written, they were written out of order.

ERROR ON ffffffff. CALL A SYSTEMS PROGRAMMER

This message goes to the console. A misformatted line was discovered, or an error occurred reading, writing, or locking ffffffff (CMDSTATc, CMDDIR, or CMDFDIR). CMDTAPE will not continue after printing this message (a response to the read will cause it to stop). Fix the file causing the problem as necessary, remove any B flags in CMDDIR, and start *DWB again.

CMDSTAT TAPE ERROR; PLEASE TRY ANOTHER DRIVE

This message is printed on the console, followed by a mount request. A tape error has occurred on the current tape, and CMDTAPE is about to try again. If an error occurs on the new drive, CMDTAPE will abandon the tape and ask for a new one.

ERROR: JOBDUMP AND/OR CALL SYSTEMS PGMR!

This message goes to the console. Either a program interrupt has occurred, or CMDTAPE has gotten some unreasonable return code (gre and grf are saved before printing this message). In either case, the next thing to do is to get out the code. If the console read returns, CMDTAPE will stop.

IS ANOTHER COPY OF *DWB RUNNING?

This question is printed on the console. CMDTAPE has found the tape marked B (line 1 of CMDDIR). This is normal if another copy of CMDTAPE is running (this happens occasionally during heavy load). If there is not another copy running, CMDDIR is wrong (probably because CMDTAPE terminated abnormally). In this case, the cause of the problem should be found (look back up the console log to the last time CMDTAPE ran) and CMDDIR fixed. Answer the question Y or N as appropriate. A "yes" response will cause this copy of CMDTAPE to terminate quietly; a "no" response will cause it to complain about a directory error.

MOUNT NEW CMDSTAT TAPE RING IN WITH RACK Mmmm

This message appears on the console. It is not an error; CMDTAPE has either filled or abandoned the current tape and is labeling a new one. Mount a blank tape and enter the drive name; also put the sticky labels on the reel. CMDTAPE will label the tape, release it, and issue a mount request for it. If CMDTAPE can not get the drive or write on the tape, it will print a message and ask again.

Other Problems

The most likely problem is that CMDTAPE will appear not to be running (files to be copied will pile up). Either CMDPIKUP is not starting *DWB, *DWB has stopped working, or CMDTAPE itself is fouled up (the most likely explanation). CMDTAPE can be run or debugged from any terminal; if nothing else, this will cause system messages which normally go to the wastebasket to be printed. Use the ccid SYS (at UM, anyway), PROT=OFF, and see *DWB for device assignments.

FILESAVE--SCHEDULE AND TAPE STORAGE

(See *SAV and *SVW for Filesave procedure.)

Daily filesave (*SAV) is run on the UM system at 12:30 A.M., Tuesday through Saturday. Users may be actively using the system while *SAV is running. The tapes used for daily filesave are kept in racks in the machine room. They are numbered FS0001, FS0002, etc. The daily filesave program asks for a tape by number. No copies are made of the daily filesave tapes.

Usually, no filesave of any type is run on Monday.

Weekly filesave (*SVW) for the UM system is run on Sunday morning at 7:00 A.M. The complete weekly filesave usually takes about 4 hours to run to completion. No users may be on the system while *SVW is running. There are six sets of weekly filesave tapes. These tapes are stored in the machine room and are used over and over, in order, one set per week. When one of these tapes shows a serious error, it is replaced with a new tape. The program, *FSC, is used to copy the weekly tapes. There are usually only two sets of copies, leaving several sets of the original filesave tapes with out backups. The sets of copy tapes are kept in in the safe. Every sixth week, a new set of tapes is unpacked and used for the copy set of tapes. These copies are saved and stored in the safe for a few days until they can be moved to the Bendix building. The tapes sent to Bendix are kept in a special room and are assigned "Z" type rack numbers if they are UM system tapes. The directory of "Z" tapes is kept in the Computing Center second floor machine room in a notebook near the operators' console.

UB System Filesave

The weekly UB filesave is done at 1:00am on Saturday. Daily filesave is started automatically Tuesday through Friday at 1am. A partial filesave (daily) is also done at 8:30 pm on Saturday and Sunday. Original full filesave tapes are kept in the machine room and the two latest copies copies are stored in the safe. Six week copy tapes are kept at Bendix after they have been in the safe for a few days awaiting transportation. New tapes should be used when making six-week copies from the originals.

Lists-(UM and UB)

The file-save listsings that need to be printed and kept at the Computing Center are-

> The "original" listing from each weekly filesave which should have "Info" printed on the first sheet and

The "copy" listing from the six-week filesave.

USING THE 3270 OPERATOR'S CONSOLE

THE SCREEN

The Operator job operates using a 3270, 3278 or 3066 display. The display screen contains 24 to 43, 80-character lines. it is divided into three areas as follows.

1. Lines 1 and 2 - System status area
2. Lines 3-22, 3-32, 3-41 - Message area
3. Lines 23-24 ,33-34, 42-43 - Request area

The use of these three areas is further described below.

SYSTEM STATUS AREA

These two lines will contain current system status information as follows:

line 1 - Output from the system status routine (SSRTN)
line 2 - A list of keywords alerting the operator to various abnormal conditions, followed by a list of four character device names for those devices which are not ready (i.e. which need operator intervention), followed by the five digit job numbers of any jobs whose operator console output is being %IGNOREd (see below). This line will be intensified.

THE REQUEST AREA

This two line area at the bottom of the screen is used for entering four kinds of requests:

1. New job requests (e.g. MTS *HSP)
2. HASP commands (e.g. $DISPLAY JOB n)
3. System Status commands (e.g. /T M)
4. Device commands (e.g. %IGNORE 17)

Requests may be entered in this area at any time. (see the section describing the keyboard for details on entering requests and message responses). When a request has been entered, it will be moved to the message area, and the request area will be cleared for input of further requests. Thus the request area will generally be blank. If there is an error in the request, which can be detected by the console management program, the request will be left in the request area for possible editing and re-entering.

Note: only one System Status command may be active at a time. If a second command is entered while one is already active, the active command will be terminated and the new one executed.

The time at the last screen update will also appear in the request area, at the bottom of the screen.

MESSAGE AREA

These lines are used primarily for communicating with jobs in the system, although several other forms of output will also appear here. Basically, message lines and message response lines will be inserted at various points in the message area, as described below, and lines will be removed from the top of the area to make room for more output.

The handling of each output form is described in the following list:

1. Message lines from jobs - the format will be:

   job number - 5 characters
   job name - 1 to 8 characters
   message text - 65 to 72 characters maximum unless %WRAP is ON in which longer messages will be displayed on two lines.

   Example:
   ```
   00017 MTS REMOVE TAPE FROM T901
   ```

   These lines will always be added at the bottom of the message area.

2. Message response lines - When a job requests input (via SVC READ), the message area will be scanned for the last message line from the same job. This line will be intensified, and a blank line will be inserted immediately after it, in which the response should be entered. These two lines will not be removed from the screen until the response has been entered.

   If no message line for the same job is found on the screen, or if the line found is completely blank or null the message "..." in the normal message line format will be added first. This special message will not be copied to the printer.

3. Output from System Status commands (e.g. /T) - These lines will be displayed with a minus sign prefix, instead of the job number, job name prefix. Each new line will be inserted immediately after the last output line for this command, or after the command itself if this is the first line; thus output from a System Status command will not have other messages interspersed.

4. Completed requests from the request area - All requests will be echoed by adding them at the bottom of the message area. Successful new job requests will be prefixed by the 5 character job number; all others will be displayed as is.

5. Error messages from invalid requests (e.g. "JOB NAME NOT IN JOB LIST") - If a request is in error, it will generally be left in the request area, and will not be added to the message area. The error comment will be added to the message area instead.

6. Dump and machine check error recovery messages - When a dump request or machine check occurs, the entire screen will be cleared, and the messages from these routines will appear at the top of the screen. When the dump is completed, or the machine check recovered from, the original contents of the screen will be replaced, with these messages added at the bottom of the message area. An exception to this form of operation will occur if a JOB DUMP is requested as the result of a serious error within the console management program itself. In this case the system console will be transfered to the backup console support (see below).

THE PRINTER

All lines in the message area will eventually be printed on the console log printer. If no printer is available, the behavior of the display will be the same as if a two second per line printer were used. The format used there will be the same as the format used for the 3066 console. The wider printer carriage allows 100 characters of message text, and this many characters will be maintained internally for use in printing, although only 65 to 72 will be displayed on the 3270 screen, as described above. Thus a job may put information of less immediate interest in these printed but not displayed character positions.

Lines will not be removed from the message area until they have been printed. To summarize message area handling, when a line is added to the message area, a line must first be removed; the line removed is the first one which has been printed and is not part of a message response pair.

When the message area becomes almost full, with lines which cannot be removed from the screen, no further lines will be added, except in response to requests from the request area. If the message area becomes totally full, as the result of entering several requests while the screen is almost full, the message "**SCREEN FULL**" will appear in the request area, and no further requests will be allowed, until a line is printed or a pending message response is entered.

If the printer becomes "Not Ready" the 3270 screen will be cleared and the message "devn INTERVENTION REQUIRED" will be displayed until the printer becomes ready. If it is not possible to ready the printer the console management program must be stopped and restarted using another printer (or no printer, see below).

If a fatal I/O error occurs on the printer the message "devn FATAL I/O ERROR: xxxx" will be displayed on the 3270 screen and the equivalent of a %PTR NONE device command will be executed (see below). "xxxx" represents two bytes of SENSE information.

If the printer becomes "hung" the %PTR devn or %PTR NONE device commands should be entered before the message area becomes completely full. If the message area does become full before printers may be switched it will be necessary to stop and restart the console management program using the backup console support (see below).

THE CURSOR

The cursor will normally be placed at the beginning of the first message response line on the screen, or in the request area if there are no message responses pending. The operator may move it to any position on the screen, and the system will reposition it only after completion of a request or a message response.

Input may be typed only when the cursor is in a message response line or in the request area. If it is anywhere else, typing will be ignored.

NOTE: The DUP key "*" may be used to skip a read on the console when the operator wishes to keep the "read" there, but doesn't want to tend to answer it at the time. For example; if there are several tape mounts waiting and you know that you can't take care of them for a few minutes but don't want them removed from the screen, you can place a "*" using the DUP key where you would ordinalirly answer the request. Now when you move the cursor from one "read" to another, the cursor won't stop at the "read"s with a "*" after them.

THE KEYBOARD

For the following discussion, define a *field* to be either a message response line or the request area.

Editting keys - These keys affect the display, and allow editting of input lines without program interaction:

ERASE INPUT - Clears all fields and positions the cursor to the first field. This is the recommended method for deleting and retyping a line.

ERASE EOF - Clears the field containing the cursor from the cursor position to the end of the field. This may be used in combination with the backtab key to erase a single field (or part of one) without changing others, or moving the cursor.

RESET CURSOR - Positions the cursor to the first field on the screen (which may be the same field if there is only one).

RESET - Turns off the input inhibited light, thereby "unlocking" the keyboard. Also turns off insert mode.

TAB - Positions the cursor at the beginning of the next field on the screen (which may be the one it is in if there is only one).

BACK TAB - Positions the cursor to the beginning of the field it is in, or to the beginning of the previous field if it is already at the beginning. Tab and back tab are the recommended means of moving the cursor from one field to another. For example, if the cursor is at the first message response line, and you want to enter something in the request area, back tab may be used.

RETURN - Something like a carriage return. This key has little use with the system console.

Cursor positioning - These keys (up, down, left, and right arrow) move the cursor one character position in the indicated direction. Note that the two left arrow keys have exactly the same function.

DELete - Deletes the character at the cursor position, and shifts the remaining characters in the field left one character position.

INSert MODE - Puts the display in insert (edit) mode, which is indicated by a flag on the screen. In insert mode, characters are inserted into the field at the cursor position, and the remaining characters in the field are shifted to the right to make room. Insert mode is turned off with the reset key.

DUP - Causes the character "*" to be displayed on the screen, the character X'1C' to be entered into the field, and the cursor to be advanced to the next field. This character is used at the operators' console to cause an automatic skip of the cursor past a "read" where a "*" has been placed.

FIELD MARK - Causes the character ";" to be displayed on the screen, and the character X'1E' to be entered in the field. This key has little value and should not be used.

PROGRAM ENTER, CURS-SEL, IDENT and ATTN - perform no function when used with the OPERATOR job so they are not used.

> Note: The DUP, FIELD MARK, TEST, IDENT, CURS-SEL, ATTN, CURS-BLINK, ALT-CURS and PROGRAM ENTER keys are not present on all keyboards.

Keys which cause program action

Many of these keys have a function very similar to their function when the 3270 is used as a terminal. All of them require program action, and should cause the input inhibited light to come on for a fraction of a second and then go out. If these effects do not occur, something is probably wrong.

ENTER - This key terminates all input lines. The field containing the cursor is the one which is entered. Other fields may contain input, but they will remain unchanged. The position of the cursor in the field is irrelevant. If the cursor is not in any field, this key is ignored.

CNCL - This key is used to enter a cancel for a message response line. The position of the cursor determines the line to be cancelled. If the cursor is not in a message response line, the CNCL key is ignored. The characters "**CANCELLED**" will appear on the screen. On some keyboards this key must be used in conjunction with the ALT-shift key.

PA1 - Stops the execution of any System Status command which may currently be running. On the 3278 keyboard, this key must be used in conjunction with the ALT-shift key.

CLEAR - Just rewrites the current screen contents. Probably of little use, but if the screen gets messed up for some reason, try this first. On the 3278 keyboard, this key must be ised in conjuction with the ALT-shift key.

TEST REQ - This is a pause key which can be used to temporarily stop the scrolling of messages off the screen. Pushing it again will reset the pause. If it is not reset within five minutes, it will be reset automatically. The request area will contain "**PAUSE**" while in pause mode. On the 3278 keyboard, this key must be used in conjunction with the ALT-shift key.

Program function keys (PF1 through PF12 or perhaps through pf24)

These can be used to enter various requests into the request area. These may be either partial or complete input lines. For partial lines the effect would be to enter the corresponding text into the appropriate field, and position the cursor following it. For complete lines the effect would be the same as the above plus pressing the enter key.

The initial program function key definitions are:

Complete requests:

PF1:	MTS OPER
PF7:	$BACKLOG ALL

Partial requests:

PF2:	MTS *
PF3:	STOP
PF4:	BROADCST
PF5:	$DISPLAY JOB
PF6:	$RELEASE JOB
PF8:	$BACKLOG
PF9:	$DRAIN

Program function keys may be redefined by a device command, as described below. PF10, 11 and 12 are spares. If pressed, the "spare" keys cause the cursor to be moved to the request area and any partial input already in the request are is removed. When present, PF keys 13 through 24 have initial definitions that duplicate the definitions for PF1->PF12. On 3278 keyboards, the use of PF1->PF12 requires the use of the ALT-shift key.

DEVICE COMMANDS Note: Lines which begin with a "field-mark" (±) are not translated into upper case.

A request line prefixed by a percent sign is interpreted as a device command. The only valid device commands are:

%FAST - Switches into "FAST" mode. FAST mode was added to make the OPERATOR work on certain types of 3270s (which we don't have). In FAST mode, the OPERATOR tries to minimize the number of "erase-writes" done.

%SLOW - Switches into slow mode. The OPERATOR defaults to SLOW mode.

%REFRESH - Makes sure that the screen is up to date.

%EMPTY - Empties the screen.

%PF? - Lists PF key definitions

%IGNORE jobn - Causes future message lines from job number "jobn" to be ignored (i.e. neither displayed nor printed). This is useful if a job is producing large amounts of output which is known to be of no interest. Message response requests from jobn will never be ignored. The job numbers of any jobs being ignored will appear in the system status area following any "Not Ready" device names. %IGNORE requests may only be entered for currently active jobs and a job number will be automatically accepted if the job terminates for any reason. As many as 16 jobs may be ignored at once.

%ACCEPT jobn - Message lines from jobn will once again be displayed and printed. A %ACCEPT is automatically issued for any job that is being ignored upon termination.

%ACCEPT ALL - Message lines from all ignored jobs will once again be displayed and printed.

%PFn=new definition - Redefines program function key PFn. "New definition" is an arbitrary string, of length 30 characters or less. To define a partial request, the new definition should end with an underscore ("_"). Otherwise it will be interpreted as a complete request.

%3270 devn - Switches the console to the 3270 whose device name is devn.

%PTR devn - Switches the printer to the 3284, 3286, 3287, 1052 or 1403 specified by devn. The devn may also be NONE, in which case no printer is used.

%RESTART - Directs operators' console traffic back to the 3270. This is useful if the console has been switched to the 3066 or 1052 backup support (see next section) when the 3270 support is still operational (usually because of paging problems). This command may be issued at any time without harm, and will cause two "greeting" lines to be displayed on both the backup console and the 3270 screen.

%WRAP ON - Normally, the OPERATOR job simply truncates any output line that will not fit on a single line of the display, although the complete line will be printed on the console log printer. If the %WRAP ON command is given, the longer messages will be displayed using two screen lines. An attempt is made to split longs lines at a blank.

%WRAP OFF - This command cancels the affect of a %WRAP ON command.

ABNORMAL SYSTEM CONDITIONS

The following keywords may be displayed at the front of the second line on the 3270 operator's console display to alert the operator to various "abnormal" conditions. Depending on the situation these conditions may or may not represent errors.

BACKUP The 3066 or 1052 backup console is in use. The 3270 operator's console will continue to operate. HASP commands, device commands, systemstatus commands and job requests may still be entered in the request area. Existing input requests may be filled. Output from device and system status commands will continue to be displayed on the 3270 screen, but output from HASP commands and job requests as well as output from executing jobs and all requests for input will be directed to the backup console. The %RESTART command may be used to transfer control back to the 3270 operator's console or the OPERATOR job may be stopped and restarted from the backup console.

SHUTDOWN The SHUTDOWN job has been run.

HASP-DEAD HASP has detected a serious error within itself and has given up. It will need help if it is to run again without reIPLing the system and even that may not help.

HASP-PCE One or more HASP PCEs are "frozen" and will remain that way until fixed by a programmer. This message, while less serious than the HASP-DEAD message, still indicates an error condition within HASP.

...-HELD Either HASP has not been started or the HASP command $HOLD *...* has been issued.

EX-HELD Either HASP has not been started, execution has not been released after running *HSP or the HASP command $HOLD EX has been issued.

NTM No tape mounts. /MODify NTM 1

UAM The system has been placed in unattended mode using the system status command, /MODIFY UAM 1.

AB, CC, AD, AE, or AN
When one or more of these names are displayed it indicates that some abnormal condition in one of the PCPs, the data concentrator or the UMnet PCP has been detected by the system status master job. Running *HUH should explain what the trouble is.

Following any keywords that may appear may be a list of four character device names of those devices that are "not ready" and which need operator assistance. Following any not ready device names may appear a list of five digit job numbers for those jobs whose output to the 3270 operator's console is being ignored.

3066 or 1052 SUPPORT AS BACKUP

If the 3270 support fails to respond, or dies some horrible death, switch to the backup console by selecting the "X" function of the "OP" menu of the macrocode. Switch back to OPER mode to use the backup console.

The job name is OPERATOR, and no parameters are required. If it is desired to override the default devices, a 3270 should be given as the first parameter, and a 3284, 3286, 3287, 1052, 1403 or NONE as the second parameter. If a 1403 is to be used as the console log printer the UCS buffer must have been previously loaded using the PN or TN job. If only one parameter is given the default printer list will be used. If the 3270 specified (by default or otherwise) is not available the OPERATOR job will terminate with the message "NO 3270 AVAILABLE" or "devn NOT AVAILABLE". If the 3270 is available, but the printer is not, the OPERATOR job will continue but the message "NO PRINTER AVAILABLE" or "devn NOT AVAILABLE" will appear on both the 3270 screen and the backup console. In this case the %PTR device command may be used when a printer becomes available. When the OPERATOR job is successfully started two greeting lines giving the time, date, machine model and serial, the device names of the display and printer and the device type of the printer are displayed at the backup console and on the 3270 screen.

Before restarting the OPERATOR job all pending requests for input through the backup support console must be satisfied. If this is not done it will not be possible to pass operator console input to these jobs without transfering to the backup support again. If it is not possible to restart the OPERATOR job for some reason system operation may continue using the backup support, but any requests for input that were made to the 3270 operator's console cannot be satisfied. The only alternatives to running from just the backup support are contacting a systems programmer or IPLing.

Problems with the 3270 operator's console support should be called to the attention of the programmer responsible for the OPERATOR job (currently Paul Pickelmann at UM).

CONSOLE(BACKUP) SUPPORT CHANGES FOR MTS-ES3090/300J

When a task requests a reply to a message, mts hangs a read on the operator's console and waits for a reply for that task from the operator. This has been changed in mts-ES3090/300J so that these requests are queued by the supervisor which, after having queued a request, puts out a message on the operator's console giving all the tasks numbers of the tasks that have requests pending.

```
     EXAMPLE:  PENDING 10 15 12
```

The support for the 3066 display console is such that it looks as if a read is always hanging on the console (i.e. The operator doesn't have to press a REQUEST button before typing a line). Thus the 3066 console is active or alive at all times. Input is always entered on the line 2nd from the bottom of the screen. The bottom line of the screen is always used to display the list of task numbers of tasks that have requests outstanding (pending). The input lines and messages scroll upwards to the top of the screen. A one page buffer of the these input lines and messages is kept by the supervisor so that they can be logged in a file by a special logging task.

The support for PRINTER/KEYBOARD consoles such as the 3215 is similar to the old 1052 support in that the operator must press the REQUEST button to force the supervisor to hang a read on the operator's console.

The new method of replying to a request is as follows: the operator types an octothorpe (#) followed by the tasks number of the task to which he is replying , followed by a blank and then the reply. If no tasks number is given after the octothorpe, the reply will be given to the task whose task number is first on the list of pending requests i.e. If there is only one outstanding request no task number need be given (but don't forget the octothorpe!).

```
     EXAMPLES: PENDING: 12 15

               #15 OK RF9999 T904     (this would reply to a
                                   tape mount for task 15)

               #OK RF9999 T901     (this would be a reply to a
                                   tape mount for task 12)
```

NOTE: If the first character of a reply is NOT numeric, the blank after the task number may be omitted.

The operator can cause the supervisor to produce its list of task numbers with pending requests at any time by typing %R.

Phone (Pager) Numbers

Note: Every pager has an ordinary telephone number. After calling a pager's number from a touch-tone phone, you can enter numbers using the phone keypad that are then transmitted to the Digital Display pager and shown on its screen. By entering the phone number where you can be reached, the person with the pager will know where to return your call.

To reach someone,

1. Dial the pager number from a touch tone phone
2. LIsten for the ring and a tone
3. At the end of the tone, enter up to 23 numbers to be displayed on the pager (your phone number)
4. If you enter a wrong number, press * three times, then re-enter all numbers to be displayed
5. Press the # symbol and hang up

The following currently have pagers assigned to them:

```
             Baker, Jane - 996-6822
             Deneau, Carol - 996-6174
             Donnelly, Steve - 668-3692
             -----------------------------
$Run Sys.:Fire for list of Firefighters currently on duty.
       **    Fire Fighter - 930-8086
       **    Systems Manager - 930-8976 (See next line)
**Only F.F. allowed to "beep" manager. Don't call manager unles
   1) Firefighter has been called but can't fix problem or,
   2) Firefighter hasn't responded within 20 minutes.
             -----------------------------
             Gluski, Kari - 930-8976
             Gold, Steve - 668-3954
             Hankins, John - 996-6759
             Jaworski, Dennis - 996-6282 (Voice)
             Kissling, Gene - 996-6823
             Knox, Jim - 996-6504
             Lonc, Ken - 668-5699 (Voice)
             Martin, Dave - 668-5083
             Masten, Deb - 761-0881
             Mitchell, Lee - 996-6806
             Myers, Jack - 761-0920
             Olsen, Steve - 930-8074
             Palms, Andy - 747-5145
             Public Facilities Techs 761-0967
                                     761-0988
                                     930-8088
                                     761-0979
```

Rowe, Fred - 668-5480 (668-3429,668-3443)
Sullivan, Jim - 996-2802

DELIVERY

(Printer output,Plots,Diskettes,Tapes,etc.)

The file *DELIVERY, which is changed from time to time, contains information about delivery Diskettes, Magnetic tapes, output, etc.

A messenger delivers page printer output to NUBS and BSAD, leaving the Center at 8 am, 11 am, 1 pm and 3:30 pm, Monday through Friday. Magnetic and paper tapes and floppy disks and plots are delivered to NUBS (but not BSAD) according to this same schedule. Magnetic tapes and plots are delivered to UNYN.

Because of budgetary and staff restrictions, deliveries are limited to the above schedule. The schedule is tentative and will change depending on the needs of the user community and availablilty of Computing Center staff and funds.

UM PAGE PRINTER DELIVERY

To have all page printer output delivered to NUBS (or BSAD) issue the following MTS command before using *PRINT* for the first time in a terminal session:

```
          $SET DELIVERY=NUBS
```

To have a given *PRINT* job delivered to NUBS (or BSAD) use the following MTS command before the job is released:

```
          $CONTROL *PRINT* PRINTER=PAGE DELIVERY=NUBS
```

To have a batch job printed on the page printer at the CNTR and delivered to NUBS (or BSAD) use a signon command similar to the following:

```
          $SIGNON ccid PRINTER=PAGE DELIVERY=NUBS
```

Because there is currently no TN (mixed case) printer at BSAD the delivery option can also be used to have TN output printed at the CNTR and then delivered to BSAD.

DELIVERY OF PLOTS

To have a plot delivered to NUBS append the DELIVERY keyword to your plot request when running *CCQUEUE. For example:

```
          $RUN *CCQUEUE
          plotfile DELIVERY=NUBS
          $ENDFILE
```

or issue the following command before you run *CCQUEUE:

```
          $SET DELIVERY=NUBS
```

DELIVERY OF TAPES AND FLOPPY DISKS

To have a tape or floppy disk delivered to NUBS call 764-9109. Be sure to give the tape's rack number and external tape id.

Tapes and floppy disks can also be retrieved by mail, but there is a service charge for this service to cover postage and handling. $COPY *TAPERETRIEVE and *TAPESUBMIT for more information.

DELIVERY TO DEARBORN AND FLINT

Delivery of plots and page printer output to the Dearborn (DBRN) batch station is available. A messenger picks output up in Ann Arbor on Mondays, Wednesdays and Fridays.

Delivery of plots and page printer output to the Flint (FLNT) batch station is planned. This service will be on an "as needed" basis, so users should contact the Flint station when they have output they wish to have picked up in Ann Arbor.

NOTE: When a user comes to pick up output with a DELIVERY code, all of the output with that code should be picked up, i.e., a user just doesn't hand someone a receipt for one or two jobs out of the box. All or none of the jobs are picked up each time.

DELIVERY codes (06/04/82)

```
                              NUBS
                              BSAD
                               ISR
                           ISR1...ISR7
                              HSRI
                              DBRN
                               EPA
                               VRI
```

NOTE: See section on TAPE and DISKETTE mailing later on in this manual.

DISK STORAGE

NOTE: See *DSK and DMGRSTAT earlier in this manual. These are disk manipulation programs.

DASDI

The pupose of this program is to label, re-label, or format disk packs according to either the VAM2 or VAMX conventions. The following devices can be DASDI'ed: 2301, 2311, 2314, 3330, 7330, 3340, 3350, 3370, 3375, 3380, 6280.

Usage:

```
     mts *dsd (Promts for input at operators' console)

macrolib mts:cmdmaclib

macrolib file:cmdmaclib

dasdi
```

Input data consists of the following operands starting in column 1:

```
     Dxxx MTSxxx {VX|V2} {pvn#|PAGING|PRIVATE} [optional pars]
```

Parameter descriptions follow. In any place a number is called for, a decimal number may be given, or X'hex-number'.

pvn --> public volume number (if it is to be a public volume)

PAGING --> if it is to be a paging volume.

PRIVATE --> if it is to be a private volume.

LO or
LABELONLY --> if the volume is to only be labelled or re-labelled (as opposed to being formatted).

IPL or
IPL=nnn --> if it is desired to leave "nnn" pages of IPL area starting at the front of the pack (as well as formatting the pack). The IPLINIT program can place a core image of the IPLREADER program in these pages. (IPLREADER is the program which decides which system to load and loads it into the bare machine.) The area is

reserved by generating the necessary DSCBs to describe the IPL pages, writing them onto the disk, and marking the DSCB and IPL area pages properly in the PAT. If "nnn" is not specified, the maximum number of pages which can be described by a single full DSCB page is allocated - 968 (=38+15*62).

IPLSTART=nnn --> used in conjunction with the IPL keyword, the "nnn" value determines at what page the IPL pages should begin. This defaults to the first available page on the pack if not specified.

PAGES=nnn --> format the pack as if it had only "nnn" pages.

PATA=nnn or
PATSTART=nnn -->

PAT should start at the given page address.

CLEARPAT --> construct and write PAT and IPL DSCBs - useful only with the LO option.

NOSLOW --> override the PAR=SLOW (see below) option for this DASDI operation only. This is the default

SLOW --> perform a 50 millisecond wait between formatting writes. This gives other tasks access to the disk's control unit and allows a pack to be DASDI'd on a running system.

The following parameters should only be used for special situations:

WHA --> read and write "home addresses" on the volume. This is almost never necessary on "modern" disks and is very dangerous to do since it may cause information essential to proper error recoverey to be lost. This parameter is necessary on brand new disks, however, to insure that the alternate tracks are correctly formatted. If this parameter is not given, then the VIRGIN and the ALTERNATES parameters are illegal.

VIRGIN --> try to read "home addresses" but don't complain if they can't be found. If the label is unreadable when this option is specified, DASDI will continue on.

ALTERNATES=nnnnn
ALTERNATES=(nnnnn,mmmmm)

ALTERNATES=NORMAL
ALTERNATES=KEEP
ALTERNATES=NONE (the default)

--> DASDI will also format tracks which are marked as alternate tracks in the home address, even though MTS does not use alternate tracks.

The first and second forms designate the range of the relative track numbers (starting at relative track 0) for tracks to be assigned as alternates. Note that if "ALTERNATES=KEEP" is also given, then any other track which is already an alternate will be kept as an alternate in addition to the track range explicitly specified.

You cannot explicitly specify alternates for 3340/3344/3350/3370 devices. You can either specify "ALTERNATES=KEEP" or "ALTERNATES=NONE" for these devices.

You cannot specify anything other then "ALTERNATES=NONE" unless you also specify "WHA".

"ALTERNATES=NORMAL" means that the track range for the alternates is as is found in the relevant component description.

Note: The underlined portions below represent responses entered by operators or others.

EXAMPLE:

```
$RUN FILE:DASDI
Execution begins
MTS DASDI program (EB265).  Enter input line:
Dddd llllll Vx #/PAGING/PRIVATE pars ...
D354 MTS009 VX 9
D354 CURRENTLY LABELLED AS "OLD009".  PLEASE CONFIRM.
OK
PAT TO BE WRITTEN ON PAGES X'yyyyyy' THRU X'zzzzzz'
Enter next input line:
Dddd llllll Vx #/PAGING/PRIVATE pars ...
D355 UNUSED V2 PRIVATE
LABEL IS UNREADABLE.  ENTER "OK" TO CONTINUE.
OK
PAT TO BE WRITTEN ON PAGES X'yyyyyy' THRU X'zzzzzz'
Enter next input line:
Dddd llllll Vx #/PAGING/PRIVATE pars ...
D340 SPOOL1 V2 PRIVATE LO
D340 CURRENTLY LABELLED AS "OLDSPL".  PLEASE CONFIRM.
OK
Enter next input line:
Dddd llllll Vx #/PAGING/PRIVATE pars ...
$endfile
Execution terminated
```

The program, FILE:DASDI is used to initialize disk packs (3330, 3350, 3370, 6280, 3380, ...).

Description of Input

Column 1, 4 character drive name

Column 5, blank

Column 6, 6 character volume serial number

Column 12, blank

Column 13-14, VX

Column 15, Blank

Column 16, public volume number

Normally, the public volume number would correspond to the volume serial number, but not necessarily.

Options may follow the Public volume number separated from it by one blank:

LO (just write label)

IPL (reserve 150 pages for IPL writer)

SLOW (Pauses between tracks. This option should always be used when the system is busy to keep DASDI form hogging the system.)

An end-of-file terminates the run.

EXAMPLES:

```
$signon FILE

$set macros=on

macrolib cmdmaclib

dasdi

D830 MTS001 VX 1 IPL
```

```
     D838 MTS002 VX 2
```

If asked for confirmation, type, "OK" if the Volume label is correct.

```
     $endfile
```

Anything else will cause the preceding input line to be ignored.

SLOW should be specified on the DASDI command if DASDI is used during the day. A 50 MSEC. TWAIT will be done between tracks.

NOTE: The IPL parameter above means that the pack being dasdied is one to be IPLed from.

DISK COPY

PURPOSE:
To copy a disk pack to another disk.

(Last updated: Apr 06, 1992 by Tom Valerio)

Ahead of time you should:

1) Find an unused volume to copy to by checking the file W225:DISKS. The file contains the list of production volumes for various systems (UM and VM) followed by the list of idle disks (followed by a list of disks and associated notes for disks that we got rid of long ago).

2) Attach the MTS and FILE macro libraries.

```
   >macrolib mts:cmdmaclib
   >macrolib file:cmdmaclib
```

3) Use VAMREC to insure the disk you've chosen is not a production disk. (That is, make sure the internal label on the disk is of the form XXX###.) A brand new disk may have a label which VAMREC cannot even read.

```
     >VAMREC
     Volume: D200
          Volume=XXX200        Device=D200        Type=6280
          Density=Single
     Cmd: DISPLAY LABEL

Label: Volume=XXX200 Label=VOL1 Type=6280 Format=VX PVN=1
```

```
  PAT:     start=50400     (X'C4E0')     length=100800
(X'000189C0')
  Catalog pointer=X'00000000' Error byte=00
       ...
Cmd: STOP
```

4) Check the PAT (Page Address Table) starting address for the disk you wish to copy. Assume we're copying MTS005.

```
$Run Sys:Dsk
list mts005
5  MTS005 on D403   6280   3339 100800  50400
stop
```

In this example, the disk's PAT (Page Address Table) starts at the default location for a 6280 device, 50400. The PAT starting address is the last number on the line listed.

5) Use DASDI to format the unused disk. Name it OLD005 and use the SLOW option if the DASDI is occurring on a running system. DASDI takes 10-20 minutes to run; using SLOW will double or triple that time.

```
>DASDI
MTS DASDI program (AR267).  Enter input line:
Dddd llllll Vx #/PAGING/PRIVATE pars ...
D200 OLD005 VX 5 SLOW
```

If the unused disk has never been formatted before (never with our DASDI, that is), then you should use the WHA (Write Home Address) option.

```
D200 OLD005 VX 5 SLOW WHA
```

If the disk you want to copy has a non-default PAT address, you should format the unused disk with the PAT starting at the same address.

```
D200 OLD005 VX 5 SLOW PATSTART=nnnnn
```

The morning of the diskcopy:

6) Do a normal system shutdown all the way though the Shutdown All step, and also completely shut down the Resource Manager. Run the Startup job program and start an MTS task either on a terminal or on the operator console.

7) Signon and attach the MTS and FILE macro libraries by

```
     >macrolib mts:cmdmaclib
     >macrolib file:cmdmaclib
```

8) Remove the MTS volume from the Disk Manager

```
     $Run SYS:DSK
     NODMGR MTS005
     STOP
```

9) Copy the disk and swap the labels. This operation takes from 10 - 20 minutes. If there were any errors or relocations during the diskcopy that you do not understand, you should answer "NO" to the prompt to swap labels and get some help.

```
 >DISKCOPY
  Enter "FROM" device type (DISK/TAPE)
   DISK
  Enter device and volume label (Dxxx MTSyyy)
   D508 MTS005
  Enter "TO" device type (DISK/TAPE)
   DISK
  Enter device and volume label (Dxxx MTSyyy)
   D200 OLD005
  Enter options (SWAP/IPL)
   SWAP
  Ok to swap labels?
   OK
  Enter "FROM" device type (DISK/TAPE)
   STOP
```

10) Return the MTS volume to the Disk Manager.

```
     $Run SYS:DSK
     DMGR MTS005
     LIST MTS005
     STOP
```

11) Bring the system back up normally. It is *not* necessary to re-IPL the system either before or after shutting the system down for the sole purpose of doing a disk copy. It is only necessary that there be no disk activity involving the disk(s) that are being copied. The easiest way to ensure this is to just kick everybody off, it is by no means the only way. System IPL's should only occur if system crashes or if there is a new system to install.

12) Send a message to Steve Donnelly about the disk just copied. He will change the label cards in the plastic pouches on the physical disk boxes on the third floor to reflect the new labels.

13) Put a notice in the Systems Changes *Forum conference.

FILEINFO

The program in the file ACC.:FILEINFO can be used to determine how much disk space would be retrieved if the DEADFILES program were run on a given date.

The program ACC.:FILEINFO often takes about 15 minutes to run to completion.

To use the program from the operators' console:

- $signon ACC.

- $run acc.:fileinfo

The FILEINFO program supports the following keyword parameters-

[STAFF] causes a list of file staff usage to be produced.

[[DATE=]mm-dd-yy] specifies the last expiration date to be used for determining if files are dead.

[DEAD] causes all active files to be grouped together and dead files to be listed according to first letter of project number instead of vice versa.

DEADFILES

Schedule and Miscellaneous Notes

The DEADFILES program is usually run on the last Thursday of each month. There are a few exceptional months. During these months, the DEADFILES program is usually run on different days depending on current fiscal considerations. On any month, the DEADFILES program may be run at an unusual time if necessary, e.g., when the system is running low on file space. Always check with the Business Office before running this procedure.

DEADFILES is a procedure for removing files, user file catalogs, and entries in the *Restore, $Message and *Userdirectory databases that belong expired CCIDs.

This procedure, at present, is usually run on the last or second to last Thursday of each month during the 4pm-midnight shift. The "date used" is usually the second to last day of the previous month. the Deadfiles routine can be run at other times if the system is low on disk space. Always check with the ITD Accounts Office before running this procedure.

If any problems occur while running the DEADFILES procedure, check with a senior member of the systems group (Tom Valerio) before going on with later steps.

DEADFILES - The Procedure

1) Signon from a terminal or at the operators' console using the CCID RSTR.

2) Make sure that the permanent files that will be needed exist and are empty by using the following command:

```
$source deadf_start
```

   This will turn command macros on, attach the appropriate command macro libraries and then make sure that all the necessary files exist and are empty.

3) Use the Scanfiles program to create lists of the ccids, files, message system mailboxes and user directories to be removed.

```
df_scan mm/dd/yy
```

   where mm/dd/yy is the date specified by the Accounts Office, for example June 30, 1983 would be entered as 06/30/83 while July 1, 1983 would be entered as 07/01/83. Ccids that expired on or before the date given will have their files, user file catalogs, *Restore, $Message and *Userdirectory data removed from the system. If no date is given, yesterday's date will be assumed by default.

The last thing printed by this step will be the number of files to be saved. Make a note of this number for comparison with the number printed by the next step.

4) Use the Deadfiles program to save the files that are to be purged from the system on either 9-track 6250 bpi magnetic reel tape or on 18 track IBM 3480 cartridges:

```
df_save
```

When asked, enter the name of the tape drive to be used.

5) Compare the count from step 3 to the counts printer in step 4. If the totoal number of files doesn't agree with the count of files from step 3, STOP!

6) Use the filesave-copy job to make a copy of the magnetic tape that was created in step 4 by entering "mts *fsc" from the operators' console. The stream name to use is "FILES." - be sure to include the trailing perioed in the stream name. When *FSC asks if a listing is desired, reply YES.

7) Compare the count of the number of files saved in step 4 with the count printed by *FSC in step 6. If they don't not agree, STOP!

8) If there have been any problems in steps 1 through 7, DO NOT go on with the remaining steps, instead, report the problems to a senior member of the systems group.

9 Now, destroy the files that belong to the expired ccids.

```
df_destroy
```

10 Now, destroy the user file system catalogs and entries in the *Restore filesave directory for ccids that have expired:

```
df_ucatdestroy
```

11) Now adjust the disk accounting information for the ccids whose files have been deleted:

```
df_fix
```

12) Now, remove any names associated with the expired ccids from the directory used by *Userdirectory:

```
df_userdir
```

If error messages are printed, show the ouput to a member of the systems group or the person responsible for *Userdirectory. Even if there are errors in this step, you can go on with the next step.

Show any errors to a member of the systems group or the person responsible for the $Messagesystem. (The message "search failed" is not an error).

14) If there were no errors or other problems in any of the aboves steps, you can issue the following command to destroy the files used and to reduce their size.

```
df_end
```

If there were errors, skip this step.

DIRECT ACCESS STORAGE DEVICE ERROR CODES

NOTE: In another part of this manual are programs which may be used to print listings and summaries of DISK ERRORS and UNIT CHECKS. Two of these are MTA.:PUC and SYS.:EREP.
The following Disk Error information always applies.

An explanation of the error messages which are displayed on the operators console is given below. Each error produces a three line report of the error which gives device name, the sense data associated with the error (which precisely describes the error in machine terms), and a short descriptive message giving the nature of the error. A prototypic message is as follows:

```
aaaa error b cccccccc dd eeeeeeee eeeeeeee eeeeeeee eeeeeeee
eeeeeeee eeeeeeee ffffffff gg
```

Each of the underlined areas vary depending on which device declared the error and what type of error it was. These are described below:

aaaa designates the device which detected the error. Notice that this appears at the front of each of the three lines.

b is a one character code which uniquely describes the type of error. See list the list below.

Unit Check Error Codes

Stop

Code	Meaning
0	Insufficient sense data (less than the full 24 bytes)
1	All relevant sense bits are zero
2	Second equipment check
3	Command reject
4	File mask violation (unexpected)
9	Bus-out check
A	Uncorrectable data check
B	Overrun - 10 tries
C	Unwanted track overflow (CKD) or block size exception (FBA)
D	Unexpected end of cylinder
E	Intervention required (2301 or 3805)
F	Overflow incomplete
G	Unable to define extent parameters (FBA)
H	Check data error (FBA)
I	Incorrect length
J	Invalid device type
K	No record found
L	Channel detected error: program check, protection check, channel protect check, interface control check or offline control unit
M	Low spares on 3805 or 3825
P	Track condition check (3340 or 3344)
Q	Intent violation (3380)
R	Seek check
S	PCI fetch mode and correctable data check - 3330/3350

T Intervention required

U Channel data check

V Chaining check

W Environmental data present

X Truncation error during data check correction

Y Program interrupt looking at user's CCW list

Z Incremented seek address past a file mark boundry

# Regular recording of buffered log of activity on this disk

cccccccc gives the hexadecimal value of the unit check routines' input switches. This is only of use for programmers which work on the unit check routines themselves.

dd gives the hexadecimal value of the supervisor's sense flag. This value is interpreted by the unit check routines to determine the cause of some kinds of tape errors. It is mostly of interest to programmers working on the unit check routines, but also required to interpret channel detected errors (described below). If this value is anything other that 01, the sense information will be unreliable.

eeeeeeee ... is the hexadecimal sense data associated with tape error. Again, this is interpreted by the unit check routines to determine the type of error and how to recover from it. This information may be of use to both tape vendor customer engineers (if they are attempting to correct a hardware problem) and also to programmers working on the unit check routines themselves.

fffffffff is the hexadecimal value of the first seek argument (cylinder and head) used in the channel program.

gg is the hexadecimal value of the channel command the drive was trying to perform when the error occurred. Typical values for this field are:

07 - seek
1B - seek head
23 - set sector
1F - set file mask
31 - search JD equal

05 - -write
06 - read

Disk Problems-UB

If UB hangs or crashes or if the UM system shows a long Standing I/O to the disks, do the following before doing anything else.

1. Write down the number of the disk(s) which are not ready.

2. Write down the status of the system from the console (clocks on/off, process/stopped/wait).

3. Look at the channel activity lights on the disk control units that hold the "not ready" devices and write them down.

If the system is not in a "clocks off" situation, try to contact Diane, Steve Burling or Kevin Bosley and have them take a look at the problem.

If the system is in a "clocks off" situation and none of the above people are available, go ahead and reload, but run *CTD before letting the lines out.

DUMPS--PRINTING

PRINTING JOBDUMPS AND SUPERDUMPS

The program in the file MTS.:PRINTDUMP is used to print the jobdump and superdump tapes produced by the supervisor. Superdumps are taken automatically by the supervisor when it detects a fatal error within itself; all of real memory is written on the tape and system operation terminates. Jobdumps are usually initiated by operator request (typing "JOBDUMP n" to force a dump of job number n"); all of real memory and the job's virtual memory are written on the tape and system operation resumes.

System dump tapes are 9-track and labeled "DUMPxx", xx being the dump tape number. This program reads the tape on unit 1 or 0, reads commands from SCARDS and prints the dump on SPRINT. SPRINT should generally be assigned to a printer (or *PRINT*) since large amounts of output are usually produced.

Output is controlled by commands as follows:

```
CORE start_address <end_adress> <ST=segtable_address>
        This dumps real memory
        Addresses are hex byte address

DISPLAY
        This is the same as CORE

EXPLAIN
        This is the same as HELP

FOLLOW start_loc length <offset> <st=segtable_address>
        This follows a chain in REAL memory
        "start_loc" is the address of a fullword that
        points to the first item in the list
        "length" is how many bytes to dump for each item
        "end_address" is the offset of the list link from
        the start of the item.  Default end_address is 0.

HELP
        This is the same as EXPLAIN

ICTAB OPERAND
        This displays various aspects of the IN-CORE
        file table.  Legal operands are:
           "LSTAT JOB xxx" or "LSTAT FILE yyy"
           "LOCATE FILE yyy" (displays ICTAB entry offset
           for file "yyy")
           "DUMP" (to dump the whole thing)
           "TRACE" (to give a file-by-file trace)
```

```
            "VERIFY"  (to  perform  a  validity check on the
      table)

     INTERTASK JOB <TASK_NUMBER | ALL> or
          INTERTASK NETWORK <NETWORK_NAME> | ALL> or
          INTERTASK dumps intertask control blocks

     JOB <task_number | NONMTS | all>
             this dumps the job table, page tables, etc.
             "task_number" is decimal

     LOADCLASS
             This prints the status of each task (ala  taskstat)
          and then prints a categorized table (a la loadclass).

     MTS
             This calls MTS

     POOLS
             This dumps supervisor storage pools

     SUPERVISOR <FULL | CHECK>
             This dumps the PSA(s) and the supervisor error
             save region "sverr" and so on.
             "FULL" causes all supervisor pages to be dumped
             "CHECK"  invokes  a  validity  check  of supervisor
          management.

     TRACE
             This prints the trace table

     UNITS unitspec
             "unitspec" is any units job command
             dump output is the same as from the UNITS job

     VM (first_page <last_page>) or VM <MTS> <GSFS> or just VM
               These dump virtual memory
               "first_page" and "last_page"   are   hex   page
               numbers,   e.g.,   "400"  means  address  address
               starting at 40000.  MTS dumps storage in a format
               convenient for mts debuggers.  GSFS causes a dump
               of storage control  blocks.
```

Any input line beginning with a "$" is assumed to be an MTS command and is passed to MTS to be executed.

All commands begin in column 1 with parameters separated by one or more blanks.

Attention interrupts are handeled as follows: a message is printed on SERCOM acknowledging the interrupt. If PRINTDUMP IS doing something that can't be interrupted, it will print a message saying so and continue. A second interrupt will be processed by MTS. When PRINTDUMP can be interrupted, it will prompt for an input line from GUSER. This input lines can be RESTART to continue the command interrupted, or any PRINTDUMP command to terminate it.

Example (in batch):

```
$MOUNT DUMP17 9TP *DUMP* VOLUME=DUMP RING=IN 'DUMP #17'
$RUN MTS.:PRINTDUMP 0= or 1=*DUMP*
TRACE
VM
CORE 7F10 A768
```

OPERATOR FILE *DMP

This *-file is used by the operators to log all dumps and generate standard format listings of all MTS job-dumps.

A *BATCH* job is created to print out the dump at deferred priority (MTS:PRINTDUMP takes a long, long time). For MTS jobs a "full" dump is taken. For other types of jobs, a more abreviated dump is taken; enough, it is hoped, to give you a start at figuring out what went wrong. File DUMP:DUMP.CMDS contains the actual MTS:PRINTDUMP commands that are used for each job-type.

The receipt number of the *BATCH* job created by *DMP should always be recorded in the dump-log notebook.

A log of all dumps is kept in file DUMP:LOG. In addition, file DUMP:TAPES has an up-to-date list of the current un-released tapes.

ACCOUNTING-MONTHLY LISTING AND FILE CHARGE UPDATE

<u>NOTE</u>: *PRINT* must be available before the following procedures may be dealt with or the output may show up on the operators' console.

```
     $sig w035

     $sou getfilechg
```

(One or more tapes will be requested by the system. The tape(s) will be "Daily MTS Statistics File Charges".)

```
     $sig


     $sig acc.

     $sou dlist
```

(This produces a listing of Computing Center Accounts and will request tapes of the type, ACC01, ACC02, etc.)

```
     $sig
```

<u>NOTE</u>: The GETFILECHG program should always be sourced <u>before</u> the DLIST prgram.

FIRE FIGHTERS' GUIDE

This section is included in the operations manual since the duties described here are closely related to operations though the people mentioned are with the systems group.

1) FIRE-FIGHTER JOB DESCRIPTION

A fire-fighter is supposed to handle as much as possible of the day-to-day, fix-it, busy work jobs which otherwise tend to eat up the time of the systems group people.

The tour of duty lasts two weeks, overlapped, with a new person starting each Monday morning.

Specific duties include:

- being on call 24 hours per day to investigate system problems, and whenever possible, to resolve them without involving other systems people.
- printing and routing jobdumps
- keeping current system loader maps posted
- keeping the systes group advised of any problems that come up.

2) TOUR OF DUTY CHANGE

The following steps are used to identify offlicial fire-fighters for the next tour of duty:

a) $Edit the file SEG2:MONITORS to change the $Messagesystem SEND command recipient list to include your name(s). This command is used automatically to send you a message whenever the system map has changed after a reload so you know that new maps should be printed. Three copies of the system map, residing in the fileSEG2:S2MAP, should be mad and placed in the machine room, on the third floor near M. Alexander's office, and one sent to Huron Towers. A copy of each old map should be saved on the "old maps" shelf in the "systems room" (DWB's old offlice) on the third floor.

b) Put a note above the operators' console informing the operators that you are the first to be contacted in the event of systems trouble. Include phone numbers.

c) Put a bulletin into MTS.:COUNSEL announcing that you are now the current fire-fighter(s) and that questions about MTS and the system should be directed to you first.

3) HANDLING JOB DUMPS

If you are in charge of handling job dumps, you most important resonsibility is to make sure that all of the dumps get looked at. The dump log book contains a complete history of the dumps that have been taken; it should be checked frequently. Make certain that it is kept up to date. This log book is kept in the second floor machine room near the dump tapes.

Dumps of non-MTS jobs should be routed to the person in charge of the particular component in question. For instance, M. Alexander gets supervisor dumps, J. Hansen, HASP dumps, etc. Dumps of MTS jobs should be scanned to determine as specifically as possible where the problem was, e.g., CLS, DSR, file system, etc., then routed to the appropriate person. If you don't know how to read a dump listing, contact someone who does; that's the best way to learn.

Whenever there is a dump, the operators will use *DMP to produce a *batch* listing of it. They should also log it in the "dump log" book. Sometimes, DUMP:DUMPER ot MTS.:PRINTDUMP will be used to print a dump immediately in crisis situations.

The file DUMP:TAPES contains a line for each current, unreleased dump. The batch job initiated by *DMP writes a line to this file each time it prints a dump. The receipt number can be used to pick up the listing. When the dump is routed, you should edit in the name of the "routee" over the receipt number; then when the tape is released, delete the line from the file.

If there is ever confusion about the contents of a dump tape, the operator job *FLB (described in the operator's manual) can be used to quickly print out the dump header information. The program MTS.:PRINTDUMP will do the same thing.

The dump tapes in the machine room are kept in two rows. The "released" tapes are on top. When a new dump-request comes up the operators will use one of these tapes for the dump. The "un-released" tapes are on the bottom row. Whenever the operators write a new dump-tape they put it in the "un-released" row. Once a dump-tape has been analyzed and is no longer needed, it is "released" by moving it to the top row for re-use and marking it "released" in the operator's dump-log book. The operators NEVER release tapes; that is your responsibility.

Aside from crisis situations where a dump was printed (and perhaps even released) right away, your job each morning is to pick up the new bunch of dump listings and figure out what to do with them. The *BATCH* receipt numbers will be in the dump-log book. For non-MTS jobs (e.g. HASP, PDP, SUPERVISOR) usually little more is done than making sure the person responsible for that component knows about the dump and receives the *DMP-produced listing. For MTS jobs our goal is that we will all learn to do most of the detective work in figuring out what went wrong.

4) BACKGROUND INFORMATION ABOUT JOBDUMPS AND DUMP TAPES.

In many situations when something goes wrong with MTS or UMMPS, a job dump is taken. In producing a dump, the supervisor writes all real memory and all virtual memory for the job being dumped in an internal format, to a dump tape. Other miscellaneous information is also included. The rack numbers for the dump tapes are "DUMP01" through "DUMP36". The volume names, and the external label, always match the rack number.

The program, MTS.:PRINTDUMP is used to scan the dump tapes for various types of information and print a formatted listing. For example, it has commands to print out the job table and various sections of real or virtual memory. In addition, PRINTDUMP is smart enough to print a nice, formatted listing for MTS jobs. The best documentation for MTS.:PRINTDUMP is its own EXPLAIN command.

A garden variety listing of an MTS job-dump is produced as follows:

```
$Mount dump17 9tp *dump* volume=dump17 ring=in
$run mts:printdump 1=*dump* p=100
vm mts
$Endfile
```

Note: Don't forget the page limit. Depending on how much VM is being used, PRINTDUMP can produce zillions of pages of output. (100 pages is usually enough room to dump all of the system segment and most of user memory.)

Note: Always mount dump tapes with "ring=in". This makes things easier for the operators.

Note: This "long-hand" method of typing in the $MOUNT and $RUN command is rarely used. It is easier to run program DUMP:DUMPER which takes care of this stuff automatically.

We've set up a ccid, DUMP, which is used to handle most jobdump processing. It has a sigfile which only allows it to print dumps in batch mode, and which uses *CKID in interactive mode; so we no longer have to worry about security with the password (the password is DUMP). DUMP:DUMPER also keeps a log in files DUMP:LOG and

DUMP:TAPES. Hopefully this log will sometime replace the dump-log notebook the operators now keep; but not yet.

5) OPERATOR FILE *DMP

This *-file is used by the operators to log all dumps and generate standard format listings of all MTS job-dumps.

A *BATCH* job is created to print out the dump at deferred priority (MTS:PRINTDUMP takes a long, long time). For MTS jobs a "full" dump is taken. For other types of jobs, a more abreviated dump is taken; enough, it is hoped, to give you a start at figuring out what went wrong. File DUMP:DUMP.CMDS contains the actual MTS:PRINTDUMP commands that are used for each job-type.

The receipt number of the *BATCH* job created by *DMP should always be recorded in the dump-log notebook.

A log of all dumps is kept in file DUMP:LOG. In addition, file DUMP:TAPES has an up-to-date list of the current un-released tapes.

6) DUMP:DUMPER PROGRAM

This program is invoked by *DMP, but can also be used interactively to print dumps "while you wait".

a) To produce, and log, a dump-listing via a *BATCH* job:

```
$Run dump:dumper par=batch
```

...it will prompt for the tape number and the reason for taking the dump.

...it will log the dump in file DUMP:LOG and in file DUMP:TAPES.

...it will print the *BATCH* job receipt number

b) To log a dump, then generate the listing "on-line":

```
$Run dump:dumper
```

...the tape number and reason are prompted for and logged as with "par=batch"

...a log entry is made in file DUMP:LOG (but not DUMP:TAPES)

...the requested dump tape is mounted

...MTS:PRINTDUMP is started up via $RUN (which unloads DUMP:DUMPER of course) and the user is left in PRINTDUMP command mode.

The dump will be listed on *PRINT* by default unless SPRINT is explicitly assigned to something else on the $RUN DUMP:DUMPER command.

HASP EXECUTION AND PRINT PRIORITIES

The priority of a job is determined on the basis of the "size" of the job. The priority for execution is computed based on the limits for time given on the $SIGNON card. Small time requirements produce higher priorities. When execution of a job has completed, a new priority for printing is computed, based on how many pages the job actually produced (not how many were estimated) times the number of copies required.

<u>LOW, DEFERRED and MINIMUM Priorities</u>

These priorities may be specified by the user when the cost of the job is more important than the quickness with which the results can be retrieved. The user may specify PRIO=LOW OR L for LOW priority or PRIO=D or DEFER for DEFERRED priority or PRIO=MINIMUM. ')

Since the schedule for various priority classes is changed frequently, it is not listed here. If this schedule needs to be seen, <u>it can be found at the beginning of *RATES</u>.

These priorities apply to batch and terminal jobs.

<u>Notes</u>:

The commands $HOLD DEFER, $HOLD MINIMUM, $RELEASE LOW, $RELEASE DEFER and $RELEASE MINIMUM are done automatically by the system. These commands could be initiated manually by the operator if required by some unusual situation.

The status of LOW and DEFER are printed whenever a $status command is issued.

$BACKLOG lists jobs in LOW, DEFERRED or MINIMUM priority separately.

$QUE or $DISPLAY shows EXEC WL for jobs waiting in LOW priority and EXEC WD for jobs in the DEFERRED priority category.

Jobs in LOW priority are less expensive to run than those with a normal priority and jobs in DEFERRED priority are less expensive to run than LOW priority jobs. MINIMUM priority jobs are least expensive.

LOW, DEFERRED and MINIMUM jobs are not re-computed after execution. When the commands $HOLD LOW and/or $HOLD DEFER are given by HASP, priority 0 jobs in the active or waiting stage of print or punch are not affected.

Esssentailly, "normal" jobs aren't put ahead of "low" jobs etc.,

RM PRIORITIES-CLASSES-SCHEDULE

but rather the priority of the "low" job is lowered by a few points. And similarly for other priorities.

Every two hours, every print job that has been in the queue since the prvious two-hour period has its priority raised by one up to a limit of 15 for "normal", 13 for "low", 12 for "deferred" and 11 for "minimum". So, During the night, the priorities will drift up, but the jobs will remain in the same order since jobs are printed in order of arrival within a given priority class.

NOTE: Jobs that are ≥ 1500 pages are automatically held and are usually released by operators or DPAs during the second half of the midnight shift or when the print queue is very low.

Print & Execution Priorities, Categories & Classes

Execution Time (Seconds)	Priority	Print Categories	Priority Class
≤1	10	X = page	0 = unused
≤2	9	L = line	1 = minimum
≤4	8	P = PN	2 = deferred
≤8	7	T = TN	3 = low
≤16	6	S = special	4 = normal
≤32	5	M = mixed	5 = unused
≤64	4		6 = high
≤128	3		7 = flush
≤256	2		
>256	1		

Print/Page Limit by rate group:

Priority	Normal	Low	Deferred	Minimum
15	5	-	-	-
14	10	-	-	-
13	20	5	-	-
12	50	10	5	-
11	100	20	10	5
10	200	50	20	10
9	350	100	50	20
8	500	200	100	50
7	650	350	200	100
6	800	500	350	200
5	1000	650	500	350
4	1500	800	650	500
3	2000	1000	800	650
2	4000	1500	1000	800
1	6000	2000	1500	1000
0	>6000	>2000	>1500	>1000

HASP--HUNG **Obsolete**

NOTE: This section (HASP--HUNG) is mainly here for firefighters and systems programmers but may be of some use to operators.

What to do when Hasp won't accept any $COMMANDS.

If HASP does not respond to any operator commands, and HASP:HASPSTAT (as run by the operator file *HPS) does not report any errors (such as no $WTO buffers), it's probably because HASP has queued the maximum number of reply messages for a particular remote station. If *HPS reports that there are no WTO buffers, then HASP could be stuck on a remote console read -- see the section "Remote console read stuck" for how to recover from that situation.

Remote console write stuck

To avoid running out of buffers, HASP imposes a limit of 10 messages that can be queued for any one remote. If this limit is exceeded (for example, if the remote is experiencing hardware or software problems), then the WTO (write-to-operator) routine in HASP will wait until a buffer is freed. This wait causes the console processor to be suspended, thus blocking any new commands from being accepted.

The dump/analyzer program HASP:D/A can be used to detect this problem and to determine which remote is causing the trouble. Issue the following commands:

```
$Run Hasp:d/a
D PCE TYPE=1052 ALL
```

This displays all the console processor control elements (PCEs). You should see the contents of a message buffer displayed for one of the PCEs. If the preceding field PCEEWF is flagged 1052 instead of WORK, then the PCE is waiting to send a message and it's likely that the console processor is hung. The text of the message in the buffer might include the offending remote number. If not, you can determine the remote number from Hasp's globals. Issue the command

```
D GLOBALS
```

Check the field WCONBYTE. The low order 7 bits is the hex value for the remote console number that Hasp is trying to send a message. Display the remote table for this RJE via

```
D RTAB ##
```

where "##" is the remote number in decimal. If the field RTABMCNT is 0, then the maximum number of messages are queued for this remote.

RTABMCNT starts at 10 and is decremented for each queued message.

To clear up the WTO wait, it is necessary to $STOP the remote SAnn line. However, since HASP won't accept the $STOP SAnn command, you have to do it by hand. First of all you need to know which SAnn line the remote is on and whether it is a leased line or dial-up. Assuming the remote number is "##" (1 or 2 digits), type

```
D DCT RM##.RD1
```

One of the fields displayed for the device control table (DCT) for the first reader (RD1) for this remote is "DCTDCB=addr", where "addr" is the vm address of the line manager DCT for the remote. Display this DCT as follows:

```
D DCT AT=addr
```

Looking at this DCT you can find the SAnn device name in the field DCTDEVN and whether the line is leased - signified by the flag "LEASE" in the field DCTPSTAT. If "LEASE" does not appear, then the remote is a dial-up and you can get rid of it by hanging up the phone line for SAnn (given in DCTDEVN). Things should clear up shortly.

If the remote is on a leased line, then you have to stop the HASPLING doing the I/O on the SA device and set the stop bit in the DCT for the line. Find the UMMPS task number via

```
$SYS T D SAnn
```

(or look at the field DCTSJNO) and stop it via

```
$SYS O STOP task#
```

Finally, turn on the high-order bit (DCTSTOP = X'80') of the flag byte DCTFLAGS (offset X'1C' in the DCT) in the line DCT. For example,

```
$SYS MOD addr+1C 80
```

where "addr" was determined above and assuming DCTFLAGS was 00. Once the console queue clears up and the SAnn line has successfully stopped, it will be necessary to restart the line via $START SAnn.

The following is an editted HASP:D/A session demonstrating the above steps for remote 22, BSSR. In actuality, the last few times this situation has occured, remote 2 FLNT has been the culprit.

```
#run hasp:d/a
#12:17:53
why
I don't know

[same as running HASP:HASPSTAT]

d pce type=1052 all

RM PRIORITIES-CLASSES-SCHEDULE
```

```
PCE AT 228728 1052

PCESAVEA =01000000,00000000 REGISTERS(13->12):
00228728 601C7938 901C7AEE 02014114 0021A504 00000000 001A919C 00000000
00000000 00000003 00232858 00000000 00000000 00000000 001A9018 001C770C
PCEEPSAV =00000000 PCENEXT =0022877C PCEEWF =WORK PCEID =1052
 ... [ a couple of PCEs not shown ]

PCE AT 230E04 1052
PCESAVEA =01000000,4021E634 REGISTERS(13->12):
00230E04 5021ACF8 5021AC38 00000000 00000000 00230FDC 001A91B4 00000001
00000005 A021C3BA 8021C35A 0021CA9C 00000000 001B62D8 001A9018 0021ABF0
PCEEPSAV =00000000 PCENEXT =00230F20 PCEEWF =1052 PCEID =1052
230E58(000054+): 22005BD3 D6C7D5C5 E6D3D6C7 C6C9D3C5 *--$LOGNEWLOGFILE*
230E68(000064+): 40404040 40404040 40404040 40404040 * *
 ...
230ED8(0000D4+): 84858440 40404040 4040F5F1 F2F440E2 *ded 5124 S*
230EE8(0000E4+): D740C3D5 E3D94040 F1F17AF3 F640F2F3 *P CNTR 11:36 23*
230EF8(0000F4+): 40404040 00000000 00000000 00000000 * ------------*
 ...
230F18(000114+): 00000000 00000789 *-------i*

[ note PCEEWF=1052 above, remote number not in message ]

d globals
Globals at 1A9018

$JOBQPTR =001AD898 $JQFREE =001B0408 $JQENT =001AF238 $JQPV:
001B0A78 001AF418 001B0C28 00000000 00000000 00000000 001B0828 001AEEB8
001ADEB8 00000000 001B0898 001ADBB8 001AEC88 001AFFB8 001ADA08 001AF238
$S8NBR =624093.  $XEQTOTL =7671.  $DACKPT =00 $NUMPRT =57.
$NUMRJE =43.  $POOLECB =FF000000 $POOLECF =FF HASPXWT =00
$POOLCHG =00 $POOLJBF =FF $NUMDA =2.  $CKPTRAK =10000003
$CYLALOC =287.  VCYLTOTL =2044.  $PRCHKPT =00226EFC
$QUEFREE =0021A504 $BUFPOOL =001EC4C8 $TPBPOOL =001CC3E8 $LOWFLAG =00
MINXCLAS =Normal INITDONE =FF QCKPTEWF =00 $HOLDPDS =00
$HOLDEX =00 $DRAIN =00 $WAYT =00 $ERRDUMP =FF
$HASPERR =00000000 $ERRSAVE:
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
$IDLELMT =3600.  LASTTIME =973C7F3D CURNTCLK =973C7F9E,7A56D400
WCONBYTE =96 WQUEBUSY =00000000
WSPOOLQ =00000000,00000000,00000000,00000000,00000000

calc '96'-'80'                   [ from WCONBYTE above ]
=22                              [ compute remote number in decimal ]

d rtab 22                        [ display remote name ]
RTAB at 1AC944
RTABID ='BSSR' RTABRNUM =22.  RTABFLGS =INUSE,PERM,SGON,UC
RTABPAGE =38.  RTABTNPG =0.  RTABCARD =0.  RTABPRRT =BSSR
RTABPURT =BSSR RTABLIST =00 RTABMCNT =0.  RTABPSWD =' '
RTABSPPG =0.  RTABACPT =0
```

```
d dct rm22.rd1                   [ find the line DCT in DCTDCB ]
DCT at 232BDC RM22.RD1
DCTSTAT =HOLD DCTPCE =000000 DCTBUFAD =00000000 DCTPSTAT =SINON
DCTDCB =2318F8 DCTBUFCT =0.  DCTNO =22.
DCTDEVTP =REMOTE_READER DCTPCODE =CON/MRF,CPU,TRSP DCTPRINT =16
DCTPUNCH =16 DCTPRINC =0.  DCTPRLIM =0F DCTFORMS =00
DCTCHARS =Uc,Pn DCTPRLEN =132.  DCTPLINE =P20,TRSP,BSC
DCTEWF =00000000 DCTFLAGS =0 DCTCHAIN =232C18
DCTDEVN ='RM22.RD1' DCTWAYT =00000000 DCTSJNO =0.  DCTLOCK =00
DCTPRNTR =Line DCTTOTPG =0.  DCTTOTJB =0.  DCTRESCN =00
MDCTRCB =93 MDCTDCT =232C18

d dct at=2318f8                 [ address shown for DCTDCB ]
DCT at 2318F8 SA07
DCTSTAT =INUSE DCTPCE =2286D4 DCTBUFAD =001C8C38
DCTPSTAT =LEASE,ETX,POST DCTDCB =000000 DCTBUFCT =0.
MDCTATTN =TIMER DCTDEVTP =LINE DCTPCODE =FULL/BLK,TRSP,BSC
MDCTCODE =001AC06C MDCTRSEQ =06 MDCTTSEQ =6C MDCTERCT =08
DCTPLINE =TRSP,BSC MDCTOPCT =00 MDCTOBUF =000000 DCTFLAGS =0
DCTCHAIN =231940 DCTDEVN ='SA07 ' DCTWAYT =FF000000
DCTSJNO =92.  DCTLOCK =00 DCTPRNTR =Line DCTTOTPG =0.
DCTTOTJB =0.  DCTRESCN =00 MDCTRCB =00 MDCTDCT =232BDC
MDCTFCS =8FCF0000 MDCTPSWD =' ' MDCTTIME =973C7E9E
 [offending remote on SA07, DCTPSTAT shows it's a leased line ]

$sy t d sa07                     [ find out task number ]
-00092 HASPLING 04C860 Hsp FF231920 SA07; SA07

$sy o stop 92                    [ stop the haspling ]

$sy mod 2318f8+1c 80             [ set DCTFLAGS to X'80' ]

$sy t d sa07
-*** Job not found
```

Remote Console Read Stuck   **Obsolete**

HASP uses a single PCE and DCT for all console messages to and from all remotes. If the remote console PCE gets stuck in a read waiting for a command from a remote, then HASP will quickly run out of WTO buffers as all remote messages will queue up waiting for the read to complete. This situation is not supposed to happen, but it has on several occasions recently for remote BSSR.

The dump/analyzer program D/A can be used to detect and recover from this situation as for HASP stuck on a remote console write. The following is an annotated HASP:D/A session showing how to detect a stuck remote console read and a way to recover without reloading.

```
#r hasp:d/a
#14:12:13
why
No free WTO buffers.
d globals
Globals at 1C7018

$JOBQPTR =001CB8B0 $JQFREE =001D0510 $JQENT =001CD250 $JQPV:
001CD640 001CFEF0 00000000 00000000 00000000 00000000 001D0050 001CCED0
001CF2E0 001CD8D0 001CED20 001CCC20 001CBCB0 001CC180 001D04D0 001CD250
$S8NBR =650111.  $XEQTOTL =21237.  $DACKPT =00 $NUMPRT =57.
$NUMRJE =43.  $POOLECB =FF000000 $POOLECF =FF HASPXWT =00
$POOLCHG =00 $POOLJBF =FF $NUMDA =2.  $CKPTRAK =10000003
$CYLALOC =269.  VCYLTOTL =2044.  $PRCHKPT =00245114
$QUEFREE =00000000 $BUFPOOL =00225860 $TPBPOOL =001ECD40 $LOWFLAG =00
MINXCLAS =Normal INITDONE =FF QCKPTEWF =00 $HOLDPDS =00
$HOLDEX =00 $DRAIN =00 $WAYT =00 $ERRDUMP =FF
$HASPERR =00000000 $ERRSAVE:
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
$IDLELMT =3600.  LASTTIME =98618B09 CURNTCLK =98618BD6,2C6FEE00
WCONBYTE =96 WQUEBUSY =00237DD8
WSPOOLQ =00BB0068,00C52B30,00000000,00000000,00000000
```

Since $QUEFREE=0 there are no free WTO buffers. Display the remote console DCT to see what's happening.

```
d dct type=remote_console

DCT at 1E5B78 RMT-CON
DCTSTAT =INUSE DCTPCE =246948 DCTBUFAD =001ECD40
DCTPSTAT =SINON DCTDCB =24FB18 DCTBUFCT =0.  DCTNO =0.
DCTDEVTP =REMOTE_CONSOLE DCTPCODE =CON/MRF,CPU,TRSP DCTPRINT =00
DCTPUNCH =00 DCTPRINC =0.  DCTPRLIM =0F DCTFORMS =23
DCTCHARS =Mx,Uc,Pn+00000040 DCTPRLEN =126.
DCTPLINE =HARD,P20,TRSP,BSC DCTEWF =00000000 DCTFLAGS =0
DCTCHAIN =000000 DCTDEVN ='RMT-CON ' DCTWAYT =D9D4F2F9
DCTSJNO =19200.  DCTLOCK =00 DCTPRNTR =0 DCTTOTPG =0.
DCTTOTJB =0.  DCTRESCN =00 MDCTRCB =92 MDCTDCT =000000
```

The important field here is MDCTRCB. The record control byte has the value 92 which means the remote console is doing a read. If it was 91 then a write to a remote console would be in effect. The next thing is to see what the associated PCE is doing. The remote console PCE is available via the external symbol $MCONPCE. It is also saved in the field DCTPCE above.

```
d pce at=$mconpce

PCE AT 246948 1052
PCESAVEA =01000000,00000000 REGISTERS(13->12):
00246948 601E5950 901C9144 02016214 002379F8 00000000 00238788 00000000
00000000 00000003 00000000 00000000 00000000 00000000 001C7018 001E5724
PCEEPSAV =00000000 PCENEXT =0024699C PCEEWF =WORK PCEID =1052
```

The field PCEEWF shows that the PCE is waiting for WORK. This shouldn't normally be the case. The register 15 value above indicates the restart address for the PCE and can be used to check the HASP listing to see where the wait really is.

```
xcalc 1c9144-hasp
=00002144
```

This turns out to be near the beginning of the routine MBCPUGET which gets the next logical record (in this case the console command) from a CPU type remote. The PCE is stopped at the instruction "$WAIT WORK", waiting for the command to appear in its buffer. However, this is never supposed to happen (i.e., the record should already be there) since it suspends the one and only remote console PCE. Check the buffer (from the field DCTBUFAD above):

```
$res prot=off                    [ it's protected so do this first ]
dis buffer at=1ecd40 long
TPbuffer at 1ECD40
IOBFLAG1 =42 IOBFLAG2 =00 IOBSENS0 =00 IOBSENS1 =00 IOBECBCC =00
IOBECBPT =000000 IOBFLAG3 =00 IOBCSW =00000000,000000 IOBSIOCC =00
IOBSTART =24C974 IOBDCBPT =00000000 IOBREPM =00 IOBRESTR =24C974
TPBMXREC =00 TPMCONCH =000000 TPBLCCC =11 TPBLCCAD =1ECEEA
TPBRECNT =C5 TPBFDATA =1ECDC5 BUFECBCC =00 BUFACHN =000000
BUFTYPE =80 BUFDCTE =24FB18 BUFEWF =00000000 LCBMCB =00
LCBACK =00 LCBRCB =0000 BUFPCEAD =0024C908
IOBCCW1 =27,000000,CC+SLI,1.  IOBCCW2 =01,1CA06C,CD+SLI,4.
IOBCCW3 =01,1ECD76,CC+SLI,2.  IOBCCW4 =02,1ECDC0,SLI,400.
IOBCCW5 =01,1CA06C,CD+SLI,4.  IOBCCW6 =01,1ECDC0,CC+SLI,361.
IOBCCW7 =01,1CA07E,CC+SLI,2.  IOBCCW8 =02,1ECDC0,SLI,400.
TPBUFCHN =1002888F
1ECDC0: 1002888F CFF0C179 5BE2C9C7 D5D6D540 *--- -0A$SIGNON *
1ECDD0: 40404040 404040D9 C5D4D6E3 C5F2F240 * REMOTE22 *
1ECDE0: C2E2E2D9 40404040 40404040 40404040 *BSSR *
 ...
1ECF40: 40404040 40404040 40404040 40404040 * *
```

The field TPBFDATA is the pointer to the next logical record, which in this case points to the string "F0C1795B...". However, it should be pointing to a record control byte with value 92 (the same as MDCTRCB in the DCT), since HASP saw that before it initiated the console read.

```
dis 1ecdc5
1ECDC5: F0C1795B
```

One way to recover from this is to alter the data in the buffer (pointed to by TPBFDATA) to be something HASP expects. However, before doing this, it's best to $STOP the remote line as outlined in the previous section. The line DCT for the remote can be found from the field DCTDCB in the remote-console PCE or BUFDCTE in the buffer. For example:

```
d dct at=24fb18

DCT at 24FB18 SA07
DCTSTAT =INUSE DCTPCE =2468F4 DCTBUFAD =001ECDC0
DCTPSTAT =LEASE,ETX DCTDCB =000000 DCTBUFCT =1.
MDCTATTN =0 DCTDEVTP =LINE DCTPCODE =FULL/BLK,TRSP,BSC
MDCTCODE =001CA06C MDCTRSEQ =01 MDCTTSEQ =00 MDCTERCT =08
DCTPLINE =TRSP,BSC MDCTOPCT =00 MDCTOBUF =000000 DCTFLAGS =0
DCTCHAIN =24FB60 DCTDEVN ='SA07 ' DCTWAYT =00000000
DCTSJNO =0.  DCTLOCK =00 DCTPRNTR =Line DCTTOTPG =0.
DCTTOTJB =0.  DCTRESCN =00 MDCTRCB =00 MDCTDCT =250EEC
MDCTFCS =8FCF0000 MDCTPSWD =' ' MDCTTIME =985F4427
```

The offending remote is on SA07 (BSSR again), so stop its HASPLING and set the 'STOP' bit in the line DCT.

```
$sys t d sa07
-00027 HASPLING 04E950 Hsp FF24FB40 SA07; SA07
$sys o stop 27
$sys t d sa07
-*** Job not found
$sys mod 24fb18+1c 80            [ turn on stop bit in DCT for SA07 ]
```

Finally, change the data in the remote console buffer to give an end-of-file to the MBCPUGET routine - the magic sequence is 92800000:

```
$sys mod 1ecdc5 92800000         [ address form TPBFDATA ]
dis 1ecdc5
1ECDC5: 92800000
```

Fire up the remote console PCE by clearing the PCEEWF field (offset X'50' in the PCE):

```
$sys mod 246948+50 0000
```

The PCE will get an EOF, notice that the line DCT was stopped, get rid of it and go onto the next remote console write. The WTO queue should clear up a few minutes later.

Procedure for Freeing UM HASP Transmitters

If HASP seems to be stuck attempting to transmit UB jobs to the U
The following procedure may clear the impaired UM receivers.

Note: If you follow the procedures below,

1. $run hasp:d/a

2. Enter: d dct type=mnetreader all

The system should display something like the following:

```
0DCT at 2BEA64 MNT.RD1
 DCTSTAT =INUSE DCTPCE =2B7F98 DCTBUFAD =0021EE00 DCTPSTAT =0
 DCTDCB =000000 DCTBUFCT =1.  DCTNO =43.  DCTDEVTP =MNETREADER
 DCTIOTYP =WRITE DCTFLAG2 =0 DCTPRINC =0.  DCTPRLIM =0F
 DCTFORMS =00 DCTCHARS =Uc,Pn DCTPRLEN =80.  DCTPLINE =0
 DCTEWF =002B7FE9 DCTFLAGS =0 DCTCHAIN =2BEAA0
 DCTDEVN ='MNT.RD1 ' DCTWAYT =0021EE00 DCTSJNO =5969.
 DCTLOCK =00 DCTPRNTR =Line DCTTOTPG =0.  DCTTOTJB =0.
 DCTRESCN =00
WORK SPACE:
 (000034+): 00000000 00000000 *--------*
```

3. For each item from step 2, look at DCTSJNO (underlined above) to get the task number.

4. Issue the following MTS systemstatus command:

```
$sy j tasknumber
```

where "tasknumber" is the task number from step 3. The output will appear as:

```
-) Job Name JobTbl Status BTS2 User Proj Par
) PSW Prob Time Super Time VP RP RPN Page-ins File I/O
0)05969 HASPLING 000AD320 I/O on AF7E 20 MNAF ????
072D0000 002C1E44 1.299 5.068 1 1 1 2 0
R0-7: 00000001 006001A0 002C2164 00000000 0021EE00 00600000
0021EE80 802C1E08
R8-F: 005F7A80 005F7B00 802C1A86 006000DC 002C19B0 0060004C
802C1E08 402C1B96
FPRs: 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000
Dev: AF7E 70

005969 HASPLING 000AD320 I/O on AF7E 20 MNAF ????
072D0000 002C1E44 1.304 5.083 1 1 1 2 0
R0-7: 00000001 006001A0 002C2164 00000000 0021EE00 00600000
0021EE80 802C1E08
R8-F: 005F7A80 005F7B00 802C1A86 006000DC 002C19B0 0060004C
802C1E08 402C1B96
FPRs: 00000000 00000000 00000000 00000000 00000000 00000000
```

```
00000000 00000000
Dev: AF7E 70
```

5. If the task is dormant (this will show after "HASPLING" rather than "I/O" on A...) or it is using CPU time, as shown here, it's ok. If it is not dormant and not using CPU time, then issue the MTS command:

```
$run sys.:units
```

To get the status, enter:

```
d devicename s
```

where "devicename" would be AF7E in the above example. The output will look like:

```
AF7E Type: MNET DevNum: 0CFE (ON ) LPM: 80 Sch: 0324 Job: 05969

    device initialized, no errors, I/O queue active, task waiting
    I/O operation awaiting channel end

    SCHIB:  mode flags: sch=ON, lim chk=OFF, measur=(blk upd,dev
,
                        timing, devnum ok, int sbcls=3

             MBI=0324; LPM=80 PNOM=00 LPUM=80 PIM=80 POM=FF PAM=80
                      PATH=0C
    SCSW OF SCHIB:  key=0
           Fmt 1 CCW, Prefetch allowed
           Function: start,  Activity: sch actv dev actv,
           Dev status:  unit check
           Sch status:  PCI, program check
    DEVLST:  Config flags:  No timeout
    DEVTBL:  ATTN=ON, SNS=ON
    SCSW OF DEVTBL:  key=0
```

The status should be "i/o operation awaiting channel end" for this solution to work. If the status is not stated as above, then the problem is elsewhere and the action stated here will not fix it.

To be sure that you have the right problem, issue the following MTS command:

```
$run mta.:disschib*oa
```

Enter the device name at the prompt and the output will look like:

```
Path control word: 0000B6D8 189B0CFE 80008080 0324FF80 0CFFFFFF 0
Subchannel status word: 00C040C0 00B26AE0 02A00080
```

If, unlike this example, the second half of the first word of the sub-channel status word, 40C0 in this case, is hex '0', procede with the following:

Note: This step will not be necessary once UNITS is fixed to print the information out correctly. Also, "clr!" can fix a number of problems, so it may work even if the problem is not exactly the one described here.)

8. Issue the MTS command:

```
          $run sys.:units
```

Then, enter:

```
          d devicename clr!
```

This should free the stuck HASP receiver. If the receivers appear to be free but thing are still stuck, call the Firefighters since the problem may be with the UB transmitters rather than with the receivers.

A simpler version of the above may be tried. See below.

```
     /t d ab7e

/t d ad7e

     /t d ae7e

/t d af7e

     /t d an7e
```

The above tasking will get you the "task" numbers of the devices of concern. Check the hex number under "super time" from the "/J tasknumber" (see above) and if there's no change, then do a "units d a-7e clr!".

One can also use "$status all" on the UB screen, wait a couple of minutes, do it again and see if the jobs are changing.

Host Interface Machines (HIMs)

HIMs 1, 2 and 3 are located in the Electronics shop located at the east end of the second floor (CNTR).

The HIMs are one means for users to connect to MTS using network connections. At times, HIMs experince interface-control checks (IFCCs). If these occur with sufficient rapidity, MTS will fail and will have to be reloaded.

Under normal circumstances, one or two HIM lines, usually HIM1, will be high-lighted at the top of the console screen.
However, when a him is in trouble (usually IFCCS) there will be a long line of individual HIM lines highlighted.
Sometimes, error comments regarding IFCCs appear on the console with enough frequency to interfere with normal operation.

There are several ways to handle HIM problems.

If the screen is filling up with error comments -

```
$signon w035     (at a terminal)

$sou strt:offlinehimx (offlines all HIMx connections)

mts *him xstp     (stops HIMx)
```

Go to IBM System Console and, choose the Channel Configuration screen

```
chpid xx off        (offlines chpid (channel) to which
                    HIM is attached)

chpid xx on     (onlines chpid)
```

Go into shop and reload the HIM using the INIT switch

```
$sou strt:onlinehimx     (onlines HIMx connections)
```

If a line of HIMs is high-lighted at the top of the screen or there is some other reason that a HIM must be stopped and started -

```
mts *him xstp    (stops HIMx)

/t d hx##           (checks to see if there are
                    remaining connections)
```

The remaining connections can be STOPped or BLASTed.

```
mts *him x,,,     (starts HIMx)
```

or, another method -

```
          mts *him xstp
```

Go to IBM System Console and, choose Channel Configuration screen

```
          chpid xx off        (offlines chpid (channel) to which
                              HIM is attached)

                              chpid xx on     (onlines chpid)

                              mts *him x,,,
```

There is usually a note near the System Console listing HIM addresses. At present, these adresses are

```
               HIM1 = 17
               HIM2 = 1B
               HIM3 = 2C
```

If someone calls, for instance from the NOC, and mentions that a HIM seems to have failed, you can dial in to a HIM using Versaterm. At the "Which Host?" prompt, type -

```
          141.211.144.21  HIM1
      or
          141.211.144.22  HIM2
      or
          141.211.144.23  HIM3
```

depending on which HIM you want.

If the connection fails, you will have to stop and start the HIM as mentioned above.

Hours - Computing Center

A statement of regular Computing Center hours can be seen by $COPYing NEWS:CC.HOURS.

SHUTDOWNS--(Scheduled)

Normal scheduled shutdowns for preventive Maintenance and/or software testing, etc., are listed below.

6:00 a.m. - 8:00 a.m., Tues. and Fri.

Emregency shutdowns, depending upon their urgency are "right now", or 6:00 p.m. In some cases, a noon shutdown may be scheduled.

Weekends from 11:30 p.m., Sat. to 8 a.m., Sun.
and from 11:30 p.m., Sun. to 8 a.m., Mon.

For regular Computing Center hours, $copy NEWS:CC.HOURS.

When Computing Center hours are expected to differ from normal, the user public should be made aware of this well in advance. Sometimes, unusual hours may be anticipated far enough ahead of time to publish them in our newsletter. In any case, the Center staff is usually aware of a change in hours enough ahead of time to make signs that can be posted on the doors of the Computing Center. A signonmessage which has as part of its contents, the special scheduled hours or the same type of message pointing to a file with the hours should be posted.

HOLIDAY SIGN PROGRAM

This program draws door signs for announcing holiday or other unusual hours.

Currently, the implemented formats are (1) title line (centered), with 4 lines below for hours, and (2) title line (centered), with 3 lines below for hours.

Examples:

```
     -------------------------------------------------
  (1)|            Thanksgiving Hours                  |
     |                                                |
     |     Thurs. Nov. 25 - - Closed                  |
     |     Fri.   Nov. 26 - - Closed                  |
     |     Sat.   Nov. 27 - - Open at 9 a.m.          |
     |     Sun.   Nov. 28 - - Normal hours            |
     |                                                |
     -------------------------------------------------


     -------------------------------------------------
  (2)|            Thanksgiving Hours                  |
     |                                                |
     |     Thurs. Nov. 25 - - Closed                  |
     |     Fri.   Nov. 26 - - Closed                  |
     |     Sat.   Nov. 27 - - Normal Hours            |
     |                                                |
     -------------------------------------------------
```

The box around the sign is drawn by the program.

To run the program, enter

```
$R W035:HOLIDAY(1000)    (to use PLT1)

$R W035:HOLIDAY(2000)    (to use PLT2)

$R W035:HOLIDAY(3000)    (to use PLT3)
```

The first statement that will appear on the terminal is a request for an alphabet in the form-

COMPUTING CENTER HOURS

ALPHABET:

If RETURN is pressed with no alphabet given at this point, the default alphabet, SANSERIF.1 will be used. Other alphabets can be found in MTS Volume 11, appendix A.

The program will request a sign format--3 line, 4 line, repeat, or other (other isn't implemented, yet). A one character response (3, 4, or R) will suffice. An end-of-file terminates the program. Next, the program types "SET UP THE PLOTTER" (the plotter is assummed to be available). Respond with a carriage return when the plotter is ready for use. The position at which you leave the pen becomes the lower left hand corner of the sign (unlike the postprocessor).

If the format is repeat, the previous sign will be redrawn (if there is a previous sign).

Then the program prints "ENTER TITLE LINE". Enter the title, exactly as it is to appear on the sign (to get lower case on a terminal through the data concentrator, use CONT.-A CONT.-A UC=off). The title will be centered. Next, the program requests the format of the "hours" lines--centered (center each line), or just draw a given line, linr the other lines up with it), or just draw (specify an x-coordinate for each line). The first character of each type (C, L, J,) will suffice for a response.

For centered lines, the program will request the text for each line (in order from the top of the sign). Enter each line exactly as you want it to appear.

For lineup format, the program first requests the text of the line which is to be centered. It then asks which line this is (1, 2, 3, or 4, numbered from the top). Then the text of the other lines is requested, just as with centered.

For "just draw", the x-coordinate for each line to be drawn is requested (this is the distance, in inches entered as a floating point number-- always include the decimal point, from the left side of the sign).

Examples of each format are on the following pages.

Let the systems programmer in charge of the plot routines (currently, Gail Lift) know what you want implemented under other, or if you need more characters in the alphabet (e.g., if it doesn't currently have a semicolon), or if you don't like the way it works, or . . .

The following example produces 4 signs. They are shown in order (at 1/4 size) on the next page.

```
#R W035:HOLIDAY(1000)
#EXECUTION BEGINS
WHICH FORMAT (4 LINE, 3 LINE, REPEAT, OTHER)?
4
SET UP THE PLOTTER

ENTER TITLE LINE: uc=off
*OK
A Title CENTER, LINE UP, OR JUST DRAW LINES?
c
TEXT FOR LINE 1
first line
TEXT FOR LINE 2
second line
TEXT FOR LINE 3
third line
TEXT FOR LINE 4
fourth line
WHICH FORMAT (4 LINE, 3 LINE, REPEAT, OTHER)?
4
SET UP THE PLOTTER.

ENTER TITLE LINE:
Title
CENTER, LINEUP, OR JUST DRAW LINES?
1
LINE TO BE CENTERED:
third line (centered)
WHICH LINE IS THAT?
3,
TEXT FOR LINE 1
first line
TEXT FOR LINE 2
second line
TEXT FOR LINE 4
fourth line
WHICH FORMAT (4 LINE, 3 LINE, REPEAT, OTHER)?

SET UP THE PLOTTER.

WHICH FORMAT (4 LINE, 3 LINE, REPEAT, OTHER,)?
3
SET UP THE PLOTTER.
```

```
ENTER TITLE LINE:
title line
CENTER, LINEUP, OR JUST DRAW LINES?
j
X FOR LINE 1
2.,
TEXT FOR LINE:

First line, at 2.
X FOR LINE 2
3.,
TEXT FOR LINE:
second line, at 3.
X FOR LINE 3
4.,
TEXT FOR LINE:
third line, at 4.
WHICH FORMAT (4 LINE, 3 LINE, REPEAT, OTHER)?
#EXECUTION TERMINATED
```

NOTE: On the next page are examples of the signs which were generated by the program.

(Continued on next page)

```
 -------------------------------------------------
|                  A  Title                       |
|                  first line                     |
|                  second line                    |
|                  third line                     |
|                  fourth line                    |
 -------------------------------------------------


 -------------------------------------------------
|                    Title                        |
|              first line                         |
|              second line                        |
|              third line (centered)              |
|              fourth line                        |
 -------------------------------------------------


 -------------------------------------------------
|                   Title                         |
|             first line                          |
|             second line                         |
|             third line (centered)               |
|             fourth line                         |
 -------------------------------------------------


 -------------------------------------------------
|                Title Line                       |
|   first line, at 2.                             |
|      second line, at 3.                         |
|         third line, at 4.                       |
 -------------------------------------------------
```

How to Install a New IOCDS
Operations Version

K. Bosley/C. Lever

IOCDS: Input/Output Configuration Data Set

To install a new IOCDS, perform the following steps:

1) Shut down UM-MTS (or VM and its guests) and press the STOP key on the ES3090/300J console.

2) Type "F IOCDSM" and press ENTER to select the IOCDSM (IOCDS management) frame on the 3090.

3) The ES3090/300J saves 6 IOCDSs for each side of the machine: A0-A5 on the A (UM) side; B0-B5 on the B (UB) side (only 4 happen to be defined for the UB side currently). Observe that each is tagged with a date of the form yy.ddd, where "yy" is the year and "ddd" is the day, from 1-366. For example, January 1, 1988 is 88.001. Select the oldest IOCDS for the particular side (A or B) with which you are working.

4) Release the write-protect status of the IOCDS you've selected by typing "A3 Ix" where "x" is the number of the line on the menu designating the chosen IOCDS (the number should range between 1 and 12).

5) Type "F ACCESS" and press ENTER to select the ACCESS frame.

6) Type "A1" and press ENTER to select the access level.

7) Type "IOCP CPx", where "x" is the number of a CPU. On the UM side, "x" should be 1; on the UB side, "x" should be 4.

8) The ES3090/300J alarm will ring, and the message "No Program Console Assigned" will be displayed on the console.

9) Press the "ASSGN CONS" key. The ES3090/300J will display the "Console Assignment" frame.

10) Select option "A3" (program mode) and press ENTER; then press the "END" key. The ES3090/300J will display the IOCP general menu.

11) Select option 5 ("Build IOCDS from Cards") and press ENTER. The screen will shift to the appropriate menu.

12) Ready the IOCDS tape (S2406) on any drive (we'll use >t904 here). Tape drive addresses begin at 0680 for the >T900 series drives, and at 1680 for the >T920 series, for example, >t910= address 068A. The tape will have been written with the new IOCDS you are about to install. Use only the RETURN and arrow keys to position the cursor in the proper fields; don't press ENTER until all of the above fields have been filled in.

Fill in the "Device Address" field with the number of the tape drive; for our example, this value would be 0684.

specify "Device Type" as "T";

enter "Y" in the "Ignore Unknown Card Statements" field;

enter "NA" in the "Output Printer" field;

specify "1" for the "File Number" field;

enter "N" in the "LPAR MODE" field.

13) Press ENTER. The tape should spin, and the ES3090/300J will read in the IOCDS. Eventually the message "PROCESSING COMPLETE. PROCEED" should appear. If an error occurs at this point, return to the normal system screen (via steps 14, and 17 through 20 below) and reload the system normally. Contact a systems group firefighter via $Message.

14) Type "MENU" on the command line at the bottom of the screen and press ENTER. The IOCP general menu will appear.

15) Select option 8 ("Write IOCDS to the Processor Controller File") and press ENTER. The screen will shift to the appropriate menu.

16) Fill in the "WRITE IOCDS" field with the name (A0-A5, B0-B5) of the IOCDS you selected in step 5 above. Press ENTER and the new IOCDS will be written to permanent storage. It will take a few minutes; wait for the "Processing Completed" message.

17) Type "END" on the command line at the bottom of the screen and press ENTER. The ES3090/300J console alarm will ring.

18) Press the "ASSGN CONS" key. The ES3090/300J will display the "Console Assignment" frame.

19) Select option "A3" ("Program Mode") and press ENTER to deselect the program mode console.

20) Press the SWAP CONS key one or more times until the legend at the bottom of the screen offers the system mode console. Then press END. The system console should appear.

21) Enter the command "CONFIG POR IOCDS(sx) WP(sx)" and press ENTER, where "sx" is the name of the IOCDS you just built (i.e., A0-A5 or B0-B5). The console alarm will ring, the screen may tell you that the POR option is unnecessary; simply press ENTER. This step will take about 5 minutes. When it finishes, the ES3090/300J will have been rebooted with the new IOCDS and the new IOCDS will be write-protected.

22) Reload UM-MTS, or VM and its guests, and proceed with normal operations.

LIMITED-SERVICE-STATE (LSS)

This Writeup describing Limited-Service-State can be seen by $COPYING the public file, *LSS. The Writeup contained in the file *LSS follows.

This writeup was last changed on 20 May, 1976 when the Computing Center policy on the use of Limited-Service State (LSS) was changed so that the institution of LSS is done without regard to the length of the batch execution queue. Formerly, LSS was only instituted when the batch queue exceeded 300 jobs. Concurrent with this change, 12 more dial-up terminal ports have been made available so that busy signals should be somewhat less frequent, but LSS more frequent. A full explanation of LSS follows.

As many experienced users know, it sometimes becomes necessary for the Computing Center to limit terminal usage of MTS during periods of very heavy usage or when hardware equipment failures cause the capacity of the system to be reduced. Rather than simply reducing the number of terminal users by making some phone lines unavailable, we have chosen to use a procedure called Limited- Service State (LSS) which will restrict certain terminal users (those "in LSS") to using a subset of the normal MTS commands, i.e. LSS users are unable to run large programs but they may still run programs requiring less than .25 seconds of CPU time, create, edit, and destroy files, mount tapes, use the MERIT computer network, etc.

A more complete description of the behavior of the system after LSS is activated follows. There are approximately 55 more lines in this file; you may prefer to $COPY it to *PRINT*.

The system will enable the use of LSS only if some hardware component is overloaded and the overload condition cannot be corrected by reducing to number of concurrent batch jobs being executed to 2. The system load will be examined whenever a new terminal user establishes his telephone connection to the system. If the system happens to be heavily loaded, the user may still sign on, but he will be temporarily prohibited from performing certain computing tasks that require large amounts of CPU time. That is, he will be put into Limited-Service State, and will be so informed immediately by a pre-signon message:

```
THIS IS A LIMITED SIGNON: EXECUTION TIME LIMITED TO .25 SEC  PER
RUN
```

So long as the user is in LSS, all commands dealing with the running of programs (except $RUN) are illegal, and attempts to use them will produce the message:

```
ILLEGAL COMMAND FOR LIMITED SERVICE STATE
```

LIMITED SERVICE STATE-LSS

The illegal commands under LSS are: $DEBUG, $LOAD, $START, and $RESTART. Except for these four commands, all other commands are legal under LSS. Attempting to use any of these four commands or the $RUN command will cause the user to be removed from LSS automatically if the system load has dropped sufficiently. In this case, the message

```
LSS MODE HAS BEEN TURNED OFF
```

will be printed (unless $SET TERSE=ON has been given) and the requested command will be executed normally.

Whenever the user is in LSS, the $RUN command will have a .25 second local time estimate imposed automatically unless a time estimate is explicitly specified on the command, in which case it may not be specified as greater than .25 seconds.

A user in LSS can explicitly request a transfer to normal full-service state is by issuing the command:

```
$SET LSS=OFF
```

If the system is able to grant his request, it will respond by printing the prefix character (#), asking for the next input line. (This is the only confirmation the user will receive.) But if the system load is still too high to grant the user's request, he will receive the message:

```
CAN'T SET LSS OFF - LOAD TOO HIGH
```

Over the past few years, the functions provided by several system programs which formerly were usable only via the $RUN command, have been incorporated into the MTS command language and have thus become available even to LSS users. In fact, it is better (i.e. cheaper) for all users to become accustomed to using these commands, rather than $RUNning the older programs. The following is a list of public file utility programs whose functions have been incorporated into the MTS command language:

Public File Name	MTS Command
*CATALOG	$FILESTATUS
*DISMOUNT	$RELEASE
*ED	$EDIT
*FILESNIFF	$FILESTATUS
*FILEUSE	$FILESTATUS
*HBQ	$SYSTEMSTATUS
*MOUNT	$MOUNT
*PERMIT	$PERMIT
*PROJECTACCOUNT	$ACCOUNTING
*TASKS	$SYSTEMSTATUS
*USERS	$SYSTEMSTATUS

MERIT-HERMES SYSTEM

The UMnet 'Communications Computer' or 'CC' provides a data link between the University of Michigan's IBM ES3090/300J 'Host' Computer and 'Host' Computers at Michigan State University, Wayne State University and Western Michigan University.

CCs can be Primary Communications Processors (PCPs) or Secondary Communications Processors (SCPs). There are now several PCPs at the University of Michigan Computing Center.

The U. of M. PCPs are connected by leased phone lines to similar PCPs at MSU, WSU and WMU in addition to many SCPs located in Ann Arbor and around the state. The PCPS at the Computing Center are also connected to each other by synchronous communications links. The function of a PCP is to allow several users to simultaneously transfer data to or from a remote host on the network. At the U. of M.. the PCPs are interfaced to the ES3090/300J via a multi-address, host-interface device attached to a ES3090/300J channel. The PCP is connected to the network via two telecommunications transmitter-receiver pairs; one pair for each line to another PCP or SCP. A PCP CPU is at the heart of the PCP. The PCP executes the communications computer operating system called "CCOS" which co-ordinates the different devices in the PCPs.

Some of the major functions performed by CCOS are...

1) The multiplexing of information over the host-interface and over the phone lines.

2) The automatic retransmission of erroneous or lost data over the phone lines.

3) The automatic re-routing of data over a second phone line if the primary line becomes disabled.

HERMES SYSTEM

The UMnet Communications Computers are equipped with hardware and software which enables terminals to be dialed in directly to them. This facility has been named HERMES. Hermes appears to a user as a Data Concentrator servicing several hosts on the network.

After a terminal is dialed into Hermes, and its type and speed have been determined, the user is prompted for the name of the particular host. Hermes provides the user with a common front-end to all the hosts on the network; users can access remote hosts as if they were local and can access these hosts with the same set of keyboard editing characteristics.

To the user, Hermes appears to greatly resemble a Data Concentrator. Some of the code for the support of Hermes is from the PCP MCS system. The terminal identification procedure is almost identical to that of the Concentrator and the keyboard editing characters are essentially the same. The hardware (line adapters and Modems) is similar to that of the Concentrator, although there are additional synchronous devices for the Inter-Nodal communication.

From an operational viewpoint, it is very important to realize that although MTS may be down locally, users may still be accessing other hosts via Hermes. Therefore, do not power down or halt any of the UMnet PCPs or busy out or power down any of the MERIT modems unless it is essential to do so.

Currently, there is no simple way to broadcast to Hermes users about impending shutdowns, etc. Such a broadcast will be installed shortly. Messages broadcast from the MTS operators' console are not seen by Hermes users accessing hosts other than the one at the University of Michigan. However you can "broadcast" by using the Super Console:

```
   First select the node name desired
   Then type

     %BROADCAST message       (once for each PCP desired).
```

SETDATE

Contents: The object module of a program to set the time and date in UMnet PCPs.

Purpose: To set the time and date in a UMnet PCP, obtain a reason for the reload, print a nicely formatted message on the PCP operators' console, optionally log the time, date and reason for reloading on logical I/O unit 0 and optionally reset any reply message for the primary host from which SETDATE is being run.

Location: MNET:SETDATE

Use: The program is invoked by using the $RUN command

Logical I/O units referenced:

SCARDS - response to the "Enter Reason For Reloading" prompt.

SPRINT - prompt if the reason for reloading is not given in the PAR field of the $RUN command.

0 - optional; if assigned one line giving the time, date and reason for reloading is written to this logical I/O unit followed by a log of the commands read from logical I/O unit 1, if any.

1 - optional, if assigned lines are read from this logical I/O unit and passed to the PCP using the CONTROL subroutine.

Parameters:

PAR = {cu=xx} hostname {reason for reloading}

The PAR field may begin with a network control unit name (AA, AB, AD, AE, or AN). It must also include a network host name (this is normally "N"). The reason for reloading can be optionally specified also.

Description:

This program is normally run from *MNS when the MTS job servicing UMnet interactive and terminal jobs is started from *xDL after the UMnet PCP has been dumped and reloaded and from *MSD at the U. of M. each night at midnight to set the dates in all UMnet PCPs.

The program obtains the current date using the TIME subroutine, obtains a "reason for reloading" from the PAR field of the $RUN command or by prompting on SPRINT and reading one line from SCARDS, mounts a privileged outbound UMnet CC connection, resets any reply message for the primary host in which it is running if a reason, but no host name was given in the PAR= field, sets the time and date in the PCP, and prints a nicely formatted message which includes the time, date and reason for reloading on the PCP operators' console. If logical I/O unit 1 is assigned, lines are read and passed unchanged to the MTS CONTROL subroutine. These lines will normally begin with a "2", "3" or "5" followed by a CC command and could be used for example to reset a Hermes message, (HMESS), after the PCP is reloaded. If logiacal I/O unit 0 is assigned, the file or device is used to log a one line message giving the time, date and reason for reloading followed by any commands read from lodical I/O unit 1 together with any output that the commands may produce. When the program terminates, the network connection is released.

Examples:

```
$run mnet:setdate par=cu=an,n Al told me to!

$run mnet:setdate par=NDT

$run mnet:setdate par=cu=ae,n
```

MISCELLANEOUS MESSAGES

Following are miscellaneous messages that may appear on the operator's console. HASP messages are not included; see the HASP Operator's Manual. Machine check and similar messages presented earlier in this manual are likewise not included.

```
******************* OPERATOR MESSAGES *******************

>>> * * ACCTNG ERROR READ/WRITE-SIGNON

 User Explanation:
     System error - see an MTS systems programmer.

System Explanation:
     During signon processing an error return was received  from  either
     GETACCR

   (RC=8) or PUTACCR (RC =0).

       DESTINATION: OPER
           TEXTLOC: CMDS,636,SYACCRAR(SYGNE),CMDS1088
            SOURCE: CMDS,636,SYACMR(SYGNE),CMDS1086


>>> * * ACCTNG/STAT ERROR READ/WRITE - TSA

 User Explanation:
     System error - see an MTS systems programmer.
 System Explanation:
     An   error   was   flagged  by  the  file  routines  when  the  TSA
     (TAILSHEET/STATISTICS/ACCOUNTING) subroutine tried to read or write
     the accounting files or the statistics file.  A SNARK follows.

       DESTINATION: OPER
           TEXTLOC: MTS,42,SYACCWAR(TSA2),MTS02841
            SOURCE: MTS,42,TSA2,MTS02839


>>> CATALOG WAS EXTENDED SUCCESSFULLY

 User Explanation:
     No problems, just informative.

       DESTINATION: OPER
            SOURCE: FILE SYSTEM,482/23,GETSEG(CATALOG SUBR)


>>> DEVICE BUSY OR NOT AVAILABLE

 User Explanation:
     This is the system message given to the operator when the  operator
     (in response to a mount request) specifies a drive which is already
     in use or is off-line.

       DESTINATION: OPER
           TEXTLOC: MOUNT,101,DEVBY(MOUNTCMD),MNT02808
            SOURCE: MOUNT,101,ST234(MOUNTCMD),MNT01685
```

>>> DEVICE TYPE NOT ALLOWED

User Explanation

See an MTS systems programmer.

System Explanation:

When starting up an MTS job at the operators' console, either a device name representing a type of device not supported by MTS was specified to be used as MSOURCE or MSINK (or PUNCH output, if a batch job was started), or else the device specified for MSOURCE was not an input device (error return from CFID) or the device specified for MSINK or PUNCH was not an output device (error return from CFOD).

```
DESTINATION: OPER
    TEXTLOC: MTS,42,GRRM2(GRRR2),MTS02679
     SOURCE: MTS,42,OUTNBY(GRRR2),MTS02673
     SOURCE: MTS,42,GTMSOU(START),MTS00254
     SOURCE: MTS,42,INITCP(START),MTS00304
     SOURCE: MTS,42,INIT67(START),MTS00326
```

>>> DSECT CLOBBERED

User Explanation

System error - see an MTS systems programmer.

System Explanation:

TSA thinks that the MTS job is a free run (via operator request) but the job is not a HASP job. This preceeds a Snark message.

```
DESTINATION: OPER
    TEXTLOC: MTS,42,CHKSM(CHKSER),MTS02834
     SOURCE: MTS,42,CHKSER,MTS02832
```

>>> ERROR WRITING PAT OR DSCBS - CALL SYSTEMS PROGRAMMER IMMEDIATELY

User Explanation

When a disk error occurs while extremely sensitive data concerning disk storage allocation is being written, this message ensues. Under NO circumstances should the message be ignored as it may result in disaster if the file routines use possibly incorrect data resulting from the hardware error. Call a systems programmer immediately!

```
DESTINATION: OPER
     SOURCE: FILE SYSTEM,482/19,GETDSK
```

```
>>> FDGTPD ERROR

 User Explanation
     System error - call an MTS systems programmer.

 System Explanation:
     The FDGET2 pushdown stack underflowed.

       DESTINATION: OPER
           TEXTLOC: FSUB,635,HOPPDMSG(HOPENIT),FSUB0883
            SOURCE: FSUB,635,HOPEOOPS(HOPENIT),FSUB0879


>>> FREE RUN: nnnnnn xxxx

 User Explanation
     The  batch  job  with  recipt number "nnnnnn" and user id "xxxx" is
     being signed off without being charged.

 System Explanation:
     For example, due to the operator's answering a mount  request  with
     RER (rerun).

       DESTINATION: OPER
           TEXTLOC: MTS,42,TSAFREE(TSA),MTS06263
            SOURCE: MTS,42,TSAFREE(TSA),MTS06269


>>> FREESPACE MOUSE TRAP DUMP

 User Explanation
     System error - see an MTS systems programmer.

 System Explanation:
     A  system  component  (in  system  storage)  has  issued an invalid
     FREESPAC request.  Sorry about that!  A job dump is given.

       DESTINATION: OPER
           TEXTLOC: GETSPACE/FREESPAC,41,GFDMPMSG(FREESPAC),GSFS0755
            SOURCE: GETSPACE/FREESPAC,41,GFDMP(FREESPAC),GSFS0748


>>> HELP - ERROR WHILE EXTENDING CATALOG

 User Explanation
     Serious trouble - call a systems programmer.

       DESTINATION: OPER
            SOURCE: FILE SYSTEM,482/23,GETSEG(CATALOG SUBR)
```

>>> HELP -- SNARK IN MTS

User Explanation

System error - see an MTS systems programmer.

System Explanation:

This infamous message occurs only on the operators' console. It occurs when MTS has received a "fatal exit" (such as a program interrupt) from the supervisor and no input line has been requested from the user since the last such "fatal exit" (so we have a potential loop). The particular MTS task causing the snark is put into a permanent wait (so it can be stopped). No other task in the system is affected.

```
  DESTINATION: OPER
      TEXTLOC: MTS,42,HELPMSG(OOPS),MTS02872
       SOURCE: MTS,42,OOPSNA(OOPS),MTS02863
```

>>> ILLEGAL INPUT

User Explanation

Printed on the operator's console when, after GOOSEing a batch job, he enters an invalid response. The only legal responses are $DUM, $NEX, or $ATT.

```
  DESTINATION: OPER
      TEXTLOC: MTS,42,ZTPOPRM(ZTPA),MTS04530
       SOURCE: MTS,42,ZTPRDM(ZTPA),MTS04518
```

>>> ILLEGAL REPLY

User Explanation

This error message is printed when the operator has given an improper response to a MOUNT request.

```
  DESTINATION: OPER
      TEXTLOC: MOUNT,101,ILLREP(MOUNTCMD),MNT02802
       SOURCE: MOUNT,101,ST101(MOUNTCMD),MNT01505
```

>>> IMPROPER DEVICE NAME

User Explanation

This is the system message given to the operator when the operator replies to a mount request and specifies an invalid drive name, e.g., T990 instead of T909 for a 9-track tape drive name.

```
  DESTINATION: OPER
      TEXTLOC: MOUNT,101,BADDEVN(MOUNTCMD),MNT02805
       SOURCE: MOUNT,101,ST106(MOUNTCMD),MNT01648
```

>>> IMPROPER DEVICE TYPE

User Explanation

This is the system message given to the operator when the operator mounts a tape and specifies the wrong type of drive (e.g., 7TP instead of 9TP) or a drive which doesn't support the mode specified by the user via the MODE= parameter (e.g., a 6250 bpi drive when MODE=800 is specified).

```
  DESTINATION: OPER
      TEXTLOC: MOUNT,101,BADDEVT(MOUNTCMD),MNT02807
       SOURCE: MOUNT,101,ST33(MOUNTCMD),MNT01678
       SOURCE: MOUNT,101,ST114(MOUNTCMD),MNT01781
```

>>> INCORRECT OR MISSING RACK NUMBER

User Explanation

This is the system message given when the operator replies with an incorrect rack number where it is required in a reply to the $MOUNT routines, e.g., "CI324 T900" instead of "C1324 T900" to mount tape C1324 on drive T900.

```
  DESTINATION: OPER
      TEXTLOC: MOUNT,101,BADRNUM(MOUNTCMD),MNT02804
       SOURCE: MOUNT,101,ST104(MOUNTCMD),MNT01594
```

>>> IT'S TIME TO ADD ANOTHER PUBLIC VOLUME - METHINKS

User Explanation

It is desirable to keep a minimum number of pages available on each pack for expansion purposes (currently 500 pages). When this message appears, ALL packs are below the minimum. The appropriate person should be notified to add another public volume.

```
  DESTINATION: OPER
       SOURCE: FILE SYSTEM,482/17,CREATE(OPEN, etc.)
```

>>> JOB nnnn WAITING ON FULL SHARED FILE TABLE.

User Explanation

The system keeps track (a pun?) of all currently active files in a table in shared memory. When this message occurs, either the table has overflowed legitimately and hence should be increased in size, or some program has "gone wild" and "activated" a large number of files. In the latter case, the job should be stopped. A systems programmer should be notified to determine which condition occurred.

```
DESTINATION: OPER
     SOURCE: FILE SYSTEM,482/027,ICLOCKx/ICUNLKx
```

>>> MOUNT MESSAGES TO THE OPERATOR

User Explanation

Messages to the operator for $MOUNT requests and reports of mount errors are produced in this section.

```
DESTINATION: OPER
    TEXTLOC: MOUNT,101,ST12...(MOUNTCMD),MNT01270
     SOURCE: MOUNT,101,ST12...(MOUNTCMD),MNT01270
```

>>> *** MTS DEBUG JOBDUMP FOR SYSTEMS ***

User Explanation

The operator is notified that the MTS debug job dump is occurring.

System Explanation:

The SETEXIT entry into MTS from the supervisor was taken and the EXITDUMP flag was on with appropriate exit type and address match.

```
DESTINATION: OPER
    TEXTLOC: CMDS,636,EXITDMSG(ERRRTN),CMDS1361
     SOURCE: CMDS,636,EXITSVC0(ERRRTN),CMDS1223
```

>>> NO SPACE FOR RELOCATION ENTRIES - VOLGET

User Explanation

This message may, but shouldn't, occur after adding a new pack to the system. It should be followed by the message "THE FOLLOWING VOLUME(S) HAVE NOT BEEN FOUND: ...". The operator should respond with CANCEL and call a systems programmmer.

```
DESTINATION: OPER
     SOURCE: FILE SYSTEM,482/5,VOLGET
```

>>> NOT A HASP JOB

User Explanation

This error message is printed when the operator has given a "RER" response to a MOUNT request by other than a HASP job. If, for example, all drives of that type (e.g., 9-track tape drives) are in use, the operator should use "BY" (for busy) for a nonHASP job and "RER" for a HASP batch (HB) job.

```
  DESTINATION: OPER
      TEXTLOC: MOUNT,101,NOTHSP(MOUNTCMD),MNT02803
       SOURCE: MOUNT,101,ST109(MOUNTCMD),MNT01518
```

>>> *** "OUT" STACK ERROR

User Explanation

System error - see an MTS systems programmer.

System Explanation:

The 3-level stack for the subroutine OUT etc., has either overflowed or underflowed during the SINK/MSINK error printing recursion.

```
  DESTINATION: OPER
      TEXTLOC: MTS,42,OVERM(OUT),MTS02238
       SOURCE: MTS,42,OUTCKSS(OUT),MTS02231
```

>>> POSITIONING FAILED

User Explanation

The batch job not specifying QUIT=NO will print this message to the operator if the requested tape cannot be positioned as the user specified, either because the specified data-set name did not exist or the file number specified was greater than the number of files on the tape. The tape will be unloaded and the job terminated.

```
  DESTINATION: OPER
      TEXTLOC: MOUNT,101,POSNCOM(MOUNTCMD),MNT02430
       SOURCE: MOUNT,101,ST174/ST225(MOUNTCMD),MNT01879/MNT01882
```

```
>>> RATE INITIALIZATION ERROR.

 User Explanation
     System error - see an MTS systems programmer.

 System Explanation:
     The rate vector RATEVEC could not be initialized by GETRATES  (from
     *RATEFILE) during MTS task initialization.

       DESTINATION: OPER
           TEXTLOC: MTS,42,INITERR,MTS00357
            SOURCE: MTS,42,INITERR,MTS00358


>>> REMOUNT TAPE WITH RING how

 User Explanation
     This  is the system message given to the operator when the operator
     mounts a magnetic tape with the file-project ring improperly placed
     (e.g., OUT instead of IN).  "how" will be either "OUT" or "IN".

       DESTINATION: OPER
           TEXTLOC: MOUNT,101,ST113(MOUNTCMD),MNT01770
            SOURCE: MOUNT,101,ST113(MOUNTCMD),MNT01770


>>> SIGNOFF QUIT

 User Explanation
     System error - see an MTS systems programmer.

 System Explanation:
     This message occurs if ICDEQ (in-core file table routine) gives  an
     error  return through DEQIT to SYGNE while a signoff job cleanup is
     in progress.  An MTS SNARK follows.

       DESTINATION: OPER
           TEXTLOC: CMDS,636,SYGNUGN(SYGNE),CMDS0610
            SOURCE: CMDS,636,SYGNUGN(SYGNE),CMDS0611
```

>>> *** *STATISTICS IS FULL ***

User Explanation

Use *STA to dump *STATISTICS on tape.

System Explanation:

This message is written to the operators' console whenever a line is written into the file *STATISTICS (which is whenever anybody signs off) and there are more than 7500 lines in the file already. The operator should dump *STATISTICS on tape via MTS *STA before the file gets really full and jobs start SNARKing.

```
DESTINATION: OPER
    TEXTLOC: MTS,42,MAXSTM(PUTSTATR),MTS05269
     SOURCE: MTS,42,PUTSTATR,MTS05237
```

>>> TAPE ERROR ON VOLUME LABEL

User Explanation

This is the system message given to the operator when the specified tape drive is unable to read the first block on the (to verify the presence or absence of a volume label). The operator may then try another drive or give an "LE" response to give the user an error message.

```
DESTINATION: OPER
    TEXTLOC: MOUNT,101,VOLERR(MOUNTCMD),MNT02809
     SOURCE: MOUNT,101,ST183(MOUNTCMD),MNT01816
```

>>> TAPE UNIT DOES NOT SUPPORT DENSITY OF VOLUME LABEL

User Explanation

The user specified a $MOUNT of a tape and the drive that the operator specified did not support the actual mode of the tape. If the user did not specify some density which is unavailable on a drive which also supports the actual density of the tape, then the operator should try again on that drive; otherwise, an "IM" reply should be given (or "BY" if all such drives are busy).

```
DESTINATION: OPER
    TEXTLOC: MOUNT,101,BADDENS(MOUNTCMD),MNT02810
     SOURCE: MOUNT,101,ST253(MOUNTCMD),MNT01807
```

>>> THE FOLLOWING VOLUME(S) HAVE NOT BEEN FOUND: xxxxxx RETURN TO RETRY, OR TYPE "CANCEL" IF NOT AVAILABLE

User Explanation

Generally means a pack is not mounted or the drive is not ready. If preceded by a unit-check message, try moving the pack to a different drive. If it still fails, call a systems programmer.

DESTINATION: OPER
SOURCE: FILE SYSTEM,482/5,VOLGET

>>> USER: xxxx (yyyy) FILE zzzz

User Explanation

Whenever anyone modifies a public file, this message is printed on the operator's console followed by the first line inserted into the file. The "xxxx" and "yyyy" are the two forms of user id kept in the MTS dsect. They should be identical. "zzzz" is the name of the public file being changed.

DESTINATION: OPER
TEXTLOC: DSRS,637,WOD1(WODISK),DSRS2386
SOURCE: DSRS,637,WOD1(WODISK),DSRS2394

>>> VOLUME vvvvvv, PVN xx NOW FOUND ON Dyyy. PLEASE CONFIRM.

User Explanation

May result from moving a pack to a different drive, but probably means a spurious pack, such as a test pack, has been found. After verifying that the indicated drive contains the right pack (do not move any packs), confirm by typing OK.

DESTINATION: OPER
SOURCE: FILE SYSTEM,482/5,VOLGET

>>> VOLUME NAME INCORRECT

User Explanation

This is the system message given to the operator when the tape that the operator has mounted in response to a mount request does not have the label that was specified in the the request.

DESTINATION: OPER
TEXTLOC: MOUNT,101,BADVOL(MOUNTCMD),MNT02806
SOURCE: MOUNT,101,ST182(MOUNTCMD),MNT01812

MYLAR RIBBON COPIES

The following procedure may be used for producing mylar ribbon copy on the TN printer

(1) $DRAIN printer to be used (a TN printer)

(2) Change ribbon and print train or chain:

a. Remove cloth ribbon

b. Be sure a TN train or chain is mounted

c. Install mylar ribbon

(3) Reverse paper to side without lines.

(4) Set TN print buffer: "TN PTRX ECHO",X being the number of the printer to be used.
This is a "JOBS" command and must be entered from Operators console.

(5) Adjust timing and print density.

a. $RUN MTA.:TNRIP(1000) SPRINT=>PTRX

b. While TNRIP is running, adjust forms thickness and print density. The characters [and] are best used for the adjustment.

c. $UNL above program to release the printer. The program is stopped by an attention interrupt.

(6) Adjust paper alignment

a. Set printer to neutral

b. Press CHECK RESET and CARRIAGE RESTORE

c. If TEXT360 manual output, move paper down one line.

d. Set printer to "DRIVE".

OLTS

(Aug. 6, 1974, VERSION)

There is a version of OLTS available in MTS which is the same as in release 21.7 of OS. The procedures required to run it are somewhat different than before and are given below.

To run OLTS, signon under any staff userid and enter

```
$source mta.:oltepcmd
```

This will load a special version of FAKEOS and start OLTEP. When this is completed, OLTEP will print "IFD102I OLTS RUNNING" and further communication will be as described in IBM Manual C28-6650-6 on pages 41-47. The operation of OLTEP in MTS is as described in this manual except for the following:

1. Pages 1-40 describe how to install and invoke OLTEP in OS and are irrelevant. They are replaced by the paragraph above.

2. The procedure required to stop running a test is to press the attention key on your terminal, instead of the procedure described on page 47.

3. All communication with OLTEP is done via your terminal, not the Operators console.

4. It is not necessary to type "r 00,'xxx'", since "xxx" is treated exactly the same.

5. All devices to be tested must be ONLINEd in order for OLTS to work. The Channel(s), Control Unit(s), and paths to the devices may be OFFLINEd if desired to keep normal system I/o from them.

If anyone has more test sections they wish to have added, they should give them to M. T. Alexander.

The printed output from OLTEP (referred to as the DIAGMSG DD in the manual) will be put in the temporary file -DIAG which may be copied to *PRINT* after OLTEP terminates.

NOTE: Some of the following information, though re-worded, was taken from IBM volume C28-6650-6.

When OLTEP is running, it will send you the message

```
          IFD102I  OLTS RUNNING

          IFD134I  WARNING - DASD  VOLUME  LABELED  CEPACK   NOT
```

```
PROTECTED FROM WRITE

IFD107I  OPTIONS ARE  FE,NMI,NPP,NEL,CP,EP,PR,NRE
```

For a more complete description of these options, see the IBM publication, C28-6650-6.

PR - (Print) - sends you messages produced by the OLT programs, i.e., test results, requests for operator assistance, etc.

CP - (Control Print) - OLTEP sends you a message letting you know when a test section starts and terminates. The termination message states whether errors were detected.

EP - (Error Print) - OLTEP sends you messages that diagnose errors detected by OLT programs.

FE - (First Error Communication) - When the OLT program detects a device error, OLTEP suspends testing and asks for a new test definition. You can then define a new test, resume testing from the point at which the error occurred, or terminate the job step. If you choose to resume testing, you can change the OLTEP options that apply to the test.

PP(n) - (Parallel Print) - OLTEP sends output to you as well as to file.

TL OR TL(N) - (TEST LOOP) - REPEAT TEST N TIMES UP TO 32767.

EL - (Error Loop) - When an OLT program detects an I/O error, OLTEP repeatedly executes the program that produced the error. Execution is repeated n times up to 32767. n defaults to 32767.

MI - (Manual Intervetion) - OLTEP bypasses routines which require manual intervention unless MI is specified.

<u>NOTE</u>:The MI and RE options are mutually exclusive.

RE - (Remote) - For sending message to and receiving replys from a remote site when RETAIN/370 is active. We don't use this. The MI and RE options are mutually exclusive.

```
IFD105D  ENTER-DEV/TEST/OPT/
```

Message IFD134I warns you that any user pack that is physically on the system and that has the serial number CEPACK, will be written upon if the volume is accidentally entered in response to message IFD105D.

Message IFD107I tells you the OLTEP options which are currently in effect. In this case, the default options will be listed because you have not enterd a test definition.

Message IFD105D asks you to define the first test; it will be repeated when the test is completed.

```
Example:  180-184/2400/
```

This will test units with address 180 to 184. Run basic tape test. Use standard OLTEP options. OLTEP will run the test and then issue messages IFD107I and 00 IFD105D.

When you receive message 00 IFD105D, you can define the next test to be run using devices/tests/options/.

CANCEL terminates the program.

When the ENTER-DEV/TEST/OPT/ message appears, you can type one of the following-

```
prompt dev
prompt test
prompt opt
```

You will be given examples and other helpful information when you do this.

```
IFD147I Examples of Device Field

IFD147I 0181/          test device 181
IFD147I 185-187/       test device 185, 186, 187
IFD147I CHICAGO|,TP2/  TEST SYMBOLIC TP DEVICES
IFD147I                CHICAGO1 AND TP2
IFD147I .NDR/          NO DEVICE REQUIRED FOR TEST
IFD147I .CH282/        TEST CH2 USING DEVICE 82
IFD147I / (SLASH ALONE)
                       TEST PREVIOUSLY SELECTED DEVICE(S)


IFD148I EXAMPLES OF TEST FIELD

IFD148I 2400/          TAPE TESTS (T2400A-T2400Z)
IFD148I 2400A/         SEC.  A OF TAPE TEST 2400
IFD148I 2400C,2/
        RTN 2, SEC.  C, TEST 2400
IFD148I 2400A-C,E,G/   SEC.  A,B,C,E AND G OF TEST 2400
IFD148I IOEPZZ/        SEC.  ZZ OF EDIT-PRINT FUNCTION
```

```
IFD148I R2540AA/       SEC.  AA OF READER TEST 2540
IFD148I / (SLASH ALONE) RUN PREVIOUSLY SELECTED TEST(S)


IFD149I                             TABLE OF OPTIONS
IFD149I                     TO REQUEST    TO OMIT    BY
IFD149I                       OPTION    OPTION  DEFAULT
IFD149I TESTING LOOP           TL       NTL       NTL
IFD149I                         TL(VALUE) VALUE=1-32767
IFD149I ERROR LOOP             EL       NEL       NEL
IFD149I                         EL(VALUE) VALUE=1-32767
IFD149I ERROR PRINT            EP       NEP       EP
IFD149I CONTROL PRINT          CP       NCP       NCP
IFD149I PARRALEL PRINT         PP       NPP       NPP
IFD149I                         PP(LEVEL) LEVEL=0-3
IFD149I PRINT                  PR       NPR       NPR
IFD149I FIRST ERROR            FE       NFE       FE
IFD149I COMMUNICATION
IFD149I MANUAL INTERVENTION    MI       NMI       NMI
IFD149I REMOTE FE CONTROL      RE       NRE       NRE
IFD149I EXTERNAL DATA          EXT=

IFD149I EXAMPLES OF OPTION FIELD
IFD149I PP,NMI,RE/
IFD149I EP,TL(50),FE,EXT=A,B/
```

PLOTTING SYSTEM

NOTE:

- There is a ring binder below the operator's console which contains two techical manuals to be referred to for pen cleaning, assembling, etc. One of the manuals is for the CalComp model 1051. The other, a two section manual, is for the model 936. The second section of the 936 manual (Pen and Paper Users Guide) shows as examples pens that we don't use.

- For information on deferred plots, read the material in the section entitled *CCD.

- For information on plot tape initialization, see the descrition of *CCL in this manual.

- For information on the length of the impending plot session and the number of user tapes involved, see *CCT.

- To print indentifying information about a plot tape (actually, from it) including its name (e.g.,"TUE 1000"), when it was last written, and the minimum time to reuse, see *PLB in this manual.

Note: To get a copy of the current plot receipt sheet, $copy sys.:plotreceipt to a terminal or printer.

Plot Times

Plot sessions are started automatically by the "clock watcher" program. These sessions are started at the following times:

Plots with special pen sizes - 8:30am, Mon. - Fri.

Regular plots:

- Mon. - 10am, 2pm, 6pm & 10pm

- Tues. - Fri. 2am, 10am, 2pm, 6pm & 10pm

- Sat. & Sun. - 2pm

Plotting

When it is time to run plots, two procedures must be followed. The first of these follows and the second (the running of PLT36 or PLT51 appears later in this section.

1) Type "mts *ccp" on the operators console. (This procedure has been started automatically by the "clock watcher" program for some time now; now being 4/18/85.)

The system will ask where a 9-track, labelled plot tape can be found.

T9XX (operator types) X being the tape unit on which the requested tape is mounted

```
<<< Tape Description- (First file only)
9TP,1600 bpi,label=PLOT,fmt=vb(6370,244) >>>
```

The plots listed in *CCPFILE will be dumped onto the tape. A receipt number will be printed on the console. The printer output from this job consists of two lists of plots. You can check this list against the plots being plotted. This sheet will also have the sizes and colors of the pens to be used. The pen size is printed on the side of the pen. The clerks will separate the plots and wrap them.

Note: The plot receipt list is also entered, at this time, into a file named SYS.:PLOTRECEIPT from which it can be copied if the original list is lost.

Time Frames for plot tapes

```
               TAPE

0900 MON - 1345 MON (MON-10AM)
1345 MON - 1745 MON (MON-2PM)
1745 MON - 2145 MON (MON-6PM)
2145 MON - 0145 TUE (MON-10PM)
0145 TUE - 0900 TUE (TUE-2AM)
0900 TUE - 1345 TUE (TUE-10AM)
1345 TUE - 1745 TUE (TUE-2PM)
1745 TUE - 2145 TUE (TUE-6PM)
2145 TUE - 0145 WED (TUE-10PM)
0145 WED - 0900 WED (WED-2AM)
                    (WED-8:30am)-Special
0900 WED - 1345 WED (WED-10AM)
1345 WED - 1745 WED (WED-2PM)
1745 WED - 2145 WED (WED-6PM)
2145 WED - 0145 THU (WED-10PM)
0145 THU - 0900 THU (THU-2AM)
0900 THU - 1345 THU (THU-10AM)
1345 THU - 1745 THU (THU-2PM)
1745 THU - 2145 THU (THU-6PM)
2145 THU - 0145 FRI (THU-10PM)
0145 FRI - 0900 FRI (FRI-2AM)
0900 FRI - 1345 FRI (FRI-10AM)
```

```
                  1345 FRI - 1745 FRI (FRI-2PM)
                  1745 FRI - 2145 FRI (FRI-6PM)
                  2145 FRI - 0145 SAT (FRI-10PM)
                  0145 SAT - 0900 SAT (SAT-2AM)
                  0900 SAT - 0900 MON (WEEKEND)
```

If the software senses that the wrong tape has been mounted, the following comment will appear on the operators' console:

```
                  TAPE IS FOR MON 1000

                  OK TO USE?
"yes" or "no" or "ok"
```

A "no" answer will result in a request for another tape.

If *CCP is run twice in the same plot <u>time</u> <u>frame</u>, the following message will appear:

```
    TAPE LAST USED mm/dd/yy  hh/mm/ss;  WILL APPEND NEW PLOTS

                  OK TO USE?

"yes" or "no" or "ok"
```

If a "no" is given, a new tape will be asked for. Occasionally, a user tape will be mounted by the plot routines while *CCP is running. This tape contains the user's file. If the plot routines attempt to mount one of these tapes during *CCP, and the tape cannot be mounted, see the following:

Reason: tape already in use or no tape drives available

Action: "na" the tape

Reason: -tape can't be found

    -gets a label error

    -fatal tape error

    wrong tape id is given

Action: "ab" the tape

"NA" will cause another mount attempt after all the plots have been done, then (if the tape still can't be mounted) the plot will be put back in the queue.

"AB" causes the user to get no second try and no plot. The plot is not requeued.

When *CCP has finished running, it will print out a comment containing information on how many minutes of plots there are and what color pens are to be used.

Example:

```
*CCP DONE.  APPROXIMATELY 410 MIN. PLOTTING; BLUE, RED USED
```

QUICK PLOTS-(*PLQ)...(PLOTQUIK)

This program processes plots for which the user has requested "quick" or ball point pen service. When this program is started, it will check the queue for such plots, and plot any that it finds. It will then wait for more plots. When more plots are queued, it will plot them and then wait again. It will continue to do this until it is stopped. Plots processed by PLOTQUIK are put onto the plot tapes by CCP with the normal plots (PLT handles these differently-- see the PLT writeup).

Running the program

Set up the plotter with the black, blue, red and optional green ballpoints in holder positions 1, 2, 3 and 4 respectively.

Start PLOTQUIK as an MTS *-file job: enter "mts *plq" on the console. (PLOTQUIK can also be started by signing on and entering a $RUN command; (see the file *PLQ for an example). PLOTQUIK will prompt for the name of the plotter to use. Enter PLT1 or PLT2. The program will then begin plotting as soon as there are plots to do. PLOTQUIK can be left running all day if you like.

If *plq fails to wake up, an operator must get it going manually by interrupting the program and issuing a "wake".

The only way to get the program to change plotters is to stop it and start it again.

To stop the program, enter a STOP command to it (see the description of commands below) or STOP the job. It is ok to stop PLOTQUIK while it is in the middle of a plot; that plot will be replotted the next time the program is run.

Every 30 minutes (you can use the RIP command to change the interval), PLOTQUIK will print a reminder to remove the finished plots and then will wait for a response from you. When you enter something, a carriage return will suffice, plotting will resume.

COMMANDS

PLOTQUIK will prompt for a command if you GOOSE the job or after it has finished executing a CON command. It prints a prompting message which contains a list of the commands. After you enter a command, the program will execute it, and then either prompt for another or begin plotting.

Enter an end-of-file to tell PLOTQUIK to continue plotting (or napping).

The commands are:

CCP [receipt]

Plot [receipt] will be changed form a "quick" plot to a "normal" plot. PLOTQUIK will not plot it, and PLT will plot it after it has been dumped to tape. If [receipt] is omitted, the plot currently being plotted is assumed.

Example: CCP 716253

CON

PLOQUIK will finish plotting the current receipt, then stop plotting and prompt for a command. (Note that PLOTQUIK stops after the receipt is done, while PLT stops after the block is done.)

MSG--(Plot Message)

"plot message" will be written onto the end of the plot currently being plotted. Use this command to send a message to the user.

Example:

```
MSG  redrawing a line many times digs holes in
the paper.
```

MTS

This command has the obvious meaning. Note that if you started PLOTQUIK with MTS *PLQ, MTS will cause it to stop.

PEN

Use this command to change the assignment of pen colors to pen holder slots. PLOTQUIK will prompt for the new assignments. Enter the holder slot numbers, in order, for black, blue, red, and green (e.g., the first number is the position of the black pen). Use 0 to indicate that a color is not being used. Example: 2 1 3 assigns black to the second holder, blue to the first, red to the third and green is not used (trailing 0s may be omitted). 2 0 3 assigns black to the second slot, red to the third. Blue and green aren't used in this situation. The assignments take effect at the next pen change (which may be sooner than the next plot. Plots which require colors that are not in use will remain in the queue until you make those colors available.

PRI n

This command sets the pen priming interval. "n" is in seconds. When PLTQUIK changes form one pen to another in a plot, it will prime the new pen only if it has been "n" seconds since that pen was last used. (This command has the same meaning that it does for PLOTCON.) If you are using ball point pens (the usual case for this program) you will probably not need to use this command.

RIP N

Use RIP to set the frequency of the "rip off the plots" reminder message. "n" is in minutes. PLOTQUIK will print the message about every "n" minutes and wait for a reply. After you reply (a carriage return will do) it will continue plotting. Initially, "n" is 30.

REP [receipt]

This means replot receipt [receipt], or the current receipt if [receipt] is omitted. Use this command to replace a defective plot any time before the plot is dumped to tape (after that, use PLT (most likely PLT36)). PLOTQUIK will not begin plotting the plot immediately (unless it was not doing anything); it will continue with what it was doing, and do the replot in due course. You can enter a REP command without disturbing the plot currently being plotted.

If the QUIK plot that you are interested in has aleady been written on tape, then yuou can do the following:

```
        goose quickplots

        type stop

        r sys.:plot36

        copy 123456 quik
```

STOP

This command terminates PLOTQUIK. If PLOTQUIK was started with MTS *PLQ, the job will stop.

WAKE

When PLOTQUIK is napping (because it ran out of plots to do) this command causes it to check the queue for plots to be done. PLOTQUIK wakes itself up periodically (every ten minutes at the time of this writing) so this command will usually not be necessary.

OPERATIONAL MATTERS

This program is usually started on the 936 plotter each morning, and is permitted to run all day. Of course, it can be stopped at any time. It may pause while CCP is running; it will start up again when CCP finishes. PLOTQUIK can be run on the 1051. Don't run PLOTQUIK on both plotters at the same time or you'll get two copies of everything.

Tear off the plots about every 30 minutes and send them to the DPAs.

The current schedule for this service is 9:30 A.M. to 5:30 P.M., Monday through Friday.

CCP, PLT and PLOTQUIK

PLOTQUIK plots from the queue. After PLOTQUIK finishes a plot, CCP will dump it on a tape with the normal plots. CCP dumps it at the next plot time after the plot is finished, not the next time after it is queued. CCP marks "quick" plots with an * in front of the plot numbers on the receipt sheet. While a "quick" plot remains in the queue, PLOTQUIK can replot it. After it has been dumped to tape, PLT must be used to replot it.

When a COPY command is issued to PLT, it will copy the "quick" plots into the file with the normal ones but will not plot them. To plot "quick" plots, specify QUIK at the end of a plot or COPY command.

Example:

```
COPY ALL QUIK
```

plots all the "quick" plots on the tape. When QUIK is specified, only "quick" plots are plotted.

UNUSUAL CONDITIONS- Plotting System

*CCP Dies

If *CCP dies, several registers and remarks will be displayed on the operators' console. Bring these to the attention of the appropriate systems programmer, for the time being, Gail Lift. Once *CCP has been started and the plot tape has been "accepted" by the system, *CCP cannot be restarted. See Gail Lift if a restart is necessary.

ERROR MESSAGES FROM THE PLOTTING PROCESSOR

This section explains the error messages which the plotting postprocessor prints on the "plot receipt sheet". The postprocessor will print one of these messages whenever it is unable to produce a plot.

USER TAPE UNAVAILABLE

> This tape containing the user's plotfile could not be found by the operator at the scheduled plotting time. The most likely cause is that the tape has been picked up by its owner.

HARDWARE/SOFTWARE ERROR IN FILE

> The postprocessor encountered a hardware error or software inconsistency in reading the file containing the plot. Have the user see the counselor for assistance.

NO ACCESS TO file; PLOT ABORTED

> The file containing the plot was not permitted to the postprocessor. Either the plot file was not permitted properly at the time it was queued (but *CCQUEUE was not able to determine that) in which case *CCQUEUE printed the message
>
> **PERMIT STATUS OF "file" UNKNOWN
>
> before assigning the receipt number, or the permit status of the file was changed after it was queued. In either case, the file must be properly permitted (READ PKEY=*CCQUEUE) and requeued. See the section <u>Queueing</u> in the *CCQUEUE writeup in MTS volume 2.

ERROR IN READING file; PLOT ABORTED

> Some error occured while the postprocessor was reading the file containing the plot. See Gail Lift (Computing

Center, 763-6051) for assistance.

The next two errors occur when the user changes the contents of the file containing the plot after running *CCQUEUE but before the postprocessor is run. *CCQUEUE merely places the plot file name on a queue; the postprocessor actually copies the plot file to a system tape for plotting. See the sections Queueing and Postprocessing in the *CCQUEUE writeup in MTS Volume 2. Sometimes the postprocessor is run late, or the plot file is not copied at the next plot time after it was queued (because the file was locked or special pens were required). The DPA (at the output window) can check the time that the postprocessor was actually run.

file DOES NOT EXIST; PLOT ABORTED

The file containing the plot was destroyed between the time the plot was queued and the time the postprocessor was run (the scheduled plot time).

PLOT file HAS CHANGED - PLOT NOT PLOTTED

The file containing the plot was changed between the time it was queued (*CCQUEUE was run) and the time the postprocessor was run (the scheduled plot time). The usual cause of this is that the user put a new plot in the same file.

The plot time and charges for a plot not plotted for any of the above reasons can be rebated. In most cases, the charges are automatically rebated - but fill out a rebate form anyway just to make sure. In some cases a rebate for additional charges may be justified. Fill out a rebate form and/or talk to Gail Lift (Computing Center, 763-6051). Rebate forms are available at the output window.

DEFERRED PLOTS...(*CCD)

When one user has some particularly long or obnoxious plots, it may be easiest to handle them separately from the normal plot runs. Such plots, queued by the user in the normal way, but placed on a separate plot tape, are called "deferred plots".

To defer a users plot(s), a program, *CCD, described below, is run to tell *CCP that all plots for a particular signon id are to be deferred. After this is done, *CCP will ignore that user's plots when doing the scheduled plot runs (the "deferred" plots will be left in the queue). The deferred plots can be put on a tape, at the operator's convenience by running *CCP using a special plot tape. When it is no longer necessary to defer the user's plots, a program is run to tell *CCP to process the plots normally.

DEFERRING PLOTS

Start the MTS job in the file *CCD. You will be prompted for a ccid. Enter all the ccids for which plots are to be deferred, one per line. Terminate the list with an end-of-file ("CNCL").

Example:

```
      21:10:01 02798..mts *ccd
02798 21:10:04      MTS..ENTER SIGNON ID:
02798 21:10:04      MTS..sys.
02798 21:10:00      PLOTS FOR SYS. WILL BE DEFERRED
02798 21:10:00      MTS..mts.
02798 21:10:05      PLOTS FOR MTS. WILL BE DEFERRED
02798 21:10:11      MTS..**CANCELLED**
```

This program writes information into SYS.:CCPPRIO telling *CCP which ccids to defer plots for. All plots for the ccid will be deferred.

RUNNING REGULAR PLOTS

Run *CCP in the normal way. Deferred plots will be left in the queue, and their receipt numbers will be listed on the plot sheet under "requeued".

RUNNING DEFERRED PLOTS

Run *CCP in the normal way, except use the plot tape marked "deferred plots". This tape is in the rack with the other plot tapes. The deferred plots will be added to the logical end of the tape, i.e., after the plots already there. The system will place only plots that have been deferred on this tape. The deferred plots will remain on the deferred tape until the tape is reinitialized (see below).

PLOTTING DEFERRED PLOTS

Run PLT in the usual way. Note that the deferred tape is added to, so that you may not want to plot all of the plots. The "WHICH TAPE" message from PLT will say that the deferred tape is being plotted:

Example:

```
MTS  PLOTS FROM DEFERRED TAPE; TAPE WRITTEN 13:46.15 03-11-78.

MTS  1=BLAK/4     2=BLAK/3     3=BLAK/6

MTS..ok
```

TO STOP DEFERRING USER PLOTS

This should be done to allow a users' plots to be processed in the normal way after they have been deferred.

Start the MTS job in the file *CCD. You should be prompted for a ccid. Enter all ccids for which you wish plots to no longer be deferred, one per line. Terminate the list with an end-of-file ("CNCL").

Example:

```
02799 21:10:21  ..mts *ccd
02799 21:10:24  MTS..ENTER SIGNON ID:
02799 21:10:24  MTS..mts.
02799 21:10:29  MTS..PLOTS FOR MTS. WILL BE PROCESSED NORMALLY.
02799 21:10:29  MTS..sys.
02799 21:10:30  MTS..PLOTS FOR SYS. WILL BE PROCCESSED NORMALLY.
02799 21:10:30  MTS..**CANCELLED**
```

This is, of course, exactly the program which defers plots. It just changes the status of any id you give it, as:

```
02805 21:10:54 02805..MTS *ccd
02805 21:12:45  MTS..ENTER SIGNON ID:
02805 21:12:46  MTS..sys.
02805 21:12:56  MTS..PLOTS FOR SYS. WILL BE DEFERRED
02805 21:12:56  sys.
02805 21:13:00  MTS..PLOTS FOR SYS. WILL BE PROCESSED NORMALLY
02805 21:13:00  MTS..**CANCELLED**
```

INITIALIZATION OF PLOT TAPES...*CCL

This program initializes plot tapes for use by *CCP.

Run the program by entering

```
    mts *ccl
```

at the operators' console. The program will print

```
    ENTER DRIVE DENSITY NAME
```

Type in the name of the drive, e.g., T903, the density, 1600 or 6250, and the name of the plot tape, e.g., TUE 1400, separated by blanks. The program will label the tape with volume name PLOT and write a file used by CCP to identify the plot tape.

The same information may then be entered for another plot tape. Use an end-of-file to terminate the program.

The name of the plot tape must appear in exactly the right format, e.g., MON 1000 but not MON 1000. The weekday names used are

```
                         MON   TUE   WED

                         THU   FRI   SAT
```

and the times are

```
               OPTS  1000  1400  1800  2200  0200
```

(OPTS refers to the 8:30 am different-pen-size plots.)

The plot tape names WEEKEND and DEFERRED are also recognized.

Examples:

```
    T903 1600 FRI 2200

    T905 1600 WEEKEND
```

NOTE: Use the program *PLB described elsewhere in this manual to display information about the format of a plot tape.

LENGTH OF PLOT SESSION...*CCT

NAME: *CCT

FUNCTION: To let the operators know how much time the plotting session would take, and how many user tapes would be used if the plots were dumped on tape and run at the time *CCT is run.

USAGE: mts *cct

Response on operator's console might be:

```
    PLOTTING TO DO: 27 MIN.  REQUIRING NO USER TAPES.
```

PLOT TAPES--IDENTIFYING INFORMATION...*PLB

NAME: *PLB

FUNCTION: This program prints identifying information about plot tapes, i.e., the name, e.g., "TUE 1000", when it was last written, and the minimum time to re-use.

USAGE: To run from the Operators' console, enter:

```
mts *plb
```

The program will print

```
ENTER TAPE NAME:
```

Respond with the name of a drive on which a plot tape is mounted, e.g.,

```
T911
```

The program will print several lines of tape information and then prompt for another tape name. Enter an end-of-file (CNCL) to terminate the program.

EXAMPLE:

```
mts *plb
ENTER TAPE NAME:
t911
TAPE "MON 2000" LAST USED 11-06-78 22:00:11
```

```
MIN REUSE TIME: 6 DAYS 0 HRS 0MIN.
NEW PLOTS REPLACE OLD.
ENTER TAPE NAME:
```

When running the program from somewhere other than at the Operators' console:

```
$run sys.:pident
```

When prompted for a tape name, enter either the name of the drive, e.g., t901 or, a pseudo-device name, e.g., *tape*. The program reads tape names from GUSER and prints its output on SERCOM.

ERROR MESSAGES:

"NOT A TAPE"

The tape name entered was not the name of a drive (or proper pseudo-device name) or, the drive was busy.

"NOT A PLOT TAPE"

either the volume name was not "plot" or, the name of the first data set was not "IDENT".

"FORMAT ERROR ON PLOT TAPE"

The tape was a plot tape but some error was found in the identifying information in the first file on the tape. The tape should be re-initialized before being written by *CCP.

<u>PLT36</u> <u>and</u> <u>PLT51</u>

The second step of running regular plots, which follows, can't be run from a library file as can the first one since it is necessary that the job be interruptable.

1) Signon using id, PLOT

```
$run sys.:plt51 (or plt36)

EXECUTION BEGINS

WHAT IS TO BE PLOTTED?
```

or

2) signon using id, SYS.

```
$run plt51 (or plt36)

EXECUTION BEGINS

WHAT IS TO BE PLOTTED?
```

When using PLT51, after plots have been dumped onto tape but just before beginning plotting, you will be prompted for a plotter. Responses are PLT2 or PLT3 or 2 or 3. An "$endfile" gets you back to the main prompt. If the requested plotter is busy or unavailable for some other reason, a message will be printed and the plot program will prompt again.

OPTIONS:

a. A single block- example: 003 -plots block 3 only

b. An s-8 number- example: 001476 -plots all blocks associated with s-8 number 001476

c. ALL (plots everything)

d. A range of blocks- example: 004...015 -plots block 4 through block 15

e. A range of blocks- example: 003...end -plots block 3 to eot

f. an EOF (via CNCL button or $endfile) will terminate execution

After option is given, the system will ask for a tape. This is the tape which was produced during *CCP. The system will then read the tape, dismount it, and list the plots found. Then it asks:

OK?

'yes' or 'y' or 'ok' or 'o'

or

'NO'

NOTE: In options b., d. and e. above, block or receipt number ranges can be given in increasing order. They must be in the same order as they appear on the tape. For example , one could state block or receipt numbers as follows:

5...8,12,15...20,23,25...27   or
504223...504227,504230,504233...504236.

To plot from a file, use

```
    plot [list]   ("list" being one of the above options)
```

Example:

```
   r plt51
   COPY, MTS, PLOT, STOP?
   copy all
   WHERE IS A PLOT TAPE?
   t904
   PLOTS FROM THU 1400 TAPE; TAPE WRITTEN 13:00:26 04-26-79.
   1=BLAK/4
   ok
   1=BLAK/4    2=BLUE/4    3=RED/4
   ok
   PLOTTING PLOT 004; NOW WHAT DO YOU WANT?
   bac 5
   1=BLAK/4
   ok
   PLOTTING COMPLETE.
   COPY, MTS, PLOT, SIG, STOP?
   plot 2
   PLOTS FROM THU 1400 TAPE; TAPE WRITTEN 13:00:26 04-26-79.
   1=BLAK/4    2=BLUE/4    3=RED/4
   ok
   COPY, MTS, PLOT, SIG, STOP?
   stop
```

MESSAGE TO USER

If it is desirable to print a message to the user at some point, e.g., when lines are being over drawn and the pen is clogging, you can use the MSG facility.

To use this, simply GOOSE the supervisor number of the plotter, type MSG followed by a space and the message.

Example:

```
     MSG  There is too much up and down motion of pen.  Penpoint
is overheating.
```

Note: The message must be entered on one line.

MULTI-PEN PLOTS (in PLT)

Multi-Pen Plots are done first, beginning with the shorter ones. Next come the one pen plots beginning with the shortest.

Just before pens are requested a tape I.D. message will appear, e.g.,

```
    PLOTS FROM MON. 1000 TAPE; TAPE WRITTEN 10:12.47 12-19-77
```

The number after the slash refers to the pen size; 4=narrow 6=medium(default) and 8=wide.

```
1=BLAK/6 3=RED/4 2=BLUE/8
```

ok If "ok" is given, PLT will begin plotting that plot.

If "no" is given, PLT will go on to next plot

1=BLAK (system asks as single pen plots begin)

ok

PEN POSITION CHANGING

Occasionally one may need to change the position of the pens, for example if one of the pen holders isn't working properly. In order to do this, one must wait for the plotting system to print out-

```
      1=black  2=blue  3=red  4=green
```

At this point, for example, you could type-

```
      3,2,4,1
```

which would expect the black pen in position 3, the blue in position 2, the red in position 4 and the green in position 1.

If you typed 4,3,2,1 the plotting system would expect green in position 1, red in postion 2, blue in position 3 and black in position 4. What counts is what number you put in which position. For instance when we said (in the above example) 3,2,4,1 the first postion tells where the black pen goes, in this case in position 3. The second position tells where the blue pen goes, in this case, position 2.

When all of the plots are done, the system will comment

PLOTTING COMPLETE

PLOTTING SYSTEM

```
EXECUTION TERMINATED

$sig $
```

REPLOTTING

After the plot tape has been dumped into a file, plots can be replotted without remounting the tape as long as these plots were included in the latest tape-to-file operation.

After the operator has typed $run plt

Operator can type:

plot all (replots everything in the file)

plot 014 (plots block 14, only)

plot 007...009 (plots range of blocks, 7-9)

plot s-8 (replots all blocks under s-8 given)

ATTENTION INTERRUPTS

NOTE: After PLT has been interrupted and is waiting for a response on the operators console, do not enter a command until the pen carriage has stopped with the plotter in automatic.

Plotting can be interrupted by 'GOOSING' the supervisor number of the plot job. This will result in the printing of the comment 'PLOTTING PLOT XXX; NOW WHAT DO YOU WANT?'

The following are options which can be used at this time:

pri n (n being the time in seconds that the plot routines will wait after last usage of the pen before priming it - default = 2 min.)

bac (start the current plot over)

bac n (start at plot N, continue to previously specified end)

con (continue, stop at end of current plot)

con n (continue, stop at end of plot N)

mts (call MTS) NOTE: PLT is still loaded at this point and can be RESTarted sta (go back to "what is to be plotted?")

End-of-file (go back to whatever you were doing at time of interrupt)

ret n (Retraces N number of records)

NOTE:
If the comment 'HELP! I/O TROUBLE ON PLT1' occurs, see if someone is using the plotter; if not type 'RES' If, after using RES, the I/O trouble still occurs, signoff and start again.

PENS

A description of cleaning and filling pens is found on p.11 of the first (OPERATIONS MANUAL) of two manuals which were supplied by the Vendor. When it is time to refill a pen, simply put the plotter in manual (while it is running) and remove the pen from the pen carriage. To do this, grasp the pen by the top of the pen holder and turn it counter clock wise 1/8 turn and lift it out.

See p.11 of the Vendor-supplied Operators' Manual for pen maintenance and filling.

See p.5 of same manual for paper changing.

OUT-OF-PAPER SWITCH - MODEL 936

A photocell has been placed at the bottom, center of the rear vacuum column of the CalComp 936 plotter. The plotter now automatically stops if it runs out of paper as long as the overhead lights are on. If these lights are off, the plotter will behave as it did without the switch.

Operating Instructions...

IMPORTANT: Before doing anything else, put plotter in manual since the manual/auto switch is over-ridden by the photocell

1) Be sure that the room lights over the 936 plotter are on

2) Be sure that the plotter is in "manual"

3) Interrupt the plotter

4) Let concentrator run out plot buffer by putting plotter in

auto covering the photo cell with something, e.g., a card or your hand.

5) Install new roll of paper

6) Set up the plotter to begin a plot

7) Do a BAC (check to make sure that only the last plot was lost; otherwise backup further)

<u>NON-QUEUED PLOTS USING SYS.:PLOT36 or SYS.:PLOT51</u>

The program SYS.:PLOT36 uses the CalComp 936 and SYS.:PLOT51, the CalComp 1051.

Before a plotfile can be routed to the plotter, it must be translated into plotter commands by a program such as SYS.:PLOT36 or SYS.:PLOT51. *CCQUEUE is a program users usually use to queue their plots for scheduled runs. *Ccqueue does no translation.

If one wishes to plot a file which has not yet been modified to be plotted directly, or does not want the plot to be queued, then the program, SYS.:PLOT36 or (PLOT51) may be used.

<u>Note</u>: For net work attached plotters, >pltn doesn't work.

Instead, do the following-

```
$mou mnet *plot* d=hXxCcb7

   where xx is 00 for plt2 and 01 for plt3
```

Run sys.:plot51 and enter *plot*Bin at "where to".

*plot*Bin for "WHERE TO?".

```
$run sys.:plot36 (or 51)  (user types)

WHERE FROM:
(program asks)

w035:asd
(user types as example of
plotfile to be used)

WHERE TO?
(program asks)

*plot*Bin
(user types, if the user wants to route the plotfile directly to
```

the plotter) -

```
SCALE FACTOR:
(user enters scale factor)

  0.25   (example)

PRIMIMG TIME(SEC.):
(user enters time interval in seconds that is wanted before  pen
priming occurs after last use of pen)  (or user presses ENTER)

HEADER(CCIDDATETIMERECEIPT):
(user  enters  information  shown  in the parenthesis that is to
appear on the plot)  (or user presses ENTER)

NUBS?
(if the user wants the plot shipped to NUBS, "yes" is entered)

PENS: (COLORS,SIZES,NEW OR NULL)
(user can press ENTER at this point if the default pen setup  is
wanted - "new" causes the plot routines to print:

      1=  after which the user can type color, size and type

      e.g.  1=blak,med,liq
```

Following this, the routines type 2=, 3=, and 4= until all pen positions have been attended to. If a null line is entered after an n=, than the default setup for that position is assumed.

(plot is plotted) then;

```
WHERE FROM?  (system asks)

$endfile          (user types to terminate program)
```

If the file to be plotted has been routed to another file for later plotting, the "where to" file can be copied to the plotter.

```
$copy file >plt1@bin             (user types)
```

The programs, SYS.:PLOT36 and SYS.:PLOT51, formerly W030:PLOTX and W030:PLOT1X, have been modified (Nov. 1980) to handle pen changes in a different manner than before.

PLOTTER TESTING PROGRAMS

SYS.:PLTEST36 AND PLTEST51

PLTEST36 is for the CalComp Model 936 and PLTEST51 for the 1051.

To run,

```
$R SYS.:PLTEST36 or SYS.:PLTEST51
```

If you get an undefined symbol message from the loader, type IGNORE and press "enter".

The program will ask

```
PLOTTER NAME:
```

respond with PLTn (e.g., PLT1).

The program will ask

```
WHICH TEST?
```

The available tests are currently

```
     CHARS - draws characters
     LINES - draws lines in various directions
     GRID - draws a grid
     DOTS - draws dots
     SIN - draws a sin curve
     POLY - draws polygons and circles
     BLACKEN - draw lines close together
     DASH - draw dashes
     LINE - draw without picking up the pen
     NOTE - print something in .2 inch letters
     STAR - draw lines intersecting in a single point
     SYMTEST - draw characters in decreasing sizes
```

The details of each plotter test are described below. For each test, the program will plot for a bit, then plot the time, and then repeat that until you hit the attention. After an attention, the plotter will plot a bit more (ie, it won't stop immediately), then the program will ask WHICH TEST again.

SPECIFIC TEST DESCRIPTIONS

For each test there are several variable parameters. Values are suggested for each to get you started; feel free to try other values as well.

Whenever the program requests two or more numbers on the same line, enter them separated by commas. Most of the test ask which pen number to use:

PEN:

Respond with a number between 1 and 4 inclusive.

SIN: plots a sine curve along the drum. Program will ask

PERIOD, AMPLITUDE, INCREMENT(IN.):

Enter 3 floating point numbers separated by commas.

The increment is the smoothness of the curve; a smaller number is a smoother curve. For starters, try 2.,1.,.01. Finally, the program will ask for a pen number.

DOTS: plots dots ad infinitum. Program will ask

Y #DOTS, DISTANCE BETWEEN

Respond with 2 numbers separated by commas (the second number is a floating point number of inches). The program will then ask

X #DOTS, DISTANCE BETWEEN

Respond similarily. Finally, it will ask

Enter the number of a pen holder to use, from 1 through 4.

You might try 100,.125 for y and 10,.125 for x.

LINES: This is a check of line quality in various directions.

The program will ask

LINE LENGTH:

Enter a number in inches,<=16. All lines drawn will be this length. It will then ask

ANGLE INCREMENT (DEGREES):

Enter a number >=0 but <=45 for the degrees between successive lines. The program will then ask for a pen number.

The program draws

Try a length of 10. and an angle of 5 to start.

GRID: draws a rectangular grid, lifting the pen very seldom. It

tests long lines and drawing with the pen down for long periods.

The program will ask

Y-LENGTH, INTERVAL:

Enter the length of lines in the y direction (along the drum) and the distance between lines, separated by commas. The program will then ask

X-LENGTH, INTERVAL:

This is the same question, but for the x-direction. The program will then ask for a pen number.

Try 20,.125 for y and 4,.125 for x to start.

CHARS: draws character sets. The program will ask

ALPHABET:

Enter the name of any alphabet described in Volume 11. SANSERIF.1 is a good choice. (Be sure to use uppercase). It will then ask

LETTERING SIZE:

This is the size of the characters, in inches. Use a number <=.25 or so; .15 or .2 is a good choice. The program will then ask for a pen number, as for the other tests.

POLY: draws sets of concentric polygons. The program will ask

LARGEST & SMALLEST RADII, INCREMENT:

Enter the radii of the largest and smallest polygons in each set of concentric polygons and the increment in radius between successive ones. (For example, if 2.,1.,1. is entered, each set will have a polygon with a radius of 1 inch inside a 2 inch one.) The program will then ask

SIDES (<0 FOR PCIRC):

Enter the number of sides in each polygon. A large number of sides will cause the polygons to look like circles (this is a reasonable test). If you enter a number less than 0, a circle will be drawn using very fine steps. Finally, the program will ask which pen to use.

You might try 3.,1.,.5 for the radii and increment, and 5 for the number of sides (pentagons), or perhaps 200 (near circles).

For each repetition, the program will plot as many sets of polygons or circles across the drum as will fit.

STAR: draws a group of lines which intersect in a single point.

The program will ask

# LINES:

Enter the number of lines which are to be drawn through one point. The program will then ask which pen to use.

This program can be used to test plotting paper.

For each repetition, the program will plot as many 'stars' as will fit across the drum.

NOTE: prints a character string along the drum. The program

will ask

MSG:

Enter the text to be printed. The program will then ask for the pen number, and will print the message.

BLACKEN: draws parallel lines which may overlap. It can be used to draw over the same line many times. The program will ask

LENGTH OF LINE (INCHES)

All lines drawn will be this length. It will then ask

SPACING OF LINES (INCHES)

Enter the distance between lines. Usually you will want the lines to overlap, so enter a small number (e.g., .01). Enter 0 to have the same line redrawn over and over. Then the program will ask

# LINES TO A BUNCH

Enter the number of lines to be drawn. If the spacing is 0, this will be the number of times the same line is redrawn. Finally, the program will ask for a pen number.

For each repetition, the program will draw the specified number of lines. It will then draw the time message and begin a new group of lines.

This can be used as a test of fiber pick-up for various inks, pens, and papers.

DASH: draws dashes. The program will ask

DASH, INTER-DASH LENGTHS:

Enter the length of dash to be used, and the distance between consecutive dashes in inches. The program will then ask

TIMES:

Enter the number of dashes to be drawn in a repetition. Numbers in the hundreds or thousands are quite reasonable here. Finally, the program will ask for a pen number.

This can be used as a test of ink globbing as the pen is lowered or as a test of pen block overheating. For these purposes, use short dashes and short interdash lengths, e.g., .1 and .05. Several thousand to a repetition is also good.

For each repition, the program will draw the specified number of dashes going back and forth along the drum. It will then draw the time message and start a new set of dashes.

LINE: draws a continuous line back and forth along the drum. The line does not cross itself. The program will ask

NUMBER OF TIMES:

Enter the number of times the program is to draw back and forth (60 inches total) in each repetition. The pen will not be raised within a repetition. The program will then ask for a pen number.

The purpose of this program is to test ink and pen points. Usually you will want a large number of repetitions. For example, 50 times would produce a 250 foot line before the pen is raised to draw the time message.

SYMTEST: draws a character string several times in decreasing sizes. Enter the character string to be drawn when the program prints

STRING

The program will next print

LARGEST, SMALLEST SIZES

Enter the largest and smallest sizes, in inches, at which the string is to be drawn. Then enter the difference in size (inches) between successive strings. The program will then ask for a pen number.

This program is useful for checking "jittery" corner turning on the Calcomp 936. String sizes between about .3 inches and .05 inches are good for this kind of testing. Try an increment around .05.

PM2 - AN STC ERROR ANALIZING PROGRAM

(((((((((NOT PRESENTLY USED AT U. OF MICHIGAN)))))))))))

Running PM2 in MTS

Example "run":

```
     $emp pm.sep0683 ok            (Empties output file)

     $rename pm.sep0683 pm.sep0983 ok
                                   (Renames   output   file   to
                                   comply with current date.)

     set macros=on

     >set var maclib(1)="etc.:pm2.mac"

     pm2daly print=pm.sep0983      (Runs program)
```

PM2 is a program developed by STC to maintain a history of device errors on their equipment and print reports related to this. It is run in MTS by invoking one of several command macros, which must first be made available to MTS by executing the following two commands:

```
          SET MACROS=ON

          >SET VAR MACLIB(1)="ETC.:PM2.MAC"
```

After executing the above commands, the usual manner in which we run PM2 can be done.

Usually, PM2 output is sent to a file where the limits of its three sections can be determined via the editor so that they can be printed separately if desired. The three sections are Tape Perfomance Report, Disk Performance Report & Tape Volume Report. The name of the output file is changed daily, i.e., if PM2 is actually run every day.

```
  $emp pm.may0683 ok

  $rename pm.may0683 pm.may0983    (Changes  the  name  of   the
                                   output    file    from    last
                                   Friday's date (05/06/83), the
                                   last time  PM2  was  run,  to
                                   Monday's  date,  the  current
                                   PM2 run.)

  pm2daly print=pm.may0983         (runs the program)
```

Note the ">" on the "SET" command. It is a flag indicating that this is a command for the macro processor and is required on this command.

Invoking the two command macros makes six command macros available for the duration of the current signon. These macros may be used exactly as if they were MTS commands. The prototype of each of them is given below, followed by a description of the parameters they take. Note that many of the parameters are for use only when debugging PM2, and are included here for completeness only. Only the parameters given on the first line of each prototype are needed for normal use of PM2.

To create or update a PM2 history file:

```
     PM2HMAN      CFIGIN=config      SAVEIN=save      PAR=params
     PRINT=printfdname
               VSSCMDS=cmdfdname VSS=vssobj PM2LIB=pm2obj DEBUG
               HISTFILE=pm2history
```

To dump a PM2 history file

```
     PM2HDMP PAR=params PRINT=printfdname
               VSSCMDS=cmdfdname VSS=vssobj PM2LIB=pm2obj DEBUG
               HISTFILE=pm2history
```

To update the tape volume error history

```
     PM2TVDL PAR=params PRINT=printfdname TVOLCMDS=volfdname
               VSSCMDS=cmdfdname VSS=vssobj PM2LIB=pm2obj DEBUG
               HISTFILE=pm2history TAPEHIST=tapehistory
```

To run the daily PM2 run

```
     PM2DALY PAR=params PRINT=printfdname TVOLCMDS=volfdname
               VSSCMDS=cmdfdname VSS=vssobj PM2LIB=pm2obj DEBUG
               HISTFILE=pm2history TAPEHIST=tapehistory
               CMVTEREP=cnvterepobj CNVTCTL=ctlfdname
               LOGREC=historyfile
```

To pop the stack of history files

```
     PM2POP HISTORY TAPEVOL
               HISTFILE=pm2history TAPEHIST=tapehistory
```

To push the stack of history files

```
     PM2PUSH HISTORY TAPEVOL
               HISTFILE=pm2history TAPEHIST=tapehistory
```

The parameters required for normal use of PM2 are as follows:

PAR=params

"params" is any set of parameters from chapter 5 of the STC PM2 manual that is appropriate for the PM2 program being run. The defaults are as given in the STC manual, except that PM2DALY defaults P=N if and only if not explicit "PAR=param" parameter is given. Also, PM2STAT defaults to PAR='P=N,TYPE=OT', but this is not given much use since PM2STAT requires a DATE= or DTTM=parameter which must be given on the PM2STAT command If more than one parameter is given, they must be separated by commas and enclosed in quotes.

PRINT=printfdname

The printed output from PM2 will be directed to "printfdname". If this is a file, the output will be automatically appended to the end of the file; you need not specify (*L+1) on the fdname. The default is "PRINT=-PRINT".

TVOLCMDS=volfdname

"Volfdname" is an fdname from which will be read tape volume control functions as described starting on page 5-16 of the PM2 manual. Normally, "volfdname" will be a file. The default is "TVOLCMDS=*DUMMY*".

CFIGIN=config

"Config" must be an fdname from which will be read configuration cards as described in chapter 3 of the PM2 manual. It is normally a file. The default is "CFIGIN=*DUMMY*".

SAVEIN=save

"save" must be an fdname from which will be read save cards as described on pages 3-16 and 3-17 of the PM2 manual. It also is normally a file. The default is "SAVEIN=*DUMMY*".

HISTORY and TAPEVOL

These parameters indicate to PM2PUSH and PM2POP which of the two sets of history files to push or pop. Two consecutive pushes or pops will have no effect. All PM2 macros except PM2HDMP automatically push the appropriate history files upon successful completion. The default is to push or pop neither set of files.

All of the following paramaters are useful only when debugging PM2, they should be ignored for normal use of it and are included here only for completeness.

VSSCMDS=cmdfdname

"Cmdfdname" must be an fdname from which VSS commands will be read immediately before PM2 is invoked. It may be *MSOURCE*.

VSS=vssobj

"Vssobj" is the name of the file containing the object for the version of VSS to be used to run PM2.

PM2LIB=pm2obj

"Pm2obj" is the name of the file containing the object library for PM2 itself. It must be a VSS PDS as produced by the MVS link editor.

CNVTEREP=cnvterepobj

"Cnvterepobj" must be the name of a file containing the object for the CNVTEREP program that converts MTS unit check log records into the format accepted by PM2 (and EREP).

CNVTCTL=ctlfdname

"Ctlfdname" is the name of the file containing control information for CNVTEREP.

LOGREC=historyfile

"Historyfile" is the name of the file containing output from CNVTEREP. If this parameter is given, CNVTEREP will not be run.

DEBUG

If this parameter is given, VSS (and hence PM2) will be run under control of SDS.

HISTFILE=pm2history

"Pm2history" is the "root" of the name of the stack of history files for PM2. It is suffixed with ".N" for the new history file created by this run, by ".O" for the first backup history file, and by ".VO" for the second backup history file, and by nothing for the current history file.

TAPEHIST=tapehistory

"Tapehistory" is the "root" of the name of the stack of tape history files. It is suffixed in the same way as pm2history described above.

Examples:

In all these examples, the two set commands need to be given only once per signon. They will be shown in the first example only to emphasize this.

To make the daily PM2 run and print the output, use the following commands from user id SYS.

(Another example of starting up PM2 follows.)

```
          SET MACROS=ON
          >SET VAR MACLIB(1)="ETC.:PM2.MAC"
          PM2DALY PRINT=*PRINT*
```

Alternatively, to get the output "2-up" on the page printer use

```
          PM2DALY

          RUN *PAGEPR SCARDS=-PRINT PAR=TWOUP
```

To give an example of how parameters may be passed to PM2, suppose you want to retain entries on the tape volume history for 100 days and ignore printer errors on the daily PM2 run. You could use

```
          PM2DALY PRINT=*PRINT* PAR='P=N,RET=100'
```

To delete volumes from the tape volume history file, put the appropriate "DVOL" commands (as described on page 5-18 of the PM2 manual) in the file "-dvol" and execute the commands

```
          PM2TVDL TVOLCMDS=-DVOL
```

This will leave its printed output (which will consist of the normal tape volume history report) in -PRINT.

To get the monthly summary report for August, 1983, you could use the command

```
           PM2STAT PAR='P=N,TYPE=OT,DATE=83213-83243'
```

This command will leave the monthly report in the file -PRINT. It includes the P=N PARAMETER TELLING PM2 to ignore print errors. The DATE= parameter gives the first and last day in August, 1983 in YYDDD format, where DDD is the day number in the year.

Special Rates for Holidays - How to Set

I. Before the Holiday Period-

1. Change the times that minimum priority jobs are released by the RM by changing the control file, SYS:AUTOHASP.CTL.

```
$Signon SYS.
$Duplicate AUTOHASP.CTL AUTOHASP<CTL
$Edit AUTOHASP.CTL
   --  make  necessary  changes  to the times and days
using the editor.
```

Below is the contents of the file sys.:autohasp.ctl. Mentioning a specific day such as Monday takes priority over "weekday". The first line below (SUNDAY 00:00 6:59 ON ON ON) means that Low, Def, and Min jobs can run on Sunday from midnight until 6:59. The next line shows that on Sunday from 7:00 to 17:59, Low and Deferred jobs can run but Minimum priority jobs can't. Don't change the line numbers in this file. The line Beginning with "Day" should be line "0" and the next line, 0.x.

```
  Day      Begin End Low Def Min

SUNDAY    00:00 06:59 ON  ON  ON
SUNDAY    07:00 17:59 ON  ON  OFF
SUNDAY    18:00 23:59 ON  ON  ON
MONDAY    00:00 06:59 ON  ON  ON
FRIDAY    18:00 21:59 ON  ON  OFF
FRIDAY    22:00 23:59 ON  ON  ON
WEEKDAY   00:00 01:59 ON  ON  OFF
WEEKDAY   02:00 06:59 ON  ON  ON
WEEKDAY   07:00 17:59 OFF OFF OFF
WEEKDAY   18:00 21:59 ON  OFF OFF
WEEKDAY   22:00 23:59 ON  ON  OFF
SATURDAY  00:00 06:59 ON  ON  ON
SATURDAY  07:00 17:59 ON  ON  OFF
SATURDAY  18:00 23:59 ON  ON  ON
```

2. Change the file of initial reload commands to include setting the special holiday rates. This is done so that if the machine crashes during the holiday and is reloaded, the rates will still be set to minimum.

```
$Signon INIT
$Duplicate *INITCMD INITCMD<
$Edit *INITCMD
```

The lines (below) to be inserted into file initcmd should be placed after the line (probably line 40) which contains "mod csdw2 12".

```
: insert 40 (or whichever line is appropriate)
```

```
                 ("insert 40" inserts a line (or lines) after
                 line 40 when in edit mode).  The three lines
                 which follow should be inserted.
          ?  MOD btchprio 1
          ?  MOD termprio 1
          ?  O $Release minimum
          ?  (Press "Enter")
```

Note: Btchprio and Termprio values

0 indicates that rates are based on the time of day

1 indicates minimum rates

2 indicates deferred rates

3 indicates low rates

4 indicates normal rates

3. When the holiday period begins, the operations staff should specifically set minimum rates before they leave. While operators are on duty, these two priorities should be set to "0".

   Using the MTS operators' consoles, enter:

   ```
   MOD BTCHPRIO 1
   MOD TERMPRIO 1
   $RELEASE MINIMUM
   ```

II. After the Holiday is Over:

1. When the holiday period ends and regular rate periods are to be back in effect, operators should reset them by entering on the MTS operators console:

   ```
   /MOD BTCHPRIO 0

   /MOD TERMPRIO 0

   $hold deferred

   $hold minimum

   $hold low
   ```

2. Reset the HASP control file:

   ```
   $Signon SYS
   $Duplicate AUTOHASP<CTL AUTOHASP.CTL ok
   $Destroy AUTOHASP<CTL ok
   ```

3. Reset the initial reload commands file:

```
$Signon INIT
$Duplicate INITCMD< *INITCMD ok
$Destroy INITCMD< ok
```

Modifying BTCH and TERM when operators are on duty and when they leave. Changing AUTOHASP and INITCMD should be changed once at the beginnig of the holidays and once again after normal hours begin at the end of the holidays.

SCAN, CMD & STAT Records

W030:SCANX

Purpose: SCAN searches cmdstat and monthly statistics tapes for requested CMD and STAT records. A search may be done by the CCID, job number, date-time, CMD name, etc. Subroutines may be appended to do specialized searching, with SCAN providing reading, writing, and tape file selection utilities.

Location: The object module for SCAN is in the file W030:SCANX.

Philosophy: This program has two classes of users: those who use it almost daily, e.g., the Business Office and those who make occasional use of it, e.g., the programming and operations staff. The syntax and verbosity are designed mostly for the heavy users who must live with it the most. For the benefit of occasional users, however, it will print the syntax of a command once it figures out which one you're trying to use. Also, since the program expects to be searching through large volumes of data, some things are sacrificed to run-time efficiency (for instance user-supplied search programs use a non-standard calling sequence).

Description: Generally, to request a search, the user first specifies the type (stat,cmds), then a list of the search criteria. SCAN carries out the search and prints the results. The results can be copied or printed, in the whole or in part, on any desired FDNAME. The command descriptions follow.

STAT [[fdname|NOLIST]]

This command is used to request a search for statistics records. After the search has been completed, the statistics records found will be listed (in "LISTSTAT" format) on "fdname". "fdname" defaults to *SINK*. If NOLIST is specified, this listing is not produced. SCAN will prompt for a list of search requests, one per line. Each request has the form

CCID fromtime [[TO] totime]

"CCID" is the userid for which statistics records are requested. "fromtime" and "totime" specify the range of signon times of statistics records. SCAN will add a fudge factor to "totime" (currently 12 hours), and scan through signoff times up to that time. "fromtime" and "twotime" are currently in UBCDATE (JVCT) format--but this will change when a replacement for UBCDATE is written. "TO" is required only when the two consecutive times would be ambiguous. A null line or end-of-file terminates the list. After the list has been terminated, SCAN will search the monthly statistics tapes and *STATISTICS for the requested records. When the search has been completed, the records found will be printed in LISTSTAT format. The LIST and COPY commands may be used to manipulate these records.

LIST [STAT|CMDS] [fdname]

LIST prints statistics or cmdstat records which have previously been found in a "readable" format. "fdname" is optional; the defualt is *SINK*. SCAN will prompt for items to be listed, one per line. The possible items are:

```
ALL
lineno
lineno lineno
U ccid
P prjn
J jobn
```

"ALL" specifies that all statistics/cmdstat records are to be printed (not just those found in the most recent search). "lineno" is the sequential number of a record (for statiscs records, this is printed at the right side of the output in a LIST or STAT command). If a single "lineno" is entered, just that record will be printed; two "lineno"s is taken to mean a range. "ccid" is a signonid; for this type, all records for that signon id are printed. "prjn" is a project number; all records for signon ids in that project are printed. "jobn" specifies a task number, and may only be used with cmdstat records. All cmdstat records for "jobn" are listed. SCAN prints out the listing for one item before requesting another. A null line or end-of-file will cause SCAN to request a new command. A new list command must be entered to change the fdname, or whether statistics or cmdstat records are being printed.

COPY [STAT|CMDS] [fdname]

The syntax of this command is the same as for LIST. COPY copies records to "fdname" in their internal form, as read from the tape. All records are in the "new" (FEB. 1982) statistics records format.

MTS

This command causes a return to MTS; SCAN may be $RESTarted.

SET STATSCAN=[fdname|DEFAULT]

This command changes the location where the STA command searches for statistics records. STATSCAN=fdname will cause only fdname to be searched for statistics records (the monthly tapes and *STATISTICS will not be searched).

SCAN trusts that you will provide statistics records; it is likely to interrupt you if given bad records. NOTE: To have the STA command scan *STATISTICS only, enter SET STATSCAN=*statistics(3) since lines 1 and 2 do not contain statistics records. Also, note that *STATISTICS will be locked while the STA command is executing, so be judicious during periods of heavy load. STATSCAN=DEFAULT causes the STA command to resume searching in the normal places.

LISTSTAT-Type Output Codes and Meanings

JOB #:	SUPERVISOR JOB NUMBER
ID:	USER ID
PROJ #:	PROJECT ACCOUNT NUMBER
TIME:	TIME USER SIGNED ON
DATE:	DATE USER SIGNED ON
TIME ZONE:	
EL TIME:	ELAPSED TIME (SECONDS)
CPU TIME:	CPU TIME USED (SECONDS)
VMI CPU:	STORAGE INTEGRAL (VMI) WITH CPU (PG-SEC)
VMI WAIT:	STORAGE INTEGRAL (VMI) WITHOUT CPU (PG-SEC)
LAST SIGNON:	TIME AND DATE OF LAST SIGNON
TIME ZONE:	
LINE:	LINE NUMBER IN "*STATISTICS" FILE
RATE:	RATE NUMBER
$COST:	TOTAL COST
$FILES:	FILE CHARGE
C DSK	CURRENT DISK SPACE
LINES:	NUMBER OF LINES PRINTED (LINE PRINTERS)
PAGES:	NUMBER OF PAGES PRINTED (LINE PRINTERS)
CDS P:	NUMBER OF CARDS PUNCHED
CDS R:	NUMBER OF CARDS READ

FD:	NUMBER OF DISK FILES
CRE:	NUMBER OF FILES CREATED
DES:	NUMBER OF FILES DESTROYED
MERIT:	MERIT TIME (SECONDS)
DSK USED:	CUMULATIVE DISK SPACE SINCE LAST SIGNOFF (PG-MIN)
DSK OPS:	DISK OPERATIONS
PAGE-INS:	NUMBER OF VM PAGE-INS PAGED IN BY PDP
ANSBACK:	TERMINAL ANSWERBACK CODE
PPRSHT:	PAGE PRINTER SHEETS
PPR LNS:	PAGE PRINTER LINES
PPRPGS:	PAGE PRINTER PAGES
PPRIM:	PAGE PRINTER IMAGES
PLT TM:	PLOT TIME CHARGED
PLT PAP:	INCHES OF PAPER USED (LINEAR)
MNT:	TAPE MOUNTS
FMN:	FLOPPY DISK (DISKETTE) MOUNTS
RMN:	PAPER TAPE READER MOUNTS
PMN:	PAPER TAPE PUNCH MOUNTS
T DRV:	MAG. TAPE DRIVE TIME
F DRV:	FLOPPY DISK (DISKETTE) DRIVE TIME
R DRV:	PAPER TAPE READER TIME
P DRV:	PAPER TAPE PUNCH TIME
PTPP:	PAPER TAPE PUNCHED

Using the Statistics Utilities

SCAN provides the basic utility routines to search, copy, and print statistics records. Subroutines may be written which act as command processors and which may call the utility routines.

Before a command processor can be used, it must be loaded via the LOAD command. This establishes the name of the command handled by this command processor and the entry point to be called to do the processing. That entry point is called with the following parameters:

LINES	2F line numbers
STRING	nC character string from command line terminated by X'4000'
TV	nA transfer vector
STAT	routine to call to get statistics records
CMD	routine to call to get cmdstat records
LSTAT	routine to list statistics records
LCMD	routine to list type 1 cmdstat records
LESPAN	routine to do (SPAN(' ') | NULL)
LENUMB	routine to CVB a number
LEBREA	routine to do BREAK(' ')
STATFD	fdub for temporary file containing statistics records

See SCANTR in the file SCANDS for a transfer vector dsect.

The command line read by SCAN is

```
cmdname [line#1,line#2] [string]
```

"line#1" and "line#2" are a pair of line numbers. They are passes in the parameter "lines" in internal line number form, with "line#1" first. If either value is omitted in the command line, a zero is used.

"string" is an optional, arbitrary character string on the command line which is passed in the parameter "STRING" to be parsed by the command processor. SCAN does not guarantee that this string will exist after a call to any utility routine other than LESPAN, LEBREA or LENUMB.

STAT searches the monthly statistics tapes for records described in an "itemlist" given by its caller. It is a vector of items of identical length. Each item has the format

```
 ---- ---- -------- -------- -----
|ccid|stsw| stfrom |  stto  | whatever . . .
 ---- ---- -------- -------- -----
 0    4    8        10       18   (hex)
```

"ccid" is a signon id. If the high-order bit of "stsw" is on, a record without that signon id (in STAUI) cannot satisfy this item. If the high-order bit is off, STAT does not check the signon id. "stfrom" and "stto" specify a range of times in

Julian microseconds. Only records with signon times within that range can be accepted. "whatever" is ignored by STAT; it may be any number of fullwords in length, or omitted, but every item in the itemlist must have the same length. STAT sorts the itemlist (in place) by "stfrom" It scans statistics tapes beginnig with "formtime" of the oldest item, and continuing until 12 hours after the last "totime". A statistics record is matched (accepted) if it satisfies any item (thus the items are, in affect, "OR"ed). The caller of STAT may provide a routine to inspect every accepted record and determine a final acceptance.

The parameters for STAT are:

ITEM	A beginning of item list
EITEM	A end of item list <not the addressing of the last item)
LENGTH	F length of an item
ACCEPTOR	A routine to call to detrmine final acceptance of a statistics record. A(0) (not a 0 in the PAR list) if routine is not supplied.
LINES	2F on return, beginning and ending line numbers in -SSTAT of statistics records accepted.

The return codes from STAT:

0:	done
4:	no records found
8:	unable to complete

The routine whose address is in "acceptor" is called as follows:

gr1:	a (item matched by STAT record)
gr2:	a (stat record)
gre:	return address; return 0(gre) to reject the record, and 4(gre) to accept it
grf:	address called

Registers gr0, and gr3 through gr6 may be used. All other registers should have the same value on return as on call. The routine called must not use any of the routines whose addresses are in the TV except LESPAN and LENUMB. The routine should always return (STAT may have tapes mounted or files locked).

Control is not guaranteed to return to a command processor after an attention; the command processor may be re-entered from the top.

SCAN

(((((The SERCURITY CLERK SECTION IS PRESENTLY NON-APPLICABLE)))))

Security Clerk--Computing Center

Purpose and Duties

NOTE: At present, we do not employ the services of Security ClerkS, but when we do, this is standard operating procedure for them.

Purpose:

The purpose of the "Security Clerk" position is that of having an employee available to "Babysit" the Computing Center during the time the Center is not open to the public, but the Computer and the Air Conditioning units remain powered on. The presence of the Security Clerk is to insure that the integrity of the Computing Center building and all of the equipment inside of the building, espcially that of the the Computer and Air Conditioning units will be maintained.

Duties:

The Security Clerk will be stationed in the second floor machine room. Every 1/2 hour, the building should be inpected by the clerk. The areas to be inspected include the the three machine rooms, the two office areas on the second and third floors, the user work area on the first floor and the Electronics shop.

At 2AM and at 6AM, the Clerk should inspect the Boiler room (1012-M) and the Generator room (1010-E). These two rooms are located on the first floor at the north end of the user work area. Keys to all of the areas to be inspected are on a large ring at the west end of the second floor machine room hanging just below the two black recording phones.

If any problems arise with which the Security Clerk is unable to contend, the Security office (phone 31131) and/or the Fire and Police departments (phone 9-911) should be called immediately.

If any damage has been done to the Computing Center building or to the equipment inside of it, one of the following people should be called:

Mr. L. G. Mitchell (Asst. Director of Operations)

Dr. A. R. Emery (Deputy Director) 9-761-6960

Doors leading to the outside of the building should be checked on each tour to make sure that they are locked. These doors are locked by the use of an alan wrench which is on the key ring.

The Security Clerk is expected to answer the phone, that is the trunk lines 764-2121,22 and 23. The phone in the second floor machine room can be made to ring by turning the switch on the switching box "on". This box is located immediately to the left of the two black recording phones by the key ring.

Emergency Power-Off Measures

Emergency Power Off Measures June 1988

Power Off Buttons (Emergency) "pull" type.

At present, there are seven "pull" type Emergency Power Off (EPO) buttons located in the Computing Center building (1075 Beal). Two of these buttons are located in the NSF computer room and will only power the NSF computer equipment, off.

The other five will turn all computing equipment off, that is, the ES3090/300J equipment and its peripherals as well as the NSF computer and its peripherals.

The two EPO buttons for the NSF equipment are located in the NSF machine room where the IBM 4381s are housed.

Two other EPO buttons on the third floor are located near the two doors leading from the west half of the third floor to the NSF area. These can be seen as one enters the NSF area from the west end of the third floor.

The EPO button on the second floor is located just inside the east-facing double doors leading to the Communications area.

Of the three EPO buttons located on the first floor, two are located in the machine room, one inside and near the north door and the other inside and near the south east door. The third first floor EPO button is located in the Electrical room just inside the door to the left just below a shunt trip breaker.

The "Reset" button"

This button is a green push button and is located just above the shunt trip breaker mentioned in the paragraph above.

Note: Before using this reset button, be sure that all of the EPO buttons are pushed back in. Also there are some circuit breakers to be reset. Several of these are loacted in a large gray cabinet on the same wall as the reset switch but on the east side of the door. This cabinet is attached to the wall and is equiped with a door. Another breaker which must be reset is located in a breaker box just inside the first doorway leading into the UPS room. It is on a west wall. There are five vertically positioned breaker switches in this cabinet and the center one must be reset. Before using the green reset button, turn off the switches on the wall power panels which control power for the computing equipment. This will prevent possible problems when power is restored to everything all at once.

After the above has been completed, the green reset button above shunt-trip breaker may be reset.

SECURITY CLERK

SIGNON MESSAGE-(Using Messagesystem)

This method of issuing a Signon Message differs form the SIGNONM JOB program in that it is sent through the Messagesystem. Only certain privileged CCIDs can issue a SIGNONMESSAGE. The message may be placed in the system with a HOLD option that will keep it from appearing until a given time and date. Also, an expiration time and date may be specified so that the message will stop appearing after a certain time and/or date. The message may be constructed of more that one line but should be kept as short as possible for the users' sake. This type of sigonnmessage is usually issued from a terminal rather than form the operators' console.

Example:

```
 mess
 Set priv=on
 Send signonmessage to everyone holduntil='3:00 pm jun 8 1994'
```

Note: You can use a time (holduntil='3:00 pm') if it's the same day. You can use a date or both a time and date.

```
 EXPIRATION DATE/TIME: (system asks)
 6:30 pm oct 5 1983
```

At this point, you may receive a warning message stating that the expiration-hold date and time are close together.

```
  Text:  (system types)
 ?Next expected system shutdown is at 6:30 pm.
 ?The system should be up by 7pm.
 ?$endfile

  Send, edit, help, etc.?   (system asks)
```

At this point, you can type "post" if the message is ready to go or you can edit the message by typing "edit". When the message is finally ready, type-

```
 stop    (to get out of the editor)

 post    (to send the message)

 stop    (to get out of message-system)
```

NOTE: When you are in EDIT mode, the prefix is :. When you are in the message-system, the prefix is @.

For more information on the Messagesystem, refer to CCMEMOs #425 and #426. While in the Messagesystem and when in doubt, one can use the "help" command at certain points.

In the above example, the message would not begin to appear until 3pm and would cease to exist at 6:30pm, 10-05-83 even if the system was not reloaded.

More than one signonmessage may be issued since you may want different expiration dates or times; so, after you have "posted" one message, you may begin another by again issuing the command "send signonmessage to everyone" etc.

To check on all aspects of your signonmessage(s), you may-

```
     ret signonmessage

     NEXT, HISTORY, HELP, DELETE, REPLACE, ETC.   (system
     responds)

     display full
```

The system will respond with something like the example below-

```
     To:Everyone
     From:J.  Spratt
     Expires: 8:00 AM, Fri., Feb 18/83
     Current State: outgoing, incoming, seen
```

Then, the contents of the "signonmessage" are printed.

If more than one signonmessage is present, type

```
   next
```

This will print the number and contents of the next message. Note the message number and type-

```
   ret nnnnnn

   display full
```

Destroying a "Signonmessage"

If you want to get rid of a "Signonmessage for some reason before the expiration date/time, do the following:

```
mess

set priv=on

ret signonmessages
```

Note the message number, for example "976431"

```
destroy 976431    (destroys signonmessage)
```

After this, you will be promted for the next message if there is one. You could "destroy" the next one in the same way if you need to.

SUPERVISOR SENSE FLAG VALUES

(Add Values If Necessary)

FF

00=No Sense Requested (Program Error)

01=Good Sense Data

02=Unit Check On Sense Command

04=Device All Fouled Up, No Sense Possible

08=Interface Control Check

10=Channel Control Check

20=Channel Data Check

40=Protection Check

80=Channel Program Check

SUPERDUMPS

If the system is completely hung up and nothing can be done short of reloading, the Operator should force a Superdump.
Of course, before forcing a Superdump, a Systems Programmer should be found and consulted, if possible. The Systems Programmer <u>may</u> be able to remedy the situation without the necessity of a relaod.

If, however, a Superdump must be forced-

Select the "A" function of the "OP" menu of the macrocode.

The above prodedure will cause a Superdump tape request to appear on the Operators' console on the second floor.

System Reliability

Computing Center System Reliability Project
System Operation Reliability Committee

This message has three parts: (1) to explain what the CCSRP is (2) to exaplin the purpose of the SORC (3) to issue a statement on firefighter paging procedures.

(1) The CCSRP includes everyone designated as eligible to serve as a "Systems Group Firefighter".

(2) The SORC is a management level committee charged with monitoring overall system reliability.

(3) Firefighter beeper policy: At the last SORC meeting agreement was reached on the proper calling sequence and timing for contacting a firefighter (FF). When a system problem occurs which requires assistance from a FF...

```
     $Run Sys.:Fire for list of Firefighters currently
             **    Fire Fighter - 930-8086
             **    Systems Manager - 930-8976 (See next
     **Only F.F. allowed to "beep" manager. Don't call
         1) Firefighter has been called but can't fix p
         2) Firefighter hasn't responded within 20 minu
```

SYSTEM STATUS RECORDING

Information concerning the status of the hardware and software is availbale to users of MTS by listening to a taped message. The phone numbers for this message are 763-0420 and 763-0421.

The tapes are recorded and played on a Model 5001 Multi-Line Announcer manufactured by Automation Electronics Corporation.

To make a tape-

- Make certain that the OPERATE button is pushed in - this is the "Power" button.

- Snap a tape cartridge into the slot at the upper left of the recorder next to the microphone plug-in

- Push in the "Record" button and leave it in until the elapsed time equals the time printed on the tape cartridge, e.g., leave the "Record" button in for 14 seconds to erase a 14 second tape.

- Plug in the microphone (Some microphones have an ON/OFF switch which must be turned on.)

- The tape cartridges have time limits printed on the outer edge. Often, one will want to repeat a message several times on a tape with a pause in between. Of course, this depends on the physical length of the tape and the length of the message.

- Most of the buttons on this machine may be pushed in and will stay in that position until pressed again at which time they will spring back out.

- Press the red RECORD button and record your message.

- To check you message, leave the RECORD button out and press in the PLAY button. If you wish, you may re- record the message.

- When you are satisfied with the message, i.e., it's length, tone, content, etc., push in buttons 1 and 2 on the right, front of the machine as you face it. This will allow persons calling the two connected phone lines to access the recorded message.

Comments:

The two phone line buttons may be left pushed in while one is making a tape or while the PLAY button is pushed in.

When a user calls in, the recorder will disconnect the lines after ninety seconds uunless the LONG button is pushed in, in which case the recorder will terminate the call, assuming the user hasn't hung up, after three minutes.

At present, this machine is located in the Operations office on the second floor.

SYSTEMSTATUS (For Operators)

(Contributed by S. Gerstenberger)

A set of system status commands is available to provide information about the operation of the system and to allow an operator or systems programmer to modify certain parameters which control the operation of the system.

When a 3270 is being used as the operator's console, any of the system status commands listed below may be entered directly (the initial "/" is required in this case). If a 3270 is not being used, the JOBS job should be used for entering system status commands (the initial "/" is optional in this case).

```
     For Example:
     3270      /USERS
               /Q LCL

     non-3270  JOBS   (to start the JOBS job)
               USERS  (input to JOBS job)
               Q LCL  (input to JOBS job)
               .
               .
               .
               end-of-file  (to stop JOBS job)
```

Many of the system status commands listed below are also available to MTS users via the $SYSTEMSTATUS command (see MTS Volume 1). In addition, all of the commands listed below are available to "privileged" MTS systems programmers via $SYSTEMSTATUS.

Notice that certain parts of the commands are underlined to denote an abbreviation, e.g., /DISPLAY UAM could be stated as /D UAM.

```
/ALTER   name  value
/MODIFY  name  value
```

These commands may be used to modify a number of predefined variables that control system scheduling. "Name" must be one of the character identifiers described below and "value" is an unsigned decimal number. When changing bit values, zero is used to represent "off" and one is used to represent "on".

Example: /ALTER UAM 1

```
/ALTER   [R]location[+disp[+disp... ]]  value[[,]value...]
/MODIFY  [R]location[+disp[+disp... ]]  value[[,]value...]
/AREAL   [R]location[+disp[+disp... ]]  value[[,]value...]
/MREAL   [R]location[+disp[+disp... ]]  value[[,]value...]
```

These four commands may be used to alter real or virtual memory. The AREAL and MREAL commands always alter real memory, while the ALTER and MODIFY commands alter virtual memory unless the location to be altered is prefixed with an R. The replacement values are hex digits optionally separated by single commas.

```
Examples: /ALTER  600100      C1,C2,C3
          /ALTER  600100      C1C2C3
          /ALTER  600000+100  C1C2C3
          /MODIFY  R100  0F
```

/ATTN job#

Gives an attention interrupt to the specified job. This performs the same function as the GOOSE job program. This is useful when there are no job tables available.

/BLAST job#

Blasts the specified job. This performs the same function as the BLAST job program and as such should be used with great care so as not to to damage files. This is useful when there are no job tables available.

/DEVICE   [dev-name]

This command given without a device name generates a table giving the device name and the hex value of RDVBTS (from the device table) for all devices for which a device table is allocated. If a device name is specified, more detailed information for that device is printed. Both forms of the command continue to print information until interrupted by an attention interrupt (PA1 on a 3270) or the JOBS job is STOPped or GOOSEd.

Examples:

```
/DEVICE

 CB08 70 CB09 70 CB10 70 CB11 70 CB12 70 CB13 70 CB14 70  CB15 70
 CB16 70 LA06 70 LA07 70 LA08 70 LA09 70 LA10 70 LA11 70  LA12 70
 LA13 70 LA14 70 SA02 70 SA03 70 SA04 70 SA05 70 SA06 70  SA07 70
 SA08 70 SA09 70 SA10 70 SA11 70 T900 70


/DEVICE DS14

 Dev: 3270DS14 Job: 00458 Bits: 80000000 CSW: 10503048 0C000000
 Sns: 00 00 00000000 00000000 00000000 00000000 00000000 00000000

SYSTEMSTATUS
```

```
Atnx: 01503050  SNSX: 015035DB  PCIX: 00000000  ENDX: 00000000
2101B0/ 0C0E280E 80200000 00037D90 60008834 00000000 00000000 00000000
2101CC/ 00000000 0000000A 015035DB 10503048 0C000000 01503050 00000000
2101E8/ 00000000 101990F0 10000400 00000000 00000000 00000000 00000000
210204/ 00000000
```

where:

DEV is the device type followed by the device name.

The job number of the task using the device is 458.

BITS is the value in hex of four bytes of switches from the device table as follows:

```
     RDVTYP    X'80' on-->incorrect length is an error
    (bits 0-3) X'40' on-->24 byte sense
               X'20' on-->check for I/O Ops. that take too long
               X'10' on-->on block multiplexer channels

     RDVTYP    1-->direct access
    (bits 4-7) 2-->2250 display
               3-->magnetic tape
               4-->bisynch line
               5-->printer
               6-->3066 display
               7-->channel to channel adapter
               0-->other

     RDVBTS    X'80' on-->sense in progress
               X'40' on-->I/O queue busy
               X'20' on-->DE not received
               X'10' on-->proceed
               X'08' on-->wait-taken with UE or UC
               X'04' on-->program error on inop device
               X'02' on-->unit check
               X'01' on-->unit exception

     RDVBTS2   X'80' on-->spurious device end received
               X'40' on-->device has subchannel
               X'20' on-->wait for CE and not DE
               X'10' on-->start I/O pending
               X'08' on-->unused
               X'04' on-->select on same path
               X'02' on-->use RDVRSTRT as CAW
               X'01' on-->wait for any interrupt

     RDVBTS3   X'80' on-->OLTEP Attn glitch
               X'40' on-->GTUNIT queued
               X'20' on-->control unit end since SIOF
               X'10' on-->one sec.  device end pending
               X'08' on-->OLTEP operation
               X'04' on-->HALTIO completion pending
               X'02' on-->unused
```

X'01' on-->unused

CSW and SNS are the channel status word and device sense data (1 byte length, 1 byte SNS flag, SNS data).

ATNX, SNSX, PCIX, and ENDX represent the Attention, auto-sense, PCI and Endexit area addresses respectively.

The remaining lines are the entire device table printed in hex.

/DISPLAY LOCK n

where n is a lock number from 1 to 15

```
 1 = Unused (was command statistics lock)
 2 = VOLGET lock
 3 = Unused (was 3270 lock)
 4 = Unused (was PLOT lock)
 5 = unused (was BUFALLOC lock)
 6 = Tape mount queue (was ACTIVCNT lock)
 7 = SYSTEM SYMBOL TABLES and ENDSEG2 lock
 8 = STAT job lock (was FRED words lock)
 9 =Open file table lock
10 = Catalog extension lock
11 = Unused (For OS/360 Queuer)
12 = HASP CANCEL processor lock
13 = Console lock for MOUNT (unused at UM)
14 = Catalog buffer for public (*) files
15 = Create scratch file directory lock
16 = Unused
```

/DISPLAY name

This command may be used to display the current value of the predefined variables described below. The values printed are in decimal. For bit flags, a zero indicates "off" and a one indicates "on".

Example: /DISPLAY UAM

```
/DISPLAY   [R]location[+disp[+disp... ]]  [n]
/DREAL     [R]location[+disp[+disp... ]]  [n]
```

These commands cause n (decimal) words starting at the memory location specified to be displayed in hex. The DREAL command always displays real memory. The DISPLAY command displays virtual memory unless the address to be displayed is prefixed with an R.

```
Examples: /D 203140
          /D 203140+1B4
          /DR 40   2
```

```
/D R40+C0
```

/FIO {dev-name|dev-addr} csw

This command causes a fake I/O interrupt to be generated for the device specified by dev-name or for the hex dev-addr using the hex CSW given.

```
Example: /FIO DS06 0 0C000000
```

In the example above, the underlined portion of the CSW is of most concern to the one issuing the FIO commands. Some of the codes in this byte are listed below.

```
02-control unit end
08-channel end
04-device end
2C-control unit end, channel end, device end
```

FIO should be used very judiciously, probably under the supervision of a systems programmer.

/JOB [job#]

This command given without a job number generates a table showing the user id, the master source device name, the job number, number of virtual pages in use, and CPU time used for each active MTS job as follows:

```
cccc dddd jjjj vvvv tt
  |    | |  | |  | | |
  |    | |  | |  | | └---CPU time in whole seconds
  |    | |  | |  | └---*=page wait
  |    | |  | |  └---virtual pages
  |    | |  | └---vertical bar if disk attached
  |    | |  └---job number
  |    | └---blank=neutral, +=privileged, -=non-privileged
  |    └---device name or *-file
  └---user id
```

A summary line is then generated as follows:

```
hh:ss  nB  nT  nVP  nRP  n%  n XEC  n PTR  n PCH  n HLD
  |    |   |   |    |    |   |      |      |      |
  |    |   |   |    |    |   |      |      |      └-HASP jobs held
  |    |   |   |    |    |   |      |      └---HASP punch queue
  |    |   |   |    |    |   |      └---HASP print queue
  |    |   |   |    |    |   └---HASP execute queue
  |    |   |   |    |    └---%HASP spool pack in use
  |    |   |   |    └---real pages
  |    |   |   └---virtual pages
  |    |   └---active terminal jobs
  |    └---active batch jobs
  └time
```

If a job number is given, then more detailed information about that job is given:

```
jjjjj nnnnnnnn tttttt ssssssssssss BTS2 cccc pppp
   |        |      |            |    |    |   |
   |        |      |            |    |    |   |
   |        |      |            |    |    |   └---project
   |        |      |            |    |    └---user id
   |        |      |            |    └---BTS2 id
   |        |      |            └---task status
   |        |      └---job table address
   |        └---job name
   └---job number
```

```
XXXXXXXX nnnnn.nnnn nnnnn.nn nnnn nnnn nnnn nnnnn nnnnn
  |          |        |       |    |    |     |    |
  |          |        |       |    |    |     |    └---file I/O
  |          |        |       |    |    |     └---page-ins
  |          |        |       |    |    └---real pages needed
  |          |        |       |    └---real pages
  |          |        |       └---virtual pages
  |          |        └---supervisor state time (secs.)
  |          └---problem state time (secs.)
  └---PSW (right half)
```

```
R0-7:

xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx xxxxx

R8-F:

xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx xxxxx

FPRs:

xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx xxxxx
```

This command continues to produce ouput until interrupted by an attention interrupt (PA1 on a 3270) or until the JOBS job is STOPped or GOOSEd.

Examples:

```
/JOB

MTS. DS06 02181  19 07 SYS.      00006  34 62 K3S4 CA00 02282  8 15
0CX8 LA06 02211  13 07 2LFG LA26 02231  55 02 2LFM LA29 02234  7 00
2GDR CB06 02310   8 11 5C44 CB07 02311  86 04 SG0A CB08 02312  8 15
```

```
             .
             .
             .
RSTR      02462  10 00
 13:40  1B  76T  3373VP 841RP  13%  0 XEC 49 PTR  0 PCH   7 HLD


/JOB 2376

 Job    Name   Jobtbl    Status    BTS2 User Proj Par
    ILC    Prob Timt  Super Time   VP   RP  RPN Page-ins  File I/O

02376      MTS 06E090 Reading DS40  22  W225 WOPN
4030D120    19.555       4.318   77    0    1    8491     14511

R0-7:
80000001  00E1F670  00004088  00706018  0030B9AC  00000014 0000000F
000014D0

R8-F:
8030CEE6 00706000  A030F8D2  0030BB20  0030CB20  00706178  5030D0E0
001A16A8

FPRs:
00000000  00000000  00000000  00000000  00000000  00000000 00000000
00000000
Dev: DS40 00
```

/KILL job#

Stops the specified job. This performs the same function as the STOP job program. This is useful when there are no job tables available.

/LOAD [JOB]

If "job" is omitted, various information about the total system operation is displayed at 20-second intervals. For 3270 console operation, this same information is constantly displayed at the top of the screen. The information included is as follows:

DT (delta time) is the actual time between successive intervals of output (nominally 20 seconds).
EXQ (execution queue) is the number of batch jobs waiting to execute.
PTR (print queue) at the currently released priorities is the number of jobs waiting to print.
PCH (punch queue) is the number of jobs waiting to punch.
BP (batch preferred) is the desired number of batch jobs the system would prefer to execute under the current system load.

An asterisk "*" following this number indicates that the system is currently overloaded.

AB (actual batch) is the number of batch jobs currently executing.

BPH (batch per hour) is the approximate batch throughput rate (in jobs per hour) measured over the last several minutes.

LL (limited lines) is the number of terminal users currently operating in limited-service state (LSS).

AL (actual lines) is the total number of non-batch MTS users currently signed on.

VP (virtual pages) is the total number of pages of virtual memory currently in use by all jobs.

RP (real pages) is the number of virtual pages occupying real pages for all jobs.

DPA (drum pages available) is the number of pages available for use on the primary paging device(s).

PA (paging activity) is the average number of page-in operations (drum reads) per second over the previous interval.

DA (disk activity) is the average number of disk I/O operations per second over the previous interval.

CA (channel activity) is the average number of channel I/O operations per second over the previous interval.

%PI (percent processor idle) is the average CPU idle (wait state) time over the previous interval.

Q (queue) is the current number of jobs on the system CPU queue.

TQ (tape queue) is the current number of jobson the tape queue.

The values for DPA, PA, DA, CA, and %PI may be followed by a plus sign "+" indicating that the condition represented by the number is excessive, i.e., that the system is overloaded in this aspect. The presence of any plus sign causes the asterisk summary mark to be shown after the BP value.

If a job number "jobs" is specified, then a sampling of information about the specified job is displayed every 20 seconds as follows:

DT (delta time) is the same as above.

DPT (delta processor time) is the number of seconds of CPU time used by the job during the previous interval.

VP (virtual pages) is the number of pages of virtual memory currently in use by the job.

RP (real pages) is the number of virtual pages owned by the job currently in real memory.

RPN (real pages needed) is the number of pages currently estimated to constitute the job's working set.

PA (paging activity) is the average number of page-in operations (drum reads) per second for the job over the previous interval.

DA (disk activity) is the number of file I/O operations performed during the previous interval.

%PT (percent processor time) is the percentage of total system processor time used by the job during the previous interval.
%VP (percent virtual pages) is the percentage of the total virtual memory being used by the job.
%RP (percent real pages) is the percentage of the total real memory in use that is being used by the job.
%PA (percent paging activity) is the percentage of the total paging activity being done by the job during the previous interval.
STATUS the status of the job (e.g., PAGE WAIT).
%PI (percent processor idle) is the same as above.

This command continues to print information every 20 seconds until interrupted by an attention interrupt (PA1 on a 3270) or until the JOBS job is STOPped or GOOSEd.

Examples:

```
/LOAD
/LOAD  123
```

/OPER [text]

This command given without a parameter causes each line that appears on the UMMPS operator's console to be printed until the command is terminated with an attention interrupt. If a parameter is given, it is treated just as if it were typed on the operator's console. Operator commands that begin with a dollar sign ($) are taken to be HASP commands, while all other commands are treated as startjob requests or UMMPS commands. System Status commands prefixed with a slash (/) or 3270 Operator console device commands prefixed with a percent sign (%) may not be entered using this command. Any output that would be produced by the OPER command, including error messages, will appear on the real operator's console.

This command,if given with no parameters, continues to produce output until interrupted by an attention interrupt (PA1 on a 3270) or until the JOBS job is STOPped or GOOSEd.

Example:

```
     /OPER
     /OPER MTS DS06
```

/QUEUE [{receipt|ALL|LCL|station|RMTn}] ...

The QUEUE command displays the status of individual batch, *PRINT*, or *PUNCH* jobs and/or the status of any or all remote batch stations.

If a receipt number is specified, the queue status of the specified job (e.g., awaiting execution, executing, awaiting print, printing, etc.), the priority and routing of the job, and the relative position of the job within its queue are printed.

If ALL or no parameter is specified, counts of the total number of pages waiting to be printed and cards waiting to be punched along with the print, punch, and execution queue lengths are printed.

If a particular station (LCL, station, or RMTn) is specified, the connect status of that station in addition to the information specified by ALL is printed. The print and punch counts and queues reflect only the jobs routed to that station. The execution queue reflects the entire execution queue, not just the jobs submitted from that station.

Multiple parameters of the form shown may be entered, separated by commas or blanks.

Examples:

```
/Q  1234
/Q  CNTR
/Q  RMT7, 171324
```

/SCREEN [{ALL|LCL|station|RMTn}]

The SCREEN command displays continuously the status of each batch, *PRINT*, and *PUNCH* job for the specified station. If ALL or no parameter is specified, the status of all jobs in the system is displayed. The name SCREEN derives from the fact that a television-like display of relevant job status information is generated.

The format of the display is as follows. First, the system SIGNON message (if any) is printed. Next, a summary line is printed consisting of the date and time, the number of terminal users, the number of batch jobs executing and awaiting execution, and the approximate batch throughput rate. Next, the status of each job is given by a one-character code following the job receipt number. The codes are:

A - awaiting execution	E - executing
M - awaiting print	P - printing
T - awaiting punch	U - punching
W - awaiting network send	S - being sent on network
X - awaiting page print	# - done (page printer)
Y - page printing	
* - done	

Finally, a summary line is printed giving the number of jobs printing, waiting to print, punching, and waiting to punch.

This command continues to print information until interrupted by an attention interrupt (PA1 on a 3270) or until the JOBS job is STOPped or GOOSEd.

Example:

```
     /SC CNTR
```

Staff Only Status (SOS)

Note: A Signonm message should be installed when the system is in SOS mode. Don't turn SOS on unless execution is held.

```
/MOD SOS 1    (begin SOS)
/MOD SOD 0    (end SOS)
```

While SOS is on, users are prohibited from signing on.

/TAPEQUEUE [descriptors]

This command permits the operator to exercise control over the queue into which tape mount requests have been placed. See the section "TAPE HANDLING" in this manual for instructions on the queueing process.

```
/TAPEQUEUE
     - prints the size of the queue

/TAPEQUEUE LIST
     - list the current queue

/TAPEQUEUE POP [6250|800]
     - releases the top (reel tape) job in the queue

/TAPequeue POP [38000]
     - releases first cartridge request from queue

/TAPEQUEUE RELEASE
     - releases all jobs in the queue

/TAPEQUEUE RELEASE JOB #
     - releases job number "#"

/TAPEQUEUE ABORT
     - aborts all of the jobs in the queue

/TAPEQUEUE ABORT JOB #
     - aborts job number "#"
```

To turn tapequeueing on,

```
/mod tapeq 1
```

To turn tapequeueing off-

```
          /mod tapeq 0
```

To display the status of TAPEQ-

```
          /dis tapeq
```

/TASKS [descriptor]

The TASKS command prints one line of output for each job which fits the descriptor category. Each output line contains, from left to right:

The job number
The job name
The job table address
The number of virtual memory pages in use by the job
if relocatable (such as MTS)
The job status
The job table parameters (if any)
The I/O devices attached to the job (if any)

For MTS, the job table parameters are the four-character signon ID (blank if the job has no user signed on), the four-character project number (IDLE if no user is attached, and BUSY if a user is attached, but not signed on), the receipt number (if it is a batch job), and the file name (if it is a *-file job).

The descriptors specify the type of jobs to be displayed. The valid descriptors are:

xxxx	job number "xxxx"
B	all batch jobs
F	all jobs (full)
M	all MTS jobs
N [xxxx]	all non-MTS jobs, or job name "xxxx"
D xxxx	device name "xxxx"
T xxxx	device type "xxxx"
U xxxx	user signon ID "xxxx"
P xxxx	project number "xxxx"

If a job number is specified, the first nonnumeric character terminates the number; the remainder of the input line is ignored. If the first nonblank character in the descriptor is B, F, or M, then the remainder of the input line is ignored, e.g., it is permissible to enter MTS instead of M. If the first nonblank character is D, T, U, or P, then the argument for the descriptor begins with the next nonblank character, unless there are more than four characters given, in which case only the last four are used. The N descriptor followed by a 1- to 8-character job name (e.g., PDP) specifies all jobs with the given name. If no name is given, all non-MTS jobs are printed. In the argument for the D, U, and P descriptors, the pound sign "#" is used as a fill character which matches any character.

Examples:

```
/T T 9TP
/T P W###
/T U 0123
```

/USERS

The USERS command displays counts of the current number of terminal users, batch jobs executing, available terminal lines, non-MTS jobs running, pages of virtual memory in use, and pages of real memory in use.

Example: /U

The following is a list of the predefined names that may be used with the DISPLAY, ALTER, and MODIFY commands:

AESTATUS	Status of UMnet PCP AE
ABSTATUS	Status of UMnet PCP AB
ADSTATUS	Status of UMnet PCP AB
ANSTATUS	Status of UMnet PCP AN
APG	Real memory page frames available.
AVAILJT	Current number of available (unused) job tables
BDL	Batch startup delay (secs).
BJP	BJPMOD enabled flag (0->off, 1->on).
BJT	Big job threshold (real pages).
BPH	Batch jobs per hour.
BTR	Number of batch processor to allow.
CASE	BJPMOD case number.
CAW	Channel activity weight.
CS	Disk cache size
CCSTATUS	Status of Data Conc C
C1L	Class 1 load (CPU).
C1S	Class 1 static (CPU).
C2L	Class 2 load (paging).
C2S	Class 2 static (paging)
C3L	Class 3 load (disk I/O).
C3S	Class 3 static (disk I/O).
C4L	Class 4 load (channel operations).
C4S	Class 4 static (channel operations).
C5L	Class 5 load (drum space).
C5S	Class 5 static (drum space).
DIW	Disk I/O weight.
DRI	Drum read increment.
GRABS	Number of job tables for 3270 grabs
ITL	BJPMOD CPU idle low threshold
ITH	CPU idle time too high threshold (*100%).
LBJ	Lower big job threshold limit (real pages).
LLF	LSS flag (0->off, 1->on).
LPJ	Least number of privileged jobs threshold.
LSS	LSS flag enabled by operator (0->off, 1->on).
LTH	Maximum execute queue length before LSS.
MAB	Maximum number of batch processors to allow.
MAXBJTSM	Minimum big job time slice multiplier
MBH	Minimum number of batch jobs per hour before LSS.
MDR	Minimum value for drum reads to time-slice end.
MIB	Minimum number of batch processors to allow.
MINBJTSM	Minimum big job time slice multiplier
MPP	Maximum real pages for all privileged jobs.
MSP	SSRTN master job sample period (300's sec).
MTHITH	Maximum time slice high threshold
MTHIVAL	Maximum slice high value
MTLOTH	maximum time slice low threshold
MTLOVAL	Maximum time slice low value
NPJ	Mean number of non-privileged jobs last interval.
NTM	No tape mounts (0->off 1->on)
PHI	Paging load high threshold (%).
PJ	Mean number of privileged jobs last interval.

PKW	Disk paging weight.
PLO	Paging load low threshold (%).
PMW	Drum paging weight.
PQ	Mean page queue length last interval.
PQW	CPU queue weight.
PRLOW	BJPMOD paging rate low threshold
PUW	CPU Utilization weight.
Q	Mean CPU queue length last interval.
QLL	Queue length limit used by BJPMOD.
SAT	System saturated flag (0->off, 1->on).
SPAREJT	Desired minimum number of spare job taples
TAPEQ	Permits operator to use tape queueing (0=off,1=on)
TAPEQL	Tape queue length
TFB	Too few big jobs threshold (%).
TFP	Too few privileged jobs (not currently used).
TMB	Too many big job threshold (%).
TMP	Too many privileged jobs threshold (%).
UAM	Unattended mode flag (0->off, 1->on).
UBJ	Upper big job limit (real pages).

TAPE HANDLING

Users wishing to submit tapes must first run *tapesubmit from a terminal and then place identifying sticky labels on the reel. One of the labels contains spaces for two pieces of information, 1) the external tape ID, 1-10 characters and 2) the receipt code which is assigned by *tapesubmit. This label looks something like the following:

```
           Tape ID:
            ___/_/___
          ____/_/UM Computing Center
             / /
            Receipt:
```

If the tape is to be initialized, a green sticker with an "I" in the center must be placed on the reel. If the user wishes to keep the tape from being written on, two "NO RING" stickers should be placed on the reel, one on the front and the other on the back.

DPA Action

When users run *tapesubmit, the information which they type in is entered in a data base. When users present tape(s) at the I/O window another label must be filled out with the user's name, department, phone, address, etc.

A *tapesubmit program is continuously running on a terminal/screen near the I/O window. A DPA, upon receiving a tape, enters information as requested by the program:

```
initials?

reel size?

tape ID?

receipt?
```

The program responds with either:

```
***tape (receipt no.) assigned to rack number xyyy
```

where x is a letter and y is a digit, or:

```
***That tape ID does not match the record
```

Accepted tapes are then taken to the operators' area.

When users wish to retrieve tapes, a program named *taperetrieve is run, at which time, information pertinent to the tape is entered. The running of this program causes a request to appear on the operators' console as follows:

```
00525 08:47:01 MTS K4MU - Tape Retrieval:(c0414) Tape ID  'SE1',
Return via NUBS
00525 08:47:02 MTS..
```

At this point, the operator may type-

```
           Enter the receipt code of the tape

           NA  "receipt  code"           (if the tape is mounted
           or unavailable)

           II "receipt code"           (if ID is incorrect)
```

If the receipt code is correct, it is typed in, which removes the tape information from the data base, and a delivery code sticker is attached to the reel. An operator should note the time and date on this sticker. Presently, the stickers are printed as NUBS, UNYN, CNTR, MAIL, FLNT and DBRN. The tape is then placed in a rack near the I/O window.

A user, signed on at a terminal or running HASP batch, may at any time request a tape to be mounted. Users do this by using the MOUNT command (or subroutine) as outlined below:

```
$mount c3296 on 9tp *t* vol=restl2 'research2'
```

(See MOUNT in MTS Volume 1 for complete description).

If a batch user were to run above mount request, the following would appear on the operator's console:

```
00097 09:37:05 MTS SQB1 HB (1 9TP) (C3297)  ON  9tp,  RING  OUT,
'RESEARCH2'
00097 09:37:25 MTS..
```

This indicates to the operator that the user signed on as task 00097 requests the tape with Rack No. c3296 to be mounted with the ring out on a 9 track drive. The tape is supposed to have the paper label 'RESEARCH' on it.

The operator would then either locate the tape and in response to request from task 00097 reply "C3296 Txxx" to indicate the tape is (or will be) mounted on drive Txxx or else indicate that for some reason the tape cannot be mounted.

If the tape is mounted and the rack number the operator types (i.e., C3296), the ring status (in or out), the device type (i.e., 9TP), and the tape volume serial that is written on the tape (magnetically) (i.e., RESTL2) agree with what the user requested; then the operator receives (after mounting the tape) the messages:

```
00097  09:38:58 MTS Mount check OK.Mount (C3297) 119-leek-786 on
T902 Ring Out
```

```
00097 09:39:00 MTS SQB1, 119-leek-786:Accepted on T902
```

To let the user know that the tape was mounted OK. If the Rack #, ring status, Device Type, or Volume Label is incorrect then an error message is generated.

If the user just wants a tape to use temporarily and not save, he requests a pool tape:

```
     $mount pool on 9tp *p*
```

and the request on the operator's console has POOL for a rack number. There are a set of special pool tapes to use. These tapes are labeled (magnetically, as well as visually): POOL91, POOL92,... always mount these with the ring in.

For detailed information on the possible responses and error messages from MOUNT see below.

RESPONSES TO MOUNT

If more than one tape is being mounted by the mount request, a preliminary message is issued to the operator listing all the rack numbers, the device types, densities and whether or not the rings are to be IN or OUT. With this information, the tapes may be gotten out ahead of time. For example,

```
     MOUNT:  C3276, C0701
```

The operator is aware a tape mount is desired when the following message appears on the operator's console:

```
ccid:q tt (n 9TP ) rrrr ON tTP, RING xxx, 'paper label'
```

ccid — is the signon id of the user requesting the tape mount

q — is 'Q' if the request is from the mount queue

q — is '¤' if this is the first instance of the request

```
tt        is 'HB' if request from hasp batch job.
             ' T' if request from terminal.
             ' F' if request from *file job
          ' B' if request from "old-style" batch job
```

n is number of 9 track tapes desired

rrrr is (a) the rack name (alphanumeric),
or (b) POOL for a pool tape.

t is '9' for 9 track tape
'7' for 7 track tape

xxx is 'OUT' for ring out on tape
'IN' for ring in on tape

'paper label' must match what is written on a paper label affixed to the tape.

Valid responses are:

[OK] rrrr Txxx [comment]

To indicate that tape with Rack Number "rrrr" is on drive "Txxx".
e.g., ok c3296 t902
When mounting a pool tape, the response must indicate which pool tape was mounted, for example, "ok pool91 t907".

NA rrrr comment

To say NOT AVAILABLE to tape request -- the comment must appear and will be printed on user's output to indicate why operator did not mount tape. Do not use if all tape drives are in use; use BY for terminal requests and RER for batch requests.
e.g., na 9876 no such tape

ABORT rrrr comment

To say NA to this tape request and skip remaining tape mounts of a multi-tape mount request. comment must appear and will be printed on user's output to indicate why tape request was aborted. (The alternative to "AB" is an "NA" for EACH of the tapes of a multi-mount.)
e.g., ab 9876 no such tape

LE rrrr [comment]

Label Error -- to be used to signal a label error to the user. This response should be given only after the TAPE LABEL INCORRECT message to the operator and only after the operator has re-verified that the correct tape has actually been mounted. If the wrong tape was inadvertently mounted, the correct one should be mounted and the OK response should be given again.
e.g., le c1104

IM rrrr [comment]

IM -- user gets response -
p-name: NO UNIT AVAILABLE WHICH SUPPORTS COMBINATION OF MODES REQUIRED
Example: IM C4097

The IM response can be used if a user tries to mount a tape currently recorded at 800 BPI on a 6250 tape drive. The IM response can also be used at installations which have no 6250 tape drives if someone tries to mount a tape using 6250 mode.

II rrrr [comment]

II -- user gets response -
p-name: INCORRECT RACK NUMBER OR TAPE ID
Example: IR C4097

BY rrrr [comment]

BUSY -- to be used to tell user that no drives are currently available. This response will not work for HASP jobs -- use "RER" for them.

REPEAT

The mount request is to be printed again.

MSG rrrr comment

To print a message to the issuer of the mount request.
The read is then repeated (for an 'OK', etc.).

EXAMPLE: msg c5823 please call operator

RERUN rrrr

This will $RERUN the job (works only for HASP jobs) and should be used whenever the operator desires to run the job at a later time (normally because enough free drives are not currently available). The HASP command $RELEASE JOB n is used later to release the job.

The options enclosed in brackets in the following three commands should be used to make the TAPE QUEUEING routines aware of the modes of the tapes that are being queued. The actual density of the tape should be given if the user didn't specify it. See the description of TAPE QUEUEING below.

QB rrrr [6250|800|1600]

Queue this user (when all drives are busy, etc.)

QP rrrr [6250|800|1600]

Queue this user and displays the top request in the queue (pop top job)

QN rrrr [6250|800|1600]

Queue this user and display next request after this one (NEXT). This is used to return the user to the queue because it was not possible to satisfy the user's request even though there are unused drives.

TAPE QUEUEING

If TAPE QUEUEING is enabled, i.e., TAPEQ=1, a mount request that cannot be fulfilled because there is no available drive of the give density should be given the QB reply instead of BY or RERUN. The terminal user will then be promted for approval to be placed in the tape queue, e.g.,

```
*t* (rrrr): all units busy at this time.
there are N terminal users waiting for M drives ahead
of you in the mount queue.
do you wish to wait for an available drive?
```

If the user responds OK, then the user will be placed at the end of the current mount queue. If the user's response is NO, then the mount request is aborted, and the comment,

```
CCID, RRRR: REQUEST ABORTED BY USER
```

is printed on the operators' console. Batch jobs are always queued if the QB response is given.

The terminal user may interrupt the tape queue wait at any time by sending an ATTENTION. If this is done, the system will remove the user from the queue, print the user's current position in the queue, and ask if the user wants to continue to wait. If an affirmative answer is given, the user is returned to the queue at the previous position, if possible. Otherwise, the mount is abandoned and a message is printed on the operators' console as above.

When a job releases a tape drive, not a remount, the system will release the first job in the queue that needs a drive of the same density or an unknown density. A request will have an unknown density if the user did not specify the mode of the tape, i.e., (mmmm), where mmmm is 6250, 1600, or 800 b.p.i., did not appear on the mount request, or the operator did not give a density on the QB reply, i.e., the reply was "QB rrrr" instead of "QB rrrr mmmm". The drive should be given to this request, it will be flagged with the character "Q" indicating that it came from the tape queue, before any brand new, non-queued, tape requests.

If the drive is not compatible with the tape, i.e., no density was specified at the first queueing, the request should be returned to the queue using the reply

```
qb rrrr mmmm
```

The actual density, mmmm, of the tape, 6250, 1600, or 800 b.p.i., should be given on the reply so that it will be presented again only when an appropriate drive is available. The user is returned to the queue, in this case, with no message printed on the terminal. The "QN" response will cause the next request to be presented, since this one cannot use the available drive.

If the user requests more than one drive but not enough drives are available, each request should be queued separately. The operator should QB the first request; when that is satisfied, the system will present the next request which should then be queued, if necessary, and so on. The user will be promted to see if it is OK to queue the first request but subsequent ones will be queued way, the operator need not leave several tape drives idle waiting for enough of them to completely satisfy the request.

Before a request is initially queued, the operator should check the rack number and tape i.d. to make sure that they are correct so that a user won't have to wait several minutes in the queue just to get an "IR" response. If the user did not specify the density, the operator should do this on the QB response so that it won't be "popped" until an appropriate drive is available.

The following are SYSTEMSTATUS commands and are included in the SYSTEMSTATUS section of this manual. They are mentioned here because they are part of the tape mount queueing process.

```
/tapequeue
       - prints size of queue

/tapequeue list
       - list the current queue

/tapequeue pop [6250|800]
       - releases top job in queue requesting
       given density.  If no density is given, the
       top job is "popped".

/tapequeue release
       - releases all jobs in the tape queue
```

```
/tapequeue release job #
       - releases job number "#"

/tapequeue abort   (works like the "ab" response)
       - aborts all of mounts of the jobs  in  the
       queue

/tapequeue  abort  job  #   (works  like  the "ab"
       response)
       - aborts the mounts of job "#"

/display tapeq (0=off,1=on)

/mod tapeq (0=off,1=on)
       - used  to  enable  or  disable  the  tape
       queueing mechanism.
```

Any null or cancelled response is ignored and the read is repeated.

Many responses allow an optional comment to be specified as the last parameter on the line which will be printed on the users' terminal or batch output, e.g.,

```
C0045 T902 CALL MACHINE OPERATOR
```

prints:

```
*T* (C0045) MOUNTED ON T901.  CALL MACHINE OPERATOR.
```

Any error messages generated by incorrect replies should be self-explanatory. (The normal problem being operator confusing 2 simultaneous mounts and replying to wrong one).

In checking for correct volume, I/O errors are sometimes generated (especially 7 track tapes on 9 track drives).

NOTE: Ocassionally, a tape will run off the end, i.e., the rear end of the tape is actually pulled off of the feed reel. Operators should never re-thread and/or rewind the tape until a "not ready" indication appears on the Operators' console since this may cause writing over information already present on the front of the tape. If the tape was being read, the premature rewinding would cause incorrect information to be read.

TAPE ERRORS

```
TAPE ERROR TXXX/AAAA:BB C DDEE FF-XXXXXX... IIII,JJJJ,KKKK,LLLLL,MMMMM
```

The following Tape Error information is used by *RST

TXXX:	Tape drive name
AAAA:	Path
BB:	Software flags
C:	Error codes (see TAPE ERROR CODES)
DD:	Device status (from CSW)
EE:	Channel status (from CSW)
FF:	Sense flag (should be 01)
XXXXXX...:	Sense data (up to 24 bytes)
IIII:	Rack number (up to 6 characters)
JJJJ:	Volume Serial number, if labeled (up to 6 characters)
KKKK:	Data set name if labeled (up to 17 characters)
LLLLL:	File number
MMMMM:	Block number

TAPE ERROR CODES

The tape unit check routines are a collection of subroutines which are invoked by most programs when the tape drive they are using detects an error. The purpose of these routines is to correct the error, or if it is uncorrectable, to report the error to the operators console. All errors are logged in CMDSTAT and may be retrieved by using EREP. Recoverable errors do not cause a message to be written to the console, but unrecovarable ones do.

The job programs which use the tape unit check routines are: MTS (and almost all programs which are run by MTS), HASP, the STAT job, and the collection of tape utility jobs (FSF, BSF, REW, RUN, ERG, and WTM).

The above tape error information is used by *rst

TAPE ERROR CODES

The following tape error information is pertinent to everything but *rst

An explanation of the tape error messages which are displayed on the operators console is given below. Each error produces a three line report of the error which gives device name, the sense data associated with the error (which precisely describes the error in machine terms), the position on the tape where the error was encountered, and a short descriptive message giving the nature of the error. A prototypic message is as follows:

```
aaaa error b cccccccc dd eeeeeeee eeeeeeee eeeeeeee eeeeeeee
eeeeeeee eeeeeeee ffffffff gg

aaaa gggggg hhhhhh iiiiiiiiiiiiii file jj block kk aaaa llll...
```

Each of the underlined areas vary depending on which device declared the error and what type of error it was. These are described below:

aaaa designates the device which detected the error. Notice that this appears at the front of each of the three lines.

B is a one character code which uniquely describes the type of error. These don't need to be memorized because a more readable description of the error is given on the third line of the message. See the description of the llll part of the message below for all the different errors.

Cccccccc gives the hexadecimal value of the unit check routines' input switches. This is only of use for programmers which work on the unit check routines themselves.

Dd gives the hexadecimal value of the supervisor's sense flag. This value is interpreted by the unit check routines to determine the cause of some kinds of tape errors. It is mostly of interest to programmers working on the unit check routines, but also required to interpret channel detected errors (described below). If this value is anything other that 01, the sense information will be unreliable.

Eeeeeeee ... Is the hexadecimal sense data associated with tape error. Again, this is interpreted by the unit check routines to determine the type of error and how to recover from it. This information may be of use to both tape vendor customer engineers (if they are attempting to correct a hardware problem) and also to programmers working on the unit check routines themselves.

Ff is the hexadecimal value of the channel command the drive was trying to perform when the error occurred. Typical values for this field are:

01 - write tape
02 - read tape in a forwards direction
0C - read tape in a backwards direction

07 - rewind tape
0F - rewind and unload tape (RUN)
1F - write tape mark on tape
27 - backspace block on tape
2F - backspace file on tape
37 - forwardspace block on tape
3F - forwardspace file on tape
CB - set tape mode to 800 bpi
C3 - set tape mode to 1600 bpi
D3 - set tape mode to 6250 bpi

There are other values which can be encountered here, but should not normally be. This field can be useful for customer engineers when they attempt to repair a broken tape drive.

Gggggg is the tape rack number if the tape has been mounted by the $MOUNT command. If it wasn't mounted this way, this field contains a dash.

Hhhhhh is the tape volume label if the tape is labeled. If it is not, then this field contains a dash.

Iiiiii... Is the current dataset name on the tape, which only appears if the tape is a labeled tape. If not, a dash appears here.

Jj is the file number in which the error was encountered. This, along with the block number (see below) identifies precisely where on the tape the error occurred.

Kk is the block number in which the error was encountered.

Llll... Is the description of the nature of the error. This is a line of English text which is terse, but describes the cause of the error. A list of each of these is given below, along with the error code (which appears in the first line of the error message) associated with each.

ERROR MESSAGE TEXT AND EXPLANATION

Code

0 "Program interrupt looking at channel program" This error is always caused by a program bug.

A "Invalid or insufficient sense data" This error is almost always caused by a tape drive which is broken so badly it can't tell what is wrong with itself. It can also be caused by indiscreet usage of SVC CLRIO, which is a programming problem.

C "Channel detected error" This error occurs when the tape drive is operating properly (or thinks it is) but the channel detects a problem which may or may not be the fault of the device. This can be caused by either a programming error, failing hardware in the tape drive, channel, or even the CPU. To pinpoint the error, the bits in the sense flag field (dd above) must be examined. The symptoms are:

80 - Program check (programming bug)
40 - Protection check (programming bug)
20 - Channel data check (hardware problem)
10 - Channel control check (hardware problem)
08 - Interface control check (hardware problem)
04 - Chaining check (either program or hardware problem)

To decode this value you have to decode the bits indicating each of these symptoms from the sense flag byte. For example, a sense flag of 18 means channel control check and interface control check were declared.

D "Equipment check" This error is caused by an ailing tape drive.

E "Bus out check" Either an error in the channel or an error in the cable connecting the device to the channel can cause this.

F "Command reject" Usually this is caused by a bad tape drive. It can also be caused by programming bugs if the program tries to tell the tape drive to do something it doesn't understand.

G "Overrun" This condition signifies that there is a mismatch between the speed of the tape drive and the channel to which it is connected. It can occasionally be caused by a hardware malfunction, but usually signifies an overloading of a low-performance CPU (such as an IBM 148). It can on rare occasions be caused by reading a tape which was written by an improperly adjusted tape drive.

H "Load point reached but drive not 'backwards'" This signifies a tape drive problem. The drive is reporting inconsistent status about itself.

I "Data check" This error is usually caused by a flakey tape, not necessarily a flakey tape drive. If a drive is out of adjustment, it may declare errors like this on tapes which can successfully be read from or written on another drive. The error is reported by the drive when the data it reads from or writes on the tape is incorrect. This is usually due to worn or damaged tape.

J "Data security erase check" This message (it isn't actually an error) is given whenever the tape drive has been told to erase an entire tape using the data security erase command. The reason it causes this message to be generated is because this is a drastic action to take. It does not signify an error.

K "Id burst check" This error is signalled by the drive when it is unable to determine the density of a tape from the ID burst written on the front of all tapes. If an attempt is made to read a tape which hasn't been initialized, this error will occur because no ID burst appears at the front of a brand new tape. If this occurs, the tape must be written before it can be read.

L "Intervention required - drive not ready" This happens when the tape drive is not in the "ready" state. The drive must be made ready before any further operations may occur on the device.

M "Intervention required but not 'statusb' set" See the description for error H.

N "Unable to find failing CCW in channel program" This is usually caused by a programming bug. It can be caused by injudicious usage of the FAKEIO function too.

O "Load point reached" This error is a programming error. It arises when the program attempts to read a tape backwards and encounters the front of the tape. Unlike the end of the reel, the tape cannot be pulled of the front of the reel by a properly functioning drive.

Q "Data converter check" This error only happens with seven track drives when the data converter has not been set to the proper translation mode. It is a user (or program) error.

W "Not capable" This error occurs when an attempt is made to make a drive switch to a recording density it is incapable of handling. For example, trying to read a 6250 bpi tape on a drive only rated for 800 or 1600 bpi will cause this error to occur.

Z "Physically offline" This condition is declared by the supervisor when it attempts an operation to a device which returns an indication that it is offline. You must online the device before any further operations may successfully complete on the drive.

A sample tape error message is given below along with its interpretation.

```
T901 error A 41000000 01 40800000 00000d00  00000000  00c12312  24008000
00021234 02

T901 C0024A SAMPLE DATA/1978 file 28 block 7734

T901 invalid or insufficient sense data
```

This error declares an error on the drive named T901. The tape mounted on the drive was tape C0024A (line two) and was a labeled tape (as indicated by the label SAMPLE on line two). The error occurred during a read operation (see channel command at end of line one - 02), and the dataset being read was called "DATA/1978".

This dataset was located at file 28 of the tape, and the error occured at block 7734 in that dataset. The error was caused by invalid sense data (see line three), and most likely signifies a problem with that drive's hardware.

TAPE MOUNTS -- SUSPENSION OF

When operations' staff members aren't scheduled to be in the second floor machine room, the suspension of tape mounts, as well as other operational aspects, is necessary. There are at least two ways to prohibit the mounting of tapes.

UAM

When the system is placed in UNATTENDED mode, by altering UAM, users will receive a message instructing them to copy *UAM and informing them that, among other things, tape mounting is prohibited.

To turn UAM on--

```
/m uam 1
```

To turn UAM off--

```
/m uam 0
```

NTM

At other times, even during regular hours, when the machine room is staffed, it may be necessary to prohibit tape mounts for a period of time. Altering NTM will produce the desired results, here.

```
     To turn NTM on--
                           /m ntm 1

     To turn NTM off--      /m ntm 0
```

When NTM is on, users will be informed that tape mounting isn't possible but that this is only temporary.

TAPE MAILING - SUBMITTING AND RETRIEVING TAPES BY MAIL

Users may now send magnetic tapes and diskettes (floppy disks) by mail. Also, upon request, the Computing Center will return tapes and diskettes to users by mail.

To mail a tape or diskette to the Computing Center, the user must:

1)-Obtain a mail submission form either from the Mail Librarian at the Computing Center or by $COPYing the file *TAPESUBMIT.

2)-Package all tapes and diskettes securly and send them along with the completed form(s) to the Mail Librarian at the Computing Center:

Mail Librarian
Computing Center
1075 Beal Avenue
Ann Arbor, Mich. 48109

When tapes or diskettes are received, they will be placed in the racks at the Computing Center and a receipt will be mailed to the user.

There is a charge for retrieving magnetic tapes and diskettes by mail. This charge covers the mailing and handling costs. The cost for mailing magnetic tapes $7.00 for the first tape and $3.00 for an additional tape mailed in the same package. Each package holds only two tapes. The cost for mailing diskettes is $4.00 for the first diskette and $1.00 for each additional diskette mailed in the same package. Each package holds five diskettes. The charge may be paid by check or money order, or charged to a sponsored research account. To retrieve a magnetic tape or diskette by mail, the user must:

1)-Obtain a mail retrieval form either from the Mail Librarian at the Computing Center or by $COPYing the file *TAPERETRIEVE.

2)-Send the completed form, the yellow receipt card, and a check or money order (if this is to be the mode of payment) to the Mail Librarian at the above address.

Tape Racks - Electric (Problems)

There are two sets of electric tape racks in the first floor machine room that lie in a contiuous string. As you face west, the new ones are at the left and the older ones to the right.

Older Tape Rack Operation

If power is lost to the older ones, the fault is usually a blown fuse. The fuse is located in a silver colored box on top the north tape rack. If the fuse isn't blown, then there is probably a tripped circuit breaker. If power is lost to the older racks, then a socket wrench which should be located on the the rack with the fuse, can be used to manually operate the racks. The wrench should be inserted in the holes at the bottoms of the racks.

Newer Tape Rack Operation

The newer tape racks have an ON/OFF switch on top of their right-most unit which is near the center of the row. The fuse for for the units is located near this switch.

If power is lost to the newer tape units, there is a battery operated device which can be plugged into the waist-high holes in the units. This black box with attached wire is usually kept on top of the structure that holds the operators' console. When this box has been plugged into the tape racks, it can be operated by its white button.

There are two circuit keys near the operators console which are used on <u>all</u> of the electric racks. The yellow one, when plugged into the override slot, overides the safety features. A reason for using this overide key could be because the floor is not actuating its switches to let the racks move back and forth.
The other key is colored red on one side and green on the other. This key, when inserted in the slot below the overide slot, can be used to turn the power on or off, depending on which side of the key is facing up. If the key is inserted with the green side up, the racks will be powered on. However, the key must be left in the slot for several seconds until a "beep" is heard and the rack lights come on. When this happens, the key must be removed from the slot and the "safety reset" button pressed. The racks should now be ready for use.

If the small rectangular green light below the words "in use" is blinking, this means that one of the "sweeps" is jammed or caught on something. The "sweeps" are metal safety plates at the bottom of each rack, almost touching the floor. They are about 6 inches high and extend the full length of the rack.

Tape Rack Strip Labels

Below is a program designed to make strip labels for our tape racks. It may be altered to change the height, length and number of labels, etc., in a strip.

After any necessary alterations, the program must be compiled. and then put into a form acceptable to the plotter - more about this later.

What you must do first, is determine the height and length of the strip to be produced. The height in the program is stated as "STRIPH". As you can see in the program below, STRIPH is set at 10.0/16.0. If, after measuring the strip holder, you decide that the strip height should be 11 1/2 16ths, change STRIPH from STRIPH=10.0/16.0 to STRIPH=11.5/16.0 using the editor. "STRIPL" is the length of the strip and can be changed the same way. "NSLOTS" is the nuber of rack slots that you want in the strip. "NSTRIP" is the number of strips to be plotted. It's best to try 1 or 2 strips at first as a test to make sure that they fit. LETRH=0.4*STRIPH*FACTOR is a proportion and you may need to change this (0.4) several times until the characters on the strip appear to be the right size. NCHAR is the number of single digits following the letter, e.g., C2345. K=0000 is the starting figure in the strip, and, in the way the program is set up now, C000 would be the first number produced on the strip. Changing K=0000 to K=0001 would cause the first number to be C0001. Any of the statements mentioned in this writeup may be chnaged using the editor.
In the string
"CALL PNUMBR(XLOC,YLOC,LETRH,K,THETA,'''C'',ZI4*',SC)" the '''C' can be changed to an N or a K or whichever character is needed. If changed to an N, the first slot on the strip would be N000.

After the necessary alterations (if any) to the program have been made, the program must be compiled using Fortran. To do this-

```
     $run *ftn scards=tapes
```

If the program runs successfully,

```
     EXECUTION BEGINS
     NO ERRORS IN MAIN
     EXECUTION TERMINATED
```

should be printed. The object program automatically is placed in a file named -LOAD.

The next thing to do is to put the program in a form to be plotted.

```
     $sig w035
     $run -load+*plotsys 9=-plot

     Ready the plotter and use a #8 pen
```

Now, since the 1051 plotters are attached through the network, one must be "mounted". To do this --

```
     $mou mnet *plot* d=hXxCcb7

  where xx is 00 for plt2 and 01 for plt3
```

Run sys.:plot51

```
$run sys.:plot51


Execution begins


WHERE FROM:) -plot


WHERE TO:) *plot*Bin


SCALE FACTOR)?  1.0     (usually)


PRIMING TIME (SEC):)?  press "enter"


HEADER (CCIDDATETIMERECEIPT): press "enter"


NUBS?  no


PENS: press "enter"
```

Below is a copy of the source program to be altered, if necessary, and compiled. The name of the file is W035:TAPES.

```
     REAL LETRH,LETRL
     C
     C HEIGHT OF STRIP
     C
     STRIPH=10.0/16.0
     C
     C LENGTH OF STRIP
```

```
C
STRIPL=23.1875
C
C NUMBER OF TAPE SLOTS IN A RACK UNIT
C
NSLOTS=25
C
C COMPUTE WIDTH OF A TAPE SLOT
C
SLOTW=STRIPL/NSLOTS
C
C NUMBER OF LABEL STRIPS TO BE PREPARED
C
NSTRIP=46
C
C FUDGE FACTOR FOR LETTER HEIGHT
C
FACTOR=0.669
C
C COMPUTE LETTER HEIGHT
C
LETRH=0.4*STRIPH*FACTOR
C
C NUMBER OF CHARACTERS IN A SLOT LABEL
C
NCHAR=4
C
C LENGTH OF LETTERING FOR A SLOT LABEL
C
LETRL=PSYMLN(LETRH,NCHAR)
C
C INITIALIZE VARIOUS THINGS FOR *PLOTSYS ROUTINES
C
SC=1.0
THETA=0.0
CALL PLTXMX(48.0)
CALL PLTOFS(0.0,1.0,0.0,1.0,1.0,1.0)
C
C PREPARE THE GRID FOR ALL LABELS
C
CALL PGRID(1.0,1.0,SLOTW,STRIPH,NSLOTS,NSTRIP)
C
C SET FIRST NUMBER
C
K=0000
C
C EXECUTE NESTED LOOPS TO WRITE LABELS IN THE GRID BOXES
C
DO 200 I=1,NSTRIP
YLOC=(NSTRIP-I)*STRIPH+STRIPH/2.0-LETRH/2.0
DO 100 J=1,NSLOTS
XLOC=(J-1)*SLOTW+SLOTW/2.0-LETRL/2.0
C
C POSITION FOR LETTERING IS (XLOC,YLOC) AND THE LETTER CODE
```

```
IS "C" OR "N" OR WHATEVER YOU WANT TO MAKE IT.
C
CALL PNUMBR(XLOC,YLOC,LETRH,K,THETA,'''C'',ZI4*',SC)
C
C INCREMENT THE SLOT NUMBER
C
100 K=K+1
200 CONTINUE
CALL PLTEND
END
```

TAPE DIRECTORY PROJECT

The tape directory facility is designed to manage the tapes that are used for various functions within the system such as system statistics, command statistics, billing, etc. Currently, these functions are all handled separately by individual components within the system.

This facility may manage either a single pool of tapes that may be used for all of these functions or a series of separate pools of tapes that may be used for a single function. Each function may involve the use of several tapes; however, each tape will only be used for one particular function. which indicates what each tape is being used for and what information is contained on it.

The directory facility will provide a method for locating the tape (and tape file) that contains a certain data record (e.g., a statistics record) for a certain time or to indicate where the next data file should be written (e.g., on which file and on which tape the next statistics file is to be written).

The actual mounting of tapes and the reading or writing of files or records on the tapes will be done by individual programs calling the directory facility. The directory facility will not be mounting, reading, or writing tapes (except for initializing tapes).

The directory facility consists of a set of subroutines that may be called directly by system programs to manage the directory. These subroutines provide the capability of creating, adding, and retreiving entries from the directory. These subroutines are described later.

There is also a directory program that will allow the user to conversationally interact with the directory subroutines.

The Directory File

When running the directory program, the directory file is specified either in the PAR field of the $RUN command or by the FILE command. When calling the directory subroutines directly, the directory file is specified as the first parameter in each subroutine call.

The directory is a line file organized in the form of master entries and file entries. For each function type within the directory, there is a master entry that describes the function type and its characteristics, i.e., the mode of the tape and the formats used for the primary and secondary search fields. For each tape file catalogued within the directory, there is a file entry that describes the characteristics of the tape file, i.e., the name of the tape, the location on the tape, and the ranges of the primary and secondary search fields.

In addition, there is a master entry for the pool of available tapes (type POOL) and file (tape) entries that describe the current status of each tape within the directory, i.e., whether the tape is in use and by whom.

The format of the directory file is given below. Each entry will contain the following information:

Type code (bytes 1-4)

The type code identifies the function type of the tape.
Some of the possible type codes are:

POOL - available tape (tape currently unused)
DSTA - daily statistics tape
MSTA - monthly statistics tape
MRTB - MERIT billing tape
HNDB - Hand billing tape
CMDS - Command statistics tape
FCHG - File charge update tape
-
-
-
-

Status (byte 5)

The status bits indicate:

01 -> Primary search field present
02 -> Secondary search field present
04 -> Comments information present
08 -> Entry in progress (entry unconfirmed by CONENTRY)
10 ->
20 -> Tape restricted for particular type (marked under POOL master only)
40 -> Tape in use (marked under POOL master only)
80 -> Master entry

Tape mode (byte 6)

02 -> 800 bpi
03 -> 1600 bpi
04 -> 6250 bpi

Tape rack name (bytes 7-22)

Tape volume label (bytes 23-28)

Tape file number (bytes 29-32)

Sequence ID (bytes 33-36)

The sequence ID orders (with respect to the time the tape is written) all files on all tapes of a particular function type (e.g., all files on the MSTA tapes will be ordered from 00001 to "nnnnn"). This sequence ID will be useful for finding the "next" or last tape file when its time range is not explicitly known.

Primary search field format (byte 37)

0 = No primary search field
1 = Julian microseconds (unsigned 8-byte integer)
2 = Julian seconds (unsigned 4-byte integer)
3 = Julian minutes (unsigned 4-byte integer)
4 = Julian days (unsigned 4-byte integer)
5 = Integer (signed 4-byte integer)
6 = Long integer (signed 8-byte integer)

Primary search field (bytes 38-53)

This field is composed of two 8-byte integers. Normally, this field will consist of a Julian time range (microseconds) relating to the data contained in the file. However, this field may contain integer values (4 or 8-byte) other than Julian times if the search is being done on a basis other than time. Internally, all integers are stored right-justified, i.e., 4-byte integers are in bytes 4-7 of each doubleword.

Secondary search field format (byte 54)

0 = No secondary search field
1 = Julian microseconds (unsigned 8-byte integer)
2 = Julian seconds (unsigned 4-byte integer)
3 = Julian minutes (unsigned 4-byte integer)
4 = Julian days (unsigned 4-byte integer)
5 = Integer (signed 4-byte integer)
6 = Long integer (signed 8-byte integer)

Secondary search field (bytes 55-70)

This field is composed of two 8-byte integers. Normally, this field will consist of a Julian time range (microseconds) relating to the data contained in the file. However, this field may contain integer values (4 or 8-byte) other than Julian times if the search is being done on a basis other than time. Internally, all integers are stored right-justified. The secondary search field provides an alternative or secondary searching method for the data contained in the file.

MOUNT message field (bytes 71-166)

This field contains the MOUNT message (in character (format) needed to mount the tape containing the desired file. The VOL, RING, and MODE keywords will be presented in the MOUNT message. RING={IN|OUT} will be supplied automatically depending on whether the tape is to be read or written. The POSN keyword is not included in the message.

Comments field (bytes 167-422)

This field provides an arbitrary field for use by the calling program. The maximum length is 256 bytes.

Tape Directory Program

The program to interact with the tape directory located in the file SYS.:DIRPROGRAM. This is a GOM program and may be run by issuing the command

```
$run    sys:dirprogram    scards=input    sprint=output
sercom=errors par=fdname
```

The command input is read from SCARDS and the program output is written on SPRINT. Error diagnostics are written on SERCOM. The name of the directory file may be specified in the PAR field. The asterisk "*" is the program prefix character. The command language is described below.

Command: ADD type [TAPE={rack|POOL|RES}] [VOL=label] [PFIELD=(m,n)] [SFIELD=(m,n)] [COMMENTS=string]

Explanation: This command adds a new entry into the directory. The parameter "type" specifies the type of tape entry to be added. The tape directory program will allocate a new tape file and assign a sequence ID "n" to the entry. The entry will be marked as being "in progress" and will remain as such until confirmed by the CONFIRM command.

Normally, the new tape file will be assigned to the current tape (the tape containing the entry with the highest sequence ID). If there is no current tape, a new tape will be allocated from the pool of available tapes. If required, the new tape will be initialized to the correct mode (as specified in the master entry for "type").

If the entry is to be allocated to a particular tape, the parameters TAPE=rack and VOL=label must be specified to indicate the correct tape.

If the entry is to be allocated to a new tape, the parameter TAPE=POOL or TAPE=RES must be specified to indicate that the tape tape is to be allocated from the pool of available tapes. If RES is specified, the tape is allocated from a restricted pool of available tapes.

The PFIELD and SFIELD parameters specify the primary and secondary search field values to be assigned to the new new entry. "m" is the lower-bound value and "n" is the upper-bound value. The format of "m" and "n" must correspond to the format specified in the master entry for "type". For Julian-time formats, "m" and "n" may be given in either integer format (giving the actual number of microseconds, etc. since March 1, 1900) or in character format (giving the date and time in the form "MMxDDxYY hh:mm:ss" or "MMxDDxYY" (midnight), where "x" is any character). For example,

```
  PFIELD=(01-01-78 00:00:00,01-31-78 23:59:59)
```

These search field values may be given either with the ADD command or CONFIRM command; any value given with the CONFIRM command will override an ADD command value.

The COMMENTS parameter may be used to specify the comments field for the entry. This field is an arbitrary character string of up to 256 characters. The COMMENTS parameter must be the last parameter specified.

The sequence ID assigned to the entry is printed.

Command: CONFIRM type SEQ=n [PFIELD=(m,n)] [SFIELD=(m,n)]
[COMMENTS=string]

Explanation: This command confirms that the data for a new tape entry has been successfully written to the tape. The parameter "type" specifies the type of tape entry being confirmed. The parameter SEQ=n speci- fies which entry is being confirmed; "n" is the sequence ID that was assigned to the entry by the ADD command.

The PFIELD and SFIELD parameters specify the primary and secondary search field values to be assigned to the new new entry. "m" is the lower-bound value and "n" is the upper-bound value. The format of "m" and "n" must correspond to the format specified in the master entry for "type".

For Julian-time formats, "m" and "n" may be given in either integer format (giving the actual number of microseconds, etc. since March 1, 1900) or in character format (giving the date and time in the form "MMxDDxYY hh:mm:ss" or "MMxDDxYY", where "x" is any character). For example,

```
PFIELD=(01-01-78 00:00:00,01-31-78 23:59:59)
```

These search field values may be given either with the ADD command or CONFIRM command; any value given with the CONFIRM command will override an ADD command value.

The COMMENTS parameter may be used to specify a comments field for the entry. This field is an arbitrary character string of up to 256 characters. The COMMENTS parameter must be the last parameter specified.

<u>Command</u>:

```
CREATE type [MODE={800|1600|6250}]
       [PFMT={JX|JS|JM|JD|I|LI}]
       [SFMT={JX|JS|JM|JD|I|LI}]
```

Explanation: This command creates a master entry for "type". The master entry specifies the mode to be used for tapes allocated to this type category and specifies the formats for the primary and secondary search fields.

If the MODE parameter is omitted, the default mode will be 1600 bpi.

PFMT and SFMT specify the formats for the primary and secondary search fields, respectively. The formats available are:

```
JX - Julian microseconds (unsigned 8-byte int.)
JS - Julian seconds (unsigned 4-byte integer)
JM - Julian minutes (unsigned 4-byte integer)
JD - Julian days (unsigned 4-byte integer)
 I - Integer (signed 4-byte integer)
LI - Long integer (signed 8-byte integer)
```

If PFMT or SFMT is omitted, then there is no corresponding search field assigned to the entry.

Command: DELETE type [TAPE=rack VOL=label] [SEQ=n]
[PAR=CURR|LAST]

Explanation: This command deletes an entry or series of entries from the tape directory. The parameter "type" specifies the type of tape entry to be deleted.

If PAR=CURR is specified, the entry currently in progress is deleted (if any).

If PAR=LAST is specified, the latest entry is deleted (the entry with the highest sequence ID).

If SEQ=n is specified, the entry with sequence ID "n" is deleted.

If the TAPE=rack and VOL=label parameters are specified, all entries from the specified tape are deleted.

Command: FILE filename

Explanation: This command specifies the file "filename" as the directory file.

Command: GET type [PFIELD=(m)] [SFIELD=(m)] [SEQ=n]
[PAR=LAST]

Explanation: This command gets an entry from the tape directory. The parameter "type" specifies the type of tape entry to be accessed.

The tape directory program searches the directory for the first entry which matches the search values indicated by the PFIELD, SFIELD, SEQ, or LAST parameters. If LAST is specified, the entry with the highest sequence ID is returned.

If two search values are given, the second is searched upon only after the first is satisfied. In this case, only the PFIELD and SFIELD parameters may be used. The order in which they are specified is the order in which they are used for the search.

The PFIELD and SFIELD parameters specify the primary and secondary search field values to be used for the search. The format of "m" must correspond to the format specified in the master entry for "type".

For Julian-time formats, "m" may be given in either integer format (giving the actual number of micro- seconds, etc. since March 1, 1900) or in character format (giving the date and time in the form "MMxDDxYY hh:mm:ss" or "MMxDDxYY" (midnight), where "x" is any character). For example,

```
PFIELD=(01-15-78 12:00:00)
```

For integer format, the parentheses may be omitted.

Command: HEX

Explanation: This command prints the entire directory in hexadecimal format.

Each field is printed with a separating blank. The mount message and comments fields are omitted.

Command: INIT rack label [800|1600|6250] [type]

Explanation: This command adds tapes to the available tape pool in the directory.

If the mode parameter is omitted, the tapes will be initialized as 1600 bpi tapes.

If the type parameter is given, the tape will be restricted in use to the specified type; if the type parameter is omitted, the tape will not be restricted. If the type parameter is given, the mode parameter also must precede it.

Command: LIST [ALL] [type] [TAPE={rack|POOL}]

[PAR=CURR|LAST]

Explanation: This command lists the contents of the directory.

If the ALL parameter is specified, the entire directory is listed.

If "type" is specified, all entries for the speci- fied type are listed. If PAR=CURR is specified, only the entry in progress for "type" is listed. If PAR=LAST is specified, only the latest entry for "type" is listed.

If the TAPE=rack parameter is specified, all entries contained on that tape are listed. If TAPE=POOL is specified, all unrestricted tapes in the available tape pool are listed.

If no parameters are specified, the entire directory is specified.

Command: MTS

Explanation: This command returns control to MTS command mode. The program may be resumed via the $RESTART command.

Command: PURGE type

Explanation: This command purges the master entry and all file entries for the specified type. The tapes used for the file entries are returned to the available tape pool. Confirmation of this command is required.

Command:
```
RECREATE type [MODE={800|1600|6250}]
       [PFMT={JX|JS|JM|JD|I|LI}]
       [SFMT={JX|JS|JM|JD|I|LI}]
```

Explanation: This command respecifies the master entry for "type". This allows either the format of the primary or secondary search field to be respecified, or the mode of the tape to be changed. All files added for this type will use the new format or mode.

If the MODE parameter is omitted, the default mode will be 1600 bpi.

If either the PFMT or SFMT fields are omitted, the corresponding primary or secondary field will become unspecified.

Command: REMOVE rack label

Explanation: This command removes an empty tape from the available tape pool. If the tape currently is in use, the tape is not removed.

Command: STOP

Explanation: This command terminates the program.

Tape Directory Subroutines

The tape directory subroutines are contained in the file SYS:DIRLIB. These are also written in GOM. For information on these subroutines, see Dick Salisbury.

UMNET PCPs and The "C" Data Concentrator"

CC	The "C" Data Concentrator
AB	UMNET PCPs
AD	
AE	
AF	
AN	

Operator Files for PCPs and the "C" Concentrator

*BDL	Used to dump and load UMNET PCPS AB, AD, AE, AF and AN
*DDL	
*EDL	
*FDL	
*NDL	

Startup Files

*DCS, *CCS both start CC

*MNS starts AB, AD, AE, AF and AN using *ABS, *ADS, *AES, *AFS and *ANS

*CLO	Loads but does not dump the "C" Data Concentrator.
*BLO	Loads but does not dump UMNET PCPs AB, AD, AE, AF and AN.
*DLO	
*ELO	
*FLO	
*NLO	

Note: Use *_DL only if a PCP reload is needed due to a power failure, or by a system request; otherwise, use *_LO.

UNATTENDED MODE

From midnight to 8 A.M. on Sunday and Monday mornings, MTS will be available but in "Unattended Mode". During these periods, a Security Clerk (when one is in our employment) will be present to maintain the integrity of the Computing Center but will in no case be responsible for operating or restarting the system.

Users will be made aware of the situation by a broadcast and a SIGNONM message. The signonm message will mention the fact that MTS is in unattended mode and will ask users to $COPY *UAM. The phones' recorded messages will also mention the fact that the system is in unattended mode. During unattended mode, the doors to the Computing Center will be locked and NUBS will be closed.

To display "unattended" mode (UAM), "0"=OFF, "1"=ON:

```
/d uam
```

To enter unattended mode:

```
/m uam 1
```

To leave Unattended mode:

```
/m uam 0
```

NOTE: When entering unattended mode on either system, do the following-

```
/mod uam 1

/mod termprio 1

/mod btchprio 1

$release minimum
```

When the systems are taken out of unattended mode, issue-

```
/mod uam 0

/mod termprio 0

/mod btchprio 0
```

Users will be able to use MTS via remote stations or terminals but will not be able to use parts of the system which require operator intervention, such as mounting magnetic or paper tapes.

When the operations staff is ready to leave on Saturday and Sunday nights, there are a few things to be done as unattended mode is entered.

The operator broadcasts a message telling of the impending unattended state of the computer. At this point, the operator should put in a SIGNONM message. The message should be:

```
       THE SYSTEM IS IN UNATTENDED MODE--USERS, $COPY *UAM
```

Other things to be done by the operator at this time are: $drain the printers and $drain pchx. The printer doors should be left open and the printer gates should be opened, stopping the print train (or chain) motors.

The operator should enter unattended mode by typing /MOD UAM 1 on the operators console. This entry will prevent users at terminals from mounting tapes and will $rerun batch jobs which attempt to mount tapes.

The file *UAM contains the following information:

> During the two, eight hour periods (midnight to 8am on Sunday and Monday mornings) when the system is in "Unattended Mode", there is no operator available, so that any action on the part of the user which requires operator intervention such as mounting tapes, will be rejected. If the mount request eminates from a batch job, the job will be put in the EXECUTE queue in HOLD status until unattended mode ceases and the job priority is accepted. If the system fails for any reason, it will very likely not be restarted unless a systems programmer happens to be doing systems maintenance at the Computing Center. There is also, always the possibility of a shutdown on very short notice.

UNYN Terminals and SCPs, etc.

What is an MTS terminal and where are they?

We have about 50 terminals here at UNYN. Most of these are the Ontels, DECWriters, and Visual 550s located in the user area. But there is also the Ontel in the Consultant's Room, the DPA's terminal on the wrapping table, and the TV jobs status display. These terminals are used to access MTS at the Computing Center on North Campus. There are hundreds of other MTS terminals around campus.

The IBM 360 operator's console is <u>not</u> an MTS terminal.

How are these terminals connected to MTS on North Campus?

We have three Secondary Communications Processors (SCPs) here at UNYN, each of which links some of our terminals to the Center. The SCPs are the black boxes with white front panels located on the table in the back of the machine room. Each has a blue Gandalf modem on top of it which hooks the SCP to a phone line used to transmit the data from the terminals to the Center on North Campus.

Each of our three SCPs is hooked up like this:

```
         ------     ------     ------     ------
        | term |   | term |   | term |   | term |  ...
         ------     ------     ------     ------
           |          |          |          |
           |          |          |          +----------------+
           |          |          +-------------------------+ |
           |          +----------------------------------+ | |
           +-------------------------------------------+ | | |
                                       cables from     | | | |
                                       20 terminals -> | | | |
                                                       | | | |
                                                      ---------
                                                     |         |
       Phone circuit to MTS                -------   |   SCP   |
         on North Campus    <-------------| Modem |--|         |
                                           -------    ---------
```

What should you do when the MTS terminal on the wrapping table won't respond or when the users complain that one or more of the terminals is dead?

1. Figure out if MTS is alive. If either of the printers is actually printing, then MTS is probably OK. If not, you should call the MTS operator on the red phone and find out if MTS is down.

2. If MTS is OK, you should try to test a "dead" terminal as follows. If somebody was already in the midst of using it and was signed on to MTS, do the following:

   On Ontels, press the FIX key.

   On DECWriters, press the DELETE key.

   On Visual 550s, press the DEL key.

   Then type %HELLO and press RETURN. If the terminal and SCP to which it is attached are working properly, you will get something like:

   ```
   11:48:29 03-31-83 S216 LSI-11 MINOS (Version AR293).
   ```

   and you can conclude that nothing is wrong with the terminal or SCP. If the user is still unable to get any response from MTS, call the MTS operator and explain that you have determined that a user's terminal and the SCP to which it is attached are both working OK, but that MTS still won't respond to the user. The MTS operator will then be responsible for checking it out from there.

   If the "dead" terminal was not in use, do the following:

   On Ontels, press the ATTN key.

   On DECWriters, press the RETURN key.

   On Visual 550s, press the RETURN key.

   Within 5 seconds, you should get a response something like:

   ```
   %UMnet:Hermes (S216:MP16:OP/VIS)

   %Which host?
   ```

   Again if you get this, you can conclude that nothing is wrong with the terminal or SCP. You should try typing UM (and RETURN) and see if you get something like:

```
MTS : Ann Arbor (CE49-00232)
#
```

If you don't get this, you should call the MTS operator and explain that you have determined that a user's terminal and the SCP to which it is attached are both working OK, but that MTS won't respond. *Don't do anything else unless the MTS operator or somebody else at the Center tells you to*. Explain to the users that you have reported the problem and somebody at the Center is looking into it.

3. If you don't get any response to the %HELLO, then the terminal or the SCP to which it is attached is probably dead.

   The last page of this writeup has a map of the terminal locations here at UNYN. Look at the map and find out which SCP the "dead" terminal is connected to. This is easily done since the code for each terminal gives the name of the SCP. For example, the consultants terminal is marked as S102-45 which means that it is on SCP S1, connected to port 02 of that SCP, and attached by cable number 45. All you care about is the SCP name, in this case S1.

4. Find another terminal on the same SCP and test it using the directions given above in step 2. If the terminal works, then the SCP is OK and you should try the first terminal again to see if it is still dead. If it is an Ontel, you should try pressing the PROG key while you have the SHIFT and CTRL keys held down. This will reset the Ontel completely. Press ATTN and see if you get any response. If not, power the terminal off and report it as being out of order.

5. If the second terminal doesn't work either, then the SCP is probably dead. You should then look very carefully at the lights on the front of the SCP. Make sure you look at the correct SCP. The lights are difficult to see because they are dim. You should cup your hands above and below the lights to make sure you know which ones are on.

   If either of the LOAD lights is on, then the SCP program is being reloaded from MTS. If these lights don't go off within three minutes, call the MTS operator on the red phone and explain that you have noticed that an SCP seems to be stuck loading. Give the operator the name of the SCP. Don't do anything else unless the operator or someone at the Center tells you to. All the terminals on that particular SCP will be dead until the problem is corrected.

6. If neither of the LOAD lights is on and you previously determined that the SCP wouldn't respond to any of its terminals, then you should reload it by pressing the white INIT switch down and then back up. DO NOT turn any power switches on or off.

   Whenever you reload an SCP, write it in the log book provided.

   One or both LOAD lights should come on within a minute or two and then go off within three minutes. If that doesn't happen, call the MTS operator on the red phone and explain that you determined that an SCP was dead by trying several of its terminals, that you tried to reload it using the INIT switch, but that it won't reload. Give the operator the name of the SCP. Don't do anything else unless the operator or someone at the Center tells you to.

Summary of the above actions

1. Figure out if MTS is alive.

2. If it is, test the "dead" terminal to see if you can get a response from it.

   If it was already in use, press FIX (Ontel), DELETE (DECWriter), or DEL (Visual 550). Then type %HELLO followed by RETURN. If you get a response like:

   11:48:29 03-31-83 S216 LSI-11 MINOS (Version AR293)

   then there is nothing wrong.

   If the terminal was not already in use, try pressing ATTN (Ontel) or RETURN (DECWriter and Visual 550) and see if you get the "UMnet:Hermes" message. If you do, then the terminal and SCP are both OK. Try typing UM (and RETURN) and see if MTS responds with the "MTS : Ann Arbor" message. If it doesn't, call the MTS operator and tell him that you have a working terminal and SCP, but that MTS won't respond.

3. If you don't get any response to the %HELLO, then the terminal or the SCP to which it is attached is probably dead.

   Look at the map on the last page of this writeup and figure out the name of the SCP to which the terminal is attached.

4. Find another terminal on the same SCP and test it using the directions in step 2. If it works, then the SCP is OK and you should try the first terminal again. If it is still dead, power the terminal off and report it as being out of order.

5. If the second terminal doesn't work either, then the SCP is probably dead. You should look at the lights on the front of the SCP. Be absolutely sure you are looking at the correct SCP.

   If either of the LOAD lights is on, the SCP is reloading and all terminals on that SCP will be dead until it is finished.

6. If neither of the LOAD lights is on, but the terminals are still dead, you should reload the SCP by pressing the INIT switch down and then back up.

   Whenever you reload an SCP, write it in the log book provided.

   Reloading should finish within 5 minutes. If it takes longer or doesn't seem to work, call the operator. Don't reload it again.

```
      UNYN station map as of 18:15 25 Mar 1983. (W051:UNYNMAP)
|  Keypunches                       |                              | |
|                                   |                              | |
|                                   |   Consultants                | |
| S104-51                           |           S102-45            |w|
|                                   |             -46 -47 -48 -49  |i|
| S112-50  Decwriters               |                              |n|
|                                |  |                              |d|
| S108-53 S116-52 S101-54 S100-55|  |                              |o|
| ===glass partition ============|  +     -------------------------|w|
| S111-4  S110-3  S109-2 -1 X105-5    S106-42 S107-40 -41 -43 -44  | |
|                                         Express terminals        |w|
| S113-6  S114-7  S115-8                                           |a|
| S117-10 S118-11 S103-9                                           |l|
|                                                                  |l|
| S206-18 S205-17 S203-15 S119-12         S316-37 S317-38 S318-39  | |
| S207-19 S204-16 S202-14 S201-13            Graphics terminals    | |
|                                                                  | |
| S213-25 S210-22 S208-20                                          | |
| S212-24 S211-23 S209-21                     Operator's display   | |
|                                               S200-56            | |
| S214-26 S215-27 S216-28 S217-29                                  | |
| S300-32 S218-31 S219-30 S301-33                                  | |
|                                             User display         | |
| S302-34 S303-35 S304-36                       S319-57            | |
|                                                                  | |
|
|
|
|Nomenclature:
|
|  <SCP><port>-<cable>, so S101-1 is SCP=S1, port=01, cable=1.
|  Just "-<cable>" means there is an unused cable installed. An SCP
|  name beginning with X means there is no terminal connected.
```

```
                              VENDOR INFORMATION        Updated 09-28-10

Company        Position      Name            Phone No.
======================================================================
Amdahl         Manager   Ron Pate         9-1-313-358-4440
                         Home Phone       9-1-313-887-8046
               Sales     Pam Russell      9-1-313-358-4440
               AMDEC                      9-1-800-638-1764
               Service   (Alert Center)   9-1-800-538-8486
               F.E.      Mark Klebba
Site ID=2
----------------------------------------------------------------------
Autologic (Phototypesetter)
               Service   Craig Pemberton  9-312-297-0560
  See also "Litho Service" for Developer
----------------------------------------------------------------------
CalComp
          Sales Daria Gundle              9-1-462-2200
          Supplies       Carolyn Darryberry
                                          9-1-800-225-2667
          Tech.  Support                  0-1-800-225-2667
          Service Mgr.   Tim Kator        9-1-489-0060
          Service                         9-1-800-451-7568
          Ser.  No.s - Plt1 - 2603, Plt2 - 2580, Plt3 - 3778
          Customer Service Number - 50385
----------------------------------------------------------------------
Carlisle Memory Products (Formerly "Graham Magnetics")
               Sales     Steve Howell     9-1-313-963-2380
Tape Cleaner   Service                    9-1-800-433-0916
Serial No. 3319
----------------------------------------------------------------------
Casper Systems Corp. -- See "Spacesaver"
----------------------------------------------------------------------
Canon (See ICS-International Calculator Sales INc.)
----------------------------------------------------------------------
Control Data Corporation
               Manager   Joe Michon       9-1-313-552-6768
               Sales     John Penders     9-1-313-552-6861
               Service                    9-1-313-552-6785
----------------------------------------------------------------------
ComputerVision (Handles Intel Service)
                                          9-1-800-800-7746
 SysId=17511
INTEL          Manager   John Clark       9-1-313-851-8905
               Sales     Larry White      9-1-800-468-3548
               Service   Bob Hunt         9-1-800-468-3548
                    Call service No.  and follow instructions
               Dan Oas (C.E.) Office      9-1-313-851-8905
S/Ns--(3805)FB00 "117", (3825)FB01 "105"
(3825)FB03 "1319", (3805)FB02 "299"
      Site Code 01337AA
----------------------------------------------------------------------
Data South Printers - See "Terminal Services"
```

```
----------------------------------------------------------------------
Data 100  Site Code 671723001
               Sales     Bob Turner       9-1-313-358-5065
               Service or Service Mgr.    9-1-313-968-3860
----------------------------------------------------------------------
Digital Equipment Corp.  (DEC) 11/750s
     West Unit-ID No. 11750 (86068326K) S/N BT16833
     East Unit-ID No. 11750 (82F58688H) S/N BT05367
               Unit Mgr.
                         Mike Bosak       9-995-5600
               Sales     Kent Foster      9-995-5608
               Service                    9-1-800-482-0164
----------------------------------------------------------------------
Emergency Power Engineering (UPS. Also PDU at UNYN)
     N.Y.      Mgr.      George Hamis     9-1-800-438-7373
Sales          Rathsburg Associates (Tom Stuef)
                                          9-1-313-489-1500

               Service   John Drogheo     9-1-800-438-7373
               Information                9-1-800-438-7373

               FAX                        9-1-201-265-1811
----------------------------------------------------------------------
Graham Magnetics: See Carlisle Memory Products
----------------------------------------------------------------------
Hewlett-Packard Plotter (NUBS)
               Contact   Nick Amato       1-800-824-9954
----------------------------------------------------------------------
Hitachi Data Systems (HDS)
          Intergrated Service Control Ctr.
                                          1-800-446-0744
          Sales          C.  A.  Potts    9-1-313-540-2002
          Systems Eng.   T.  Maslowski    9-1-313-540-2002
          Service        Jerry Aten       9-1-800-446-0744 or
                                          9-1-619-458-2046

   4:30pm-8am give phone no., name, problem and location to recording.

          Site Reference Number 21300-25
----------------------------------------------------------------------
IBM            Manager   Bob Hoeft        9-1-313-434-4850
               Sales     Pat McGrew       9-1-313-393-6100 or
                         A. A.  Office    995-5822
        FAX                               9-995-5838
       A. A.  Office IBM                  9-995-5817
       A.A.  Office - Sharon Ellison      995-5829
       A.A.  Office - Pat McGrew          995-5822
       A.A.  Office - Holly Knighton      995-5813
     Acct.  Market Rep.  Marge Farrand    9-1-313-393-6230 or
                                          9-434-4810
               Service (24 Hour)          9-1-800-426-7378

               Service (PCs)              9-1-800-428-2569
**Customer ref.  no.  for ES3090/300J (9021/720) - #5843392
```

```
               Systems   Mike Miotto      9-1-313-393-6100 or
                                          9-1-313-393-6168 or
               Ann Arbor Office           9-995-5825 or
                                          313-665-0816
               Parts                      9-1-313-262-3010
               Supplies                   9-1-800-426-2468
               Distribution Center        9-1-317-658-2900
     Physical Planning   Dave Parker      9-1-517-377-3780 or
                                          9-1-517-377-3700
----------------------------------------------------------------------
ICS (Canon) International Calculator Sales Inc.
                                          994-5079 or
     P.O. C81158                          9-1-358-0140
----------------------------------------------------------------------
INTEL
----------------------------------------------------------------------
Lee Data       Manager   B.  Ernst        9-1-313-647-7840 or

                                          1-708-986-5550
               Service (See ComputerVision)
Customer Code No.  - 80079-001


     ---------------------------------------------------------------------
     Litho Service (Services Developer (Photo Typesetter))
               Service & Sales            9-1-313-593-4333
----------------------------------------------------------------------
Megabit Communications Inc.               9-1-612-481-0921
----------------------------------------------------------------------
MEMOREX   Manager        Daniel N.  Bendtsen
                                          9-1-313-444-2511 X232  or
          Art Carpenter                   1-708-955-1020
               Sales                      9-1-313-354-4511
  Tech.  Support Rep.                     9-1-800-522-0822
          Michiko Nielsen                 1-313-524-8800  X212
               Service   B.  Miller       9-1-800-331-7410
          Site Code 28809150
     P.O.  No.  Tape Drives - G34536   Line Ptr.  - G02426
----------------------------------------------------------------------
Ontels (See Telcom Engineering)
----------------------------------------------------------------------
SORBUS         NUBS Site ID# 1189391-01
               Manager                    9-1-313-591-2225 or
                                          9-1-313-591-1707
               Service                    9-1-313-591-1270

     SORBUS - Services IBM 2040, 1052, 2821-2, 1403-N1 & 2701 at NUBS;
          also, IBM 2022 & 2501 at UNYN
----------------------------------------------------------------------
Spacesaver (Electric Tape Racks) (Casper Systems Corp.)
               General Phone No.          9-1-517-321-4697

               Service   Jerry            1-442-9000
          If no answer, call "general" phone no.  and ask to have Jerry
          paged.
     P.O.  no.  S84100
----------------------------------------------------------------------
StorageTek          Mgr.
                         Warren Kasting   9-1-313-354-2860
               Sales     Jane Whitehead   9-1-313-354-0850
               Service                    9-1-800-525-0369 or
If problem is unusual, etc., have F.E.  call us

     Site Code 44569        Call Collect  9-1-303-673-4056
----------------------------------------------------------------------
TAB            Terminals, etc.
                         D.  Mazzorana    9-1-313-477-5420

Racks                    R.  Smirnoff     9-1-313-477-5420
----------------------------------------------------------------------
Telcom Engineering - Services Ontels & VISUAL 550s
                         Bill Doty        9-1-483-6669
----------------------------------------------------------------------
Terminal Services (Small (DataSouth) printers attached to 9700s)
          Sales Mgr.     Mark Width       9-995-9355
               Sales     Mike McManus     9-995-9355
               Service   Kurt Shelling    9-995-9355
----------------------------------------------------------------------
UMTel (Channel Extenders, Phones, etc.)   763-9138
----------------------------------------------------------------------
VISUAL 550 (See Telcom Engineering)
----------------------------------------------------------------------
Walton Enterprises (Dish - 2.4 M, 1700 W., 120V)
          Mr.  W.B.  Walton               9-1-714-683-0930
----------------------------------------------------------------------
XEROX 9790 S/Ns CNTR 95600-2695 UNYN 95600-1784 NUBS 95600-3203
       Servc.  Mgr.  (4045) D.  Rauser Mgr.
                                          9-996-8700

          Business Office                 9-1-827-3159
          Printing Systems Rep. Debbie Eastwood
                                          9-1-313-827-3159
               Software                   9-1-517-482-2414
               Acct.  Rep.
                         Judy Freedman    9-998-4220
4045 (CNTR) S/N=V31-403615 P.O.=L54271
               Service                    9-1-800-822-2979 or
                                          9-1-356-5300
```

[1] (CalComp After 5pm) Answering service takes message-service is performed the following day. See "Maintenance Agreements".

NUBS TERMINAL STATIONS-(LA36s)

At present, there are twelve Decwriters at NUBS. Seven of these are connected to the Data concentrator and five to the Memorex 1270. Below is a simple diagram showing port and station numbers as well as a position diagram of the terminals.

```
     Stations        10    9    8    7    6    5    4
                    ┌-┼----┼----┼----┼----┼----┼----┼-┐
     MUX Ports       7[1]  6[1] 5[1] 4    3    2    1



                                               C    H
                                               O    A
                                               U    Z
                                          C    N    E
                                          O    S    L
                      O    O              N    E    T
                      P    P              R    L    I
                      E    E              A    O    N
     Stations         N    N    2    3    C    R    E    12   11
                    ┌-┼----┼----┼----┼----┼----┼----┼----┼----┼--┐
     MUX Ports        16   15   14   13   12   11   10   9[1] 8[1]
```

Station Position Diagram

```
        ┌──────────────────────────────────────────────────────
        │
        │     ┌──┐  ┌──┐  ┌──┐  ┌──┐  ┌──┐  ┌──┐  ┌──┐  ┌──┐  ┌──┐
        │     │ 4│  │ 5│  │ 6│  │ 7│  │ 8│  │ 9│  │10│  │11│  │12│
        │     └──┘  └──┘  └──┘  └──┘  └──┘  └──┘  └──┘  └──┘  └──┘
        │
        │
        │
        │  ┌──┐
        │  │ 3│
        │  └──┘                        NORTH -->
        │
        │
     ┌──┐  ┌──┐
     │ 1││  │ 2│
     └──┘│  └──┘
        │
--------┘
```

[1] Interfaced through the Memorex 1270

BUSINESS ADMINISTRATION REMOTE SUB STATION-(BSAD)

```
 COMPUTING CENTER
 -----------------------------------------------------------------
|                                                                 |
|            ---------      ----------                            |
|           | Memorex |    |   208B   |                           |
|<--Channel-|  1270   |----| Data Set |---------------------|     |
|            ---------      ----------                      |     |
|                                                           |     |
|    ---------------                     ------             |     |
|<--| Data          |--(12)-------------| Data |------      |     |
|   | Concentrator  |---                | Set  |      |     |     |
|    ---------------    |                ------       |     |     |
|                       |                             |     |     |
|                     -------            ------       |     |     |
|                    | (4)   |----------| Data |--|   |     |     |
|                    | MUX   |----------|      |  |   |     |     |
|                 ---| (8)   |          | Set  |  |   |     |     |
|                |    -------            ------   |   |     |     |
|                |                                |   |     |     |
|<--Memorex------|                                |   |     |     |
|    1270                                         |   |     |     |
|                                                 |   |     |     |
 ----------------------------------------------------------------- |
                                                  |   |     |     |
                                                  |   |     |     |
 BUS. AD. REMOTE STATION                          |   |     |     |
 ----------------------------------------------------------------- |
|               ROOM 371                          |   |     |     |
|                                                 |   |     |     |
|    24 DecWriters-(LA36s)                        |   |     |     |
|             |                                   |   |     |     |
|         ---------     ------      ------        |   |     |     |
|        |   12    |   |      |----| Data |       |   |     |     |
|        | LA36s   |---| MUX  |----| Set  |-------|   |     |     |
|         ---------     ------      ------            |     |     |
|         ---------                                   |     |     |
|        |   12    |    ------      ------            |     |     |
|        | LA36s   |---|      |----| Data |-----------|     |     |
|         ---------    | MUX  |----| Set  |                 |     |
|                       ------      ------                  |     |
|                                                           |     |
|-----------------------------------------------------------------|
|               ROOM 371-A                                  |     |
|                                                           |     |
|         ----------        ----------                      |     |
|        | DATA 100 |      |   208B   |                     |     |
|        | Computer |------| Data Set |---------------------|     |
|         ----------        ----------                            |
|                                                                 |
|Phones  1-Hot Line                                               |
|        1-U/M Line 763-4286                                      |
|          Data Set 763-4280                                      |
 -----------------------------------------------------------------
```

MAINTENANCE AGREEMENTS Update late 1989

Equipment	Hours	Days per Week
Amdahl Disks	6am-5pm Hrs.	5 Days M-F
CalComp (Plotters)	8am-5pm	5 Days M-F
Casper Systems Corp. -- See "Spacesaver"		
Emergency Power Engineering (UPS) Contract renewed yearly - P.M. every six months	24 Hrs.	7 Days
Graham Magnetics		Time & Materials
Hewlett-Packard Plotter	8am-5pm	5 Days M-F
IBM ES3090/300J	24 Hrs.	7 Days
IBM 3097	24 Hrs.	7 Days
IBM 3092	24 Hrs.	7 Days
IBM 3089	24 Hrs.	7 Days
IBM 3180	24 Hrs.	7 Days
IBM 3864	24 Hrs.	7 Days
IBM 3380	24 Hrs.	7 Days
IBM 3370	24 Hrs.	7 Days
IBM 3880	24 Hrs.	7 Days
IBM 3180	24 Hrs.	7 Days
IBM Note: For some IBM equipment, see SORBUS below.		
IBM 3274s	24 Hrs.	7 Days
IBM 3278s	24 Hrs.	7 Days
IBM 3287	24 Hrs.	7 Days
IBM[5] 2520	See & contact Sorbus	

Lee Data	7am-4pm	5 Days M-F
Memorex Tape Drives	24 Hrs.	7 Days
Memorex Printers	6am-7pm	5 Days M-F
Memorex 1270	7am-6pm	5 Days M-F
ONTELS - All locations	Serviced by staff	
Spacesaver (Electric Tape Racks) (Casper Systems Corp.)	Time & Materials	
TAB	Time and Materials	
TEKTRONIX Terminals	Serviced by staff	
Telcom Engineering (Ontels & VISUAL 550s)		
VISUAL 550	Time & Materials	
XEROX	24 Hrs.	5 Days

[3] Only call for service during the times and days mentioned above. There is no contract so service is on a time and materials basis.

[4] If problems occur with the CalComp plotter outside of the contract hours (8am-5pm), don't call CalComp but rather leave a conspicuous note for the day shift so that they may call them right at 8am.

[5] "On Call" basis--After hours, much more expensive.

[6] "On Call" basis--After hours, much more expensive.

Resource Manager

(Contributed by D. Boettner)

General Comments

1. The RM is a great deal less fragile than HASP. And it rebuilds its data structures from scratch using checkpointed info every time it is started. So when things get really hosed up, the preferred thing to do is to kill it off and then restart it.

2. For the time being, we have both HASP and the RM around. Lines entered from the operator's console, as well as $SY O ... from a terminal, go to HASP if they start with a dollar sign. Eventually they will go to the RM. But for now, you need to $RUN SYS:RMOPR and give it the RM commands. If you are using the operator's console, MTS *RMC will do this for you.

3. The remainder of this document discusses primarily BITNET processing by the RM, as this is the only current production use. More will be added later, dealing with printing, remotes, and the like.

How to tell if something is wrong

1. Use SYSTEMSTATUS to enquire about the RM tasks

```
$SY Tasks RM     (or $SY T R)
```

should print something like (on UM):

```
00032  RM.LOGR 00076AA0   19 Monitor Idle
00033  RM.MAIN 00076D60  361 Monitor Idle
00034   RM.ACC 00076EC0   17 Monitor Idle
00035  RM.CKPT 00077020  128 Monitor Idle
05318  RM.LINK 000AD520   91 Monitor Idle; SA17
02297  RM.LINK 000ADD60   92 Monitor Idle; SA07
```

There are one each of the logger, main, accounting, and checkpoint tasks. The LINK tasks are for the BITNET links, and there is one for each link. Currently, SA07 is the WAYNESTT link to Wayne State, which is our connection to the rest of BITNET. SA17 is UMICHUB which is the link to the UB machine. On UB, the same first four tasks will appear, and there should be one LINK task, with SA00 attached. This is UMICHUM and is the other end of the UM to UB BITNET link.

Currently, the links and their SDA lines are:

on UM: WAYNESTT SA07 link to Wayne State VM machine

			our link to the rest of BITNET
	UMICHUB	SA17	link to UB machine
	MERIT	SA18	link to NSFNET VM machine
	UMDSCVM	SA19	will be link to DSC not yet operational
on UB:	UMICHUM	SA00	link to UM machine

To find out which SA line is attached to which link, look at the file SFIL:DEFINITIONS. This file is read by the RM at startup time and used as commands to modify its symbol table (the source for the symbol table itself is in RMGR:SYMTABLE>SD, but I would not advise studying this if you value your sanity.). Later definitions and redefinitions can be done at any time by feeding these commands to the SYS:RMOPR program. Each link has three processors defined for it, as for example

```
define device=sa07 waynest1               type=link

define device=sa07 wsu$out                type=sysou

define device=sa07 wsu$in par=(stream=any) type=sysin
```

You can also find out this information by displaying the definitions using the SYS:RMOPR program. See next section.

2. Using the SYS:RMOPR program

To enter commands to the RM, run the program in SYS:RMOPR There are two commands used mostly to look at the RM's status, the QUEUE command (shorthand for DISPLAY QUEUE) and the PROCESSOR command (shorthand for DISPLAY PROCESSOR). In addition, JOB (shorthand for DISPLAY JOB) will give information about a specific job. And DISPLAY DEF can be used to find out current definitions of things.

A. QUEUE

If you type QUEUE ALL (or Q ALL, or just Q) it will print one line for each of the 25 (currently) queues, giving the queue name, how many jobs are in that queue, and what the criteria are for assigning jobs to that queue. (Jobs are assigned to queues by going down the list of queues in the order printed and putting a job in the first queue for which all the criteria are met.) If you just want to look at one queue, you can ask by name. Q WAYNESTT will just print the entry for queue WAYNESTT. Outgoing BITNET traffic will be found in queues the same name as the link: WAYNESTT, UMICHUB, ... Incoming BITNET traffic ends up in the queue MESSAGE if it was mail or in the queue IMPORT if it was a file.

The normal (good) case is for all the outgoing queues to have no entries, and for the MESSAGE queue to also have no entries. The IMPORT queue will normally be quite large (over 100) as the jobs that represent incoming file transfers stay there until the user copies *IMPORT* into their own file. Most other queues are currently zero or very small, except for NETPURGE and MSGPURGE, which are usually very large because they are defined with a 48-hour retention period.

The command

```
     QUEUE queuename WITH attribute1 attribute2 ...
```

can be used to show every job in a particular queue along with the value(s) of the specified attribute(s). This is as close as you get in the RM to a columnar-format queue display like HASP. As you will find when you try this, it is not completely satisfactory, but is often useful.

```
     Examples:   QUEUE WAYNESTT WITH DEST

                 QUEUE WAYNESTT WITH JOBNAME

                 QUEUE WAYNESTT WITH DEST SOURCE
```

B. PROCESSOR

Similarly, the PROCESSOR command will print information about active processors. P ALL (or just P) will provide information about all active processors. A particular processor can be looked at by giving its name. For the case of BITNET links, note that there are three processors to worry about (usually) per link, as there are two associated processors (one for input and one for output). For example, P WAYNESTT may yield:

```
        Processor WAYNESTT (529)   is waiting for work.

                  TYPE=LINK LINK=WAYNESTT.
```

Doing P ALL LINK=WAYNESTT would yield perhaps:

```
        Processor WAYNESTT (529)   is waiting for work.

                  TYPE=LINK LINK=WAYNESTT.


        Processor WSU$OUT  (533)   is processing job RSCS03

                  TYPE=SYSOUT LINK=WAYNESTT.
```

```
               Processor WSU$IN   (545)   is waiting for work.

                         TYPE=SYSIN LINK=WAYNESTT.
```

In this case, you should check again after a while to make sure the job number the output processor is handling changes. Otherwise, the link is down or otherwise incapacitated. Note that the main processor for a link will always be waiting for work, since it doesnt do anything but direct the other processors.

C. JOB

The command

```
DISPLAY JOB job-spec

  or  JOB job-spec
```

where job-spec is either the job name or the job number will display information about a specific job in the system. The amount of information presented depends on the verbosity level -- see the following section on Messages from the RM for examples. The command

```
DISPLAY NETINFO job-spec
```

has the same syntax as the DISPLAY JOB command but shows only the source and destination network addresses (at verbosity 1; at verbosity 9 it also gives additional information). It is often more useful than JOB for network jobs.

D. DISPLAY DEF

The current definition of all symbols, processors, and can be displayed at any time. For example,

```
DISPLAY DEF procname
```

will display the processor definition. And

```
DISPLAY DEF ALL LINK=linkname
```

will display all definitions for that given link.

Messages from the RM ==

1. Messages that come out on the operator's console are printed with a header. The header consists of a four-character module identification name, a dot, and a seven-digit error message number. For example,

LNKL.2023210 Link WAYNESTT: disconnected due to line protocol error. For this one, LNKL identifies the message as coming from the LINKLING. The first digit of "2" means it was from the RM layer of things ("1" means the DSP layer), "023" are messages from auxiliary processor tasks, and in the "2023" category, "2xx" messages are BITNET-related.

2. When running SYS:RMOPR, the default verbosity level is 5 so the headers do not come out normally. Enter VERBOSITY 10 to see them.

3. Other not-so-obvious things are linked to the verbosity level. For example, you may have a job that is waiting to print and you want to see how many pages it is. With the default verbosity, doing DISPLAY JOB RM024572 will yield only:

```
DWB.:RM024572 (24572) is awaiting print, P1, after 2 jobs.
```

but if you enter VERBOSITY 10 and then DISPLAY JOB RM024572 you get

```
CIJ .2012002 DWB.:RM024572 (24572) is awaiting  print,  P1,
after 2 jobs.  Entered at 20:55:32 Sun Sep 25/88.
TYPE=PRINT DELIVERY=ARGUS2 HOST=UM PRIORITY=MINIMUM
PROUTE=CNTR TWOSIDED=NO IMAGES=7 PAGES=7 SHEETS=7
PROJ=WDWB PROC=PR98 SOURCE="AF14" USER=DWB.
USERNAME="Don Boettner".
```

(there appears to be no way to just display one attribute of a job; you either get none or all.)

4. A note on job names and job numbers: The job name is a user-selectable eight character name, qualified by the userid. The job number is a six-digit integer which is generated by the RM and always unique. (The job name, even when qualified by the userid, does not have to be unique) When the user does not specify a job name, the RM generates one consisting of the letters "RM" followed by the job number. In the above message, the DWB.:RM024572 is the job name; the number in parentheses following it is the job number. If this had been an *RMBATCH* job (instead of the *RMPRINT* print job that it was), then you might have seen it as "DWB.:RM024572 (24572) awaiting execution" at start, but when it started execution a new job (say 24573) would be generated for the print output. This new job would, however, keep the same job name as the original input job, and so when you saw it awaiting print it would say "DWB.:RM024572 (24573) awaiting print".

What to do if something is wrong ==========================

1. The basic idea is to get rid of the processor for the link that isn't moving, and then start a new one up. This often requires stopping both ends of the link, making sure that both ends have stopped, and then starting each of them up again. For the link from UM to UB you can do both easily, so I usually sign onto both sides, stop things as described below on both sides, and then start things up again on both sides. (The links are named according to the other end, so on UM the link is UMICHUB and on UB the link is UMICHUM) For the other links from UM, I always try first stopping and starting the UM end. If that doesn't work, then I stop the UM end and then contact the NOC (or else have operations contact the NOC) to have them contact the people at the other end (eg, at Wayne State for WAYNESTT) to stop and start their end.

   Note that the UM to UB link just runs between two halves of the ES3090/300J and is not at all within the purview or responsibility of the NOC; local operations should be able to handle it. There is currently a policy debate about whether NOC, since they watch all the other links and notify the appropriate operations staffs when things die, should also look at this link. Keep tuned to your favorite local network to see what today's policy is.

2. Normal procedure for getting rid of things is (using WAYNESTT as an example):

   a. Using SYS:RMOPR or MTS *RMC, enter the RM command

   ```
   DRAIN WAYNESTT
   ```

   b. It should tell you

   ```
   Processor WAYNESTT drained.

   Processor WAYNESTT destroyed.
   ```

   c. Then check (using $SYSTEMSTATUS or equivalent) to make sure that the SAxx line (SA07 for WAYNESTT) has gone. If it tells you that a task named RM.LINK still has SA07 attached, then you need to STOP that task. (If doing this from a terminal, use the KILL verb of SYS:TASKS to avoid having the operators prompted for an ok.) Then wait. It sometimes takes several minutes until it goes away. (Note: this condition seems to occur mostly on the UM to UB link. If you don't make sure the task owning the SAxx line has gone before you do a $START again, then either on that

$START or on the following $DRAIN, the LINK processor will get a program interrupt and request a job dump. (If that happens, cancel the dump, clear everything out and try again.))

3. Then to start things up again,

    a. Issue the RM command

    ```
    START WAYNESTT
    ```

4. It sometimes happens that things get really hosed up, and you need to get rid of the whole RM and restart it. For this case, use the predefined MTS files to drain and restart it so if we add new steps to the startup we don't have to find every last copy of this note and modify it. What you should do is:

    a. Use operators' console (or $SY O ...) and issue

    ```
    MTS *RMD
    ```

    ("D" for Drain)

    b. It should print a message on the operators'console that the RM was drained. If nothing happens, do a /T on the MTS *RMD job and see what it is doing. If it is sitting in "Monitor wait" status, the RM intertask network is probably also hosed up, so proceed to (5) below.

    c. Use the systemstatus command TASKS RM (or /T T RM) to make sure all tasks have gone. (Be patient)

    d. Make sure the messagesystem task connected with all of this has gone. Either look for a job that has POST:NETLOG.BIT locked (should be id POST and *RM.M showing as front of msource name when SY displays it (actual name is *RM.MSG)) or else use SYS:TASKS and enter INTERTASK RMGRNET to find it. If you find it, stop it.

    e. Issue
    ```
    MTS *RMS
    ```
    to start up the system

    f. and then issue

    ```
    MTS *RMP
    ```

    to start up the processors

5. If you try running SYS:RMOPR and it immediately produces three lines of error messages from someone

(the subtasking monitor?) and then sits in "Monitor wait" limbo, the communications path to the RM is also hosed up. (The "Monitor wait" condition is the system equivalent of the "have a happy day" face with "What me worry?" printed under it when you're dealing with programs that use the subtasking monitor)

a. Run SYS:TASKS
   enter the command KILL RM

b. then (still under SYS:TASKS)
   enter the command TASKS RM
   to make sure they're all gone. Wait till they are.

c. then (still under SYS:TASKS)
   enter the command INTERTASK RMGRNET. If it finds any tasks, KILL them. (There will probably be one task, the *RM.MSG task)

d. Now things should be cleaned up and you can restart the RM in the normal way with the operator files:

```
MTS *RMS
MTS *RMP
```

e. Even though the RM has started again, a given link may not be running again (because both ends need to be drained and then both ends started). So check to see if any of the links have nonzero queues, and if so watch to see if the counts go down.

6. If the processor BITMSG keeps reporting that it is "still initializing" then there is a leftover copy with POST:NETLOG.BIT locked still around. Use the systemstatus command to find all MTS jobs with the id POST:

```
SY T U POST
```

You should find two tasks showing an msource name of "*RM.M". The newer task (one with the larger task number) should report FILE WAIT. If this is the case, stop the older task (the one with the smaller task number). This will allow the other to finish initialization.

San Diego Super Computer Consortioum

The University of Michigan ------- SDSC Remote User Access Computer
(SDSC.UMICH.EDU)

Modem (located on second floor)

The smaller satellite dish is connected to this modem via two coaxial cables. There is a larger gray interface cable leading from the modem to the VAX eqiupment on the first floor.

Modem customer service code=MI-011
Customer service phone no.=1-800-221-1617

Note: LED=Light Emitting Diode

If any error indicators are on, call HaL Marshall or Len HJarding. If the "Channel BER" indicator has a number larger than 20, call Hal Marshall or Len Harding. Make a note of the error lights.

DEC RX02

LEDs that should be on are-
"Run","DC On" and "Process"
There is a key that must be used to activate the switches. This key must be in "Local" in order to use the "Boot" switch. Turning this key to the left powers the system down.

VAX 11/750 Location Code=72Q ID=11750 86068326K
Service Phone No. = 1-800-482-0164
"After Hours = 1-800-323-6612

The proper positions for the switches on this box are-
"Boot Device" to "D"
"Power On Action" to "Boot"
"Key" in "LOCAL"
This for "booting" system
LEDs: POWER, RUN, ERROR (dim)

DF112

This is for remote diagnostics and should only be powered if requested. It is unplugged.

TU81

LEDs to be on-
"Logic On"

RA81 disk LEDs:

"RUN/STOP" "READY" "A"

Tube

Console for printer.

DecWrite IV

System Log

Some Commands to be used on the DecWriter are-

Note: The terms enclosed in brackets are the default, that is, if you just press ENTER, that is what will take effect.

```
Username: SYSTEM

Password:
$
((((Use the following command to tell all online users
of system shudown if not using the automatic
shutdown sequence.
$ reply /all "system shutdown in five minutes"
Reply received from user SYSTEM at _XXA1: 10:52:17
system shutdown in five minutes
$ (Press ENTER)
3 terminals have been notified.
((((The following is the command to begin the automatic
shutdown sequence.
$ Sys$system:shutdown

SHUTDOWN -- Perform an Orderly System Shutdown
How many minutes until final shutdown [0]: 10
((((The answer to the following question will go out to
online users.
Reason for shutdown [Standalone]: Preventive
Maintenance (Example)
Do you want to spin down the disk volumes [NO]?  (Press ENTER)
Do you want to invoke the site-specific shutdown procedure
[YES]?
(Press ENTER)
((((The next question should be yes if the system is hung
and the system needs to be back ASAP
Should an automatic system reboot be performed [NO]?  y
or (Press ENTER) if for PM
When will the system be rebooted [shortly via automatic reboot]:
Shutdown options (enter as a comma-separated list):
REBOOT_CHECK Check existence of basic system files
Shutdown options [NONE]: REBOOT CHECK
((((This next option is important.  Must do  Reboot  check.   If
the file
does not exist or is damaged the system cannot do
automatic reboot.
%SHUTDOWN-I-OPERATOR,   This   terminal   is   now   an   operator's
console.
```

SAN DIEGO SUPER COMPUTER

```
%%%%%%%%%%% OPCOM 2-FEB-1988 10:54:17.73 %%%%%%%%%%%
Operator _XXA1: has been enabled, username SYSTEM
%%%%%%%%%%% OPCOM 2-FEB-1988 10:54:17.86 %%%%%%%%%%%
Operator status for operator _XXA1:
CENTRAL,  PRINTER,  TAPES,  DISKS,  DEVICES,   CARDS,   NETWORK,
CLUSTER, SECURITY,
OPER1,  OPER2,  OPER3, OPER4, OPER5, OPER6, OPER7, OPER8, OPER9,
OPER10, OPER11,
OPER12
%SHUTDOWN-I-DISLOGINS, Interactive logins will now be disabled.
%SET-I-INTSET, login interactive limit = 0, current  interactive
value = 2
%SHUTDOWN-I-SHUTNET, The DECnet network will now be shut down.
%%%%%%%%%%% OPCOM 2-FEB-1988 10:54:20.54 %%%%%%%%%%%
Message from user DECNET
DECnet shutting down
%%%%%%%%%%% OPCOM 2-FEB-1988 10:54:21.64 %%%%%%%%%%%
Message from user SYSTEM
%PSI-I-ACCTDISABL, PSI accounting disabled
%%%%%%%%%%% OPCOM 2-FEB-1988 10:54:21.77 %%%%%%%%%%%
Message from user SYSTEM
%PSI-I-STOPPED, VAX PSI V3.2 (Multi-host,Native), network device
_NWA0: dismount
ed
SHUTDOWN message from user SYSTEM at _XXA1: 10:54:24
UMRUAC  will  shut  down  in  5  minutes;  back  up  shortly via
automatic reboot.  Pleas
e log off node UMRUAC.
Standalone
3 terminals have been notified.
SHUTDOWN message from user SYSTEM at _XXA1: 10:56:25
UMRUAC will  shut  down  in  3  minutes;  back  up  shortly  via
automatic reboot.  Pleas
e log off node UMRUAC.
Standalone
2 terminals have been notified.
SHUTDOWN message from user SYSTEM at _XXA1: 10:57:26
UMRUAC  will  shut  down  in  2  minutes;  back  up  shortly via
automatic reboot.  Pleas
e log off node UMRUAC.
Standalone
2 terminals have been notified.
%SHUTDOWN-I-STOPQUEMAN, The queue manager will now be stopped.
%%%%%%%%%%% OPCOM 2-FEB-1988 10:58:27.29 %%%%%%%%%%%
Message from user SYSTEM
%IMPRINT-I-IMPEXIT, imprint symbiont exiting
%%%%%%%%%%% OPCOM 2-FEB-1988 10:58:27.74 %%%%%%%%%%%
Message from user SYSTEM
SDSCnet: NetMgr stopping
%%%%%%%%%%% OPCOM 2-FEB-1988 10:58:29.44 %%%%%%%%%%%
Message from user SYSTEM
SDSCnet: NetMgr stopped
SHUTDOWN message from user SYSTEM at _XXA1: 10:58:30
UMRUAC will shut down in 1 minute; back up shortly via automatic
```

```
reboot.  Please
log off node UMRUAC.
Standalone
3 terminals have been notified.
SHUTDOWN message from user SYSTEM at _XXA1: 10:59:32
UMRUAC will shut down in 0 minutes; back up shortly via
automatic reboot.  Pleas
e log off node UMRUAC.
Standalone
2 terminals have been notified.
%SHUTDOWN-I-SITESHUT,  The site-specific shutdown procedure will
now be invoked.
%%%%%%%%%%% OPCOM 2-FEB-1988 10:59:33.66 %%%%%%%%%%%
Message from user SYSTEM
VAXsim exit status = %VAXSIM-S-SHUTDOWN, shutdown requested by
operator
%SHUTDOWN-I-STOPUSER, All user processes will now be stopped.
%SHUTDOWN-I-REMOVE, All installed images will now be removed.
%SHUTDOWN-I-DISMOUNT, All volumes will now be dismounted.
%%%%%%%%%%% OPCOM 2-FEB-1988 10:59:44.69 %%%%%%%%%%%
Message from user SYSTEM
_XXA1:, UMRUAC shutdown was requested by the operator.
%%%%%%%%%%% OPCOM 2-FEB-1988 10:59:45.26 %%%%%%%%%%%
Logfile was closed by operator _XXA1:
Logfile was SYS$SYSROOT:[SYSMGR]OPERATOR.LOG;184
%%%%%%%%%%% OPCOM 2-FEB-1988 10:59:45.59 %%%%%%%%%%%
Operator _XXA1: has been disabled, username SYSTEM
```

PM Times 1988

Feb 22, Apr 25, Jun 27, Aug 22 Oct 24 and Dec26

SAN DIEGO SUPER COMPUTER

Special Page-Printer Forms

Occassionally, users will request that their ouput be printed on paper of their own choice. This paper, supplied by users, must be the standard 8 1/2 x 11 inches and not be too thick to pass through a page-printer. It's probably a good idea, if the user-supplied forms are of doubtful thickness, to have a test run will a small number of sheets.

When a request is made for special forms use, the following procedures should be used to give the user access to an SP number.

1) Determine the system to be used, UB or UM.

2) Request the following information from the user-

   a. Full name of user.
   b. Signon-id to be used with the special forms job.
   c. The purpose of the print run (optional).
   d. Phone number of person making request.
   e. The name of the department that the user is associated with.
   f. Ask for the final date that this user needs the special forms.

The information gathered from the user making the request is entered into the file which has the special forms number which is to be supplied to them.

Next, do the following to determine which numbers are available. Some numbers are reserved for special ccids.

```
$signon w035        (on either UB or UM)

$f page:f_sp?  fa
```

This last command will show you the permit status (see list below) of each file that you can use. You can $copy, $list or $edit any one of these file to see the contents. In the contents will be information about users including the expiration date for the "permit" status which someone has written in the file.

```
PAGE:F_SP1        U Owner, N Others, U W035

PAGE:F_SP10       U Owner, N Others, U W035, R (K45Q, LGE5)
PAGE:F_SP11       U Owner, N Others, U W035
PAGE:F_SP12       U Owner, N Others, U W035, R SWCE
PAGE:F_SP2        U Owner, N Others, U W035, R TB97
PAGE:F_SP3        U Owner, N Others, U W035
PAGE:F_SP4        U Owner, N Others, U W035, R GC4B
PAGE:F_SP5        U Owner, N Others, U W035
```

```
     PAGE:F_SP6        U Owner, N Others, U W035, RW DWB.
     PAGE:F_SP7        U Owner, N Others, U W035

     PAGE:F_SP8        U Owner, N Others, U W035, R (SS83, SKB7)
     PAGE:F_SP9        U Owner, N Others, U W035
```

As an example, to assign a number to a user, do the following:

```
     $Permit PAGE:F_SP5 READ SXYZ
```

The contents of the file are irrelevant to the user.

When you wish to un-permit the file to the user, do this:

```
     $Permit PAGE:F_SP5 DEFAULT SXYZ
```

Below is, as an example, the contents of file PAGE:F_SP5

```
K064 - Jennie Rowe, 764-0505, CRLT
SCXP - Ray Humphrys, 517-373-1284, MDPH, Aug 84
SHJ6 - Ray Humphrys, MDPH, Nov 1984
LAP3 - Kathy Bazil, Jan.  31, 1985
STQW - Eric Quackenbush, 517-373-9340, State of Michigan, 3/7/85
K7XB - Jim Kulik, CRLT, 764-0505, 6/17/85, leave permitted to
K7xb & k067 for about a year.
SKB7,SS83 - Lewis Yen, Fitness  Res.   Center,  3-2462,  8/15/86
(until 7/1/88)
LALD - Jan Opdyke 28 July 88 - 5 Aug 88
K45Q,LGE5  -  Meryl  Klein, Bus.  Ad.  Finance Research, 6-3528,
8/24/89-9/24/89
```

Notice that a couple of the files have been permitted to more than one ccid.

Note: It is very important that you inform the person making the request for special forms that, on the outside of the package or top-sheet, there appears, in clear and boldly printed letters, 1) the SP number that you've give them (SP5, SP6, etc.) as well as 2) the CCID which will be used with the job which prints the forms.

When users are ready to print the speial-forms job, they should:

```
     $Run    *pagepr    scards=-file     PAR=PAPER=SP5,[optional
     parameters]
```

The PAPER=SP5 parameter may only be used during the running of *pagepr and not as a $CONTROL *PRINT*, $SET or $SIGNON command

Textured letterhead is not recommended as toner doesn't adhere to it very well.

Uninterruptable Power Supply

General Information

The computers and their peripherals, the air conditioning and the lights are running through the UPS system, that is, Edison power flows through the UPS to the equipment. The elevator and the emergency lights remain on a direct commercial power link.

For the time being, if problems occur with the UPS system, call Lee Mitchell or the Electrical shop (4-9444).

> Note: Don't try to fix things yourself!
>
> Do not open covers on the UPS equipment!
>
> If there is a power failure, certain lights will be off; so whenever you decide to go the the "Generater" or "UPS" room, always take a flashlight with you. As you enter the UPS room, just to your left under the light switch, there should be a light in a holder. This light is plugged into a wall socket where it is being charged. You can disconnect this light and use it until the charge runs out. You shouldn't go into either the Generator or UPS room when it is dark. It's too dangerous.

Power Failures

Following a complete power failure, where batterys are completely discharged, power through system 0 will come on automatically. This will bring on the building lights but will not include the lights in the MG and UPS rooms, so take a flashlight downstairs with you.

Following a power failure where batterys are completely discharged, the PB-I MGs on systems I, II and III will have to be restarted manually.

If a power failure occurs:

1) Wait 30 seconds and hope that the power comes back on.

2) If power does not come back on, broadcast (UM & UB) to everyone telling them of the power failure; then give them two minutes to sign off.

3) Run "Shutdown" to keep new people from signing on.

4) If power is still off after 2 1/2 minutes, force people to signon off using "shutdown all".

5) When everyone is off, press "stop" on the CPUs, then wait for the machines to stop.

6 Do a "rs sys" on each system.

*** From this point on, "HURRY". ***

7) Go to the ES3090/300J consoles and power the machine off.

8) Power off the other equipment in the building as well as unneeded lights.

9) Wait for the power to come back on.

10) Call for help to get things restarted.

PB-I

PB-I Status Panel

The Status panel on PB-I consists of four sections.

A. The Alarm section for abnormal functions

1) OVER/UNDERVOLTAGE - The light will illuminate and the audio alarm will sound if the voltage deviates outside the calibrated settings (-10% low, +10% high).

2) PHASE SEQUENCE/LOSS - The light will illuminate and the audio alarm will sound in case one of the following conditions occur-

a. Incorrect phase rotation

b. Loss of any one phase

c. Extreme low voltage on one phase

This also checks for improper phase sequence which could damage hardware.

If there is a loss of, or extereme low voltage on any phase, an alarm is actuated and a shutdown will occur if the condition persists after a short time.

3) LOAD ON BYPASS - Light will illuminate and alarm will sound when the MG is being bypassed.

4) OVERTEMPERATURE - Causes a TRANSFER. Also, light will illuminate and audio alarm will sound if there is high temperature in the MG or if in BYPASS mode, high temperature in the "bypass transformer".

5) OVERLOAD - Causes a transfer. Also, light will illuminate and alarm will sound if "bypass output contactor" or "generator contactor" is overloaded.

B. MIMIC BUS - Mimics input and output power using LEDs in the panel.

C. METERING - Displays "input voltage": A-B, B-C and C-A and A, B and C to neutral. Also shows %LOAD. Calculates maximum output as opposed to current being supplied. etc.

D. CONTROL PANEL and FUSES - includes

a) Motor Stop

b) Motor Start

c) Transfer Initiate - transfers load to "Bypass" (BYP) or "MG" when used in conjunction with the

METERING SECTION

"Transfer Switch"

d) Generator - Time meter, runs when Generator is running. Note the small square in the upper right corner - a red flag should be seen moving

e) Transfer - transfers to "Motor Generator" or "Bypass" (BYP)

f) Emergency Off - For emergencies -> Smoke or fire from Power Bloc or, electrocution

E. Non sync light - Checks generator ouput as opposed to critical bus

F. Fuses--Never pull fuses when Power Bloc is on, in either "Bypass" (BYP) or "generator" mode. Doing so will drop the load.

G. Audio Reset

H. Alarm Test Reset

PB-II - System Status Panel

Alarm Section

A) Input Power Failure - Means commercial power is gone. Becomes active anytime the control logic senses no power or reverse power. At this time, the inverter automatically goes into operation.

B) Battery Breaker Open - Indicates the status of the battery breaker - if on, means batterys not connected to PB-II

C) Battery Discharging - Batteries being used

D) Emergency Power Off - EPO switch has been pulled

E) Fuse Failure - Failure of one or more fuses

F) Over Temperature - Monitors heat sink in inverter - over temperature will cause inverter to shut down

G) Logic Failure - system is continuously monitoring logic. If something is out of sequence etc., the light will come on.

H) Audio Reset - resets horn

METERING SECTION

I) Alarm Test Reset - resets erroneous alarms

Meter Panel

A) DC Bus Voltage - Visual indication of the DC value in volts

B) Charger Amps - Charge to battery during recharge

C) Test Switch - to test system

D) Emergency Power off - Indicates whether or not EPO has been pulled

Batteries

Three Cabinets for each system. ... ...

Powering up Power Blocs

Note: PB-I must be started before PB-II.

PB-II must be powered down before PB-I.

NOTE: Before attempting to power up a PB-I, make sure that the individual "Battery Disconnect" breakers located on each battery cabinet and the main "Battery Disconnect" breaker for each system located on wall boxes are open as well as the "Main Rectifier/Charger" breakers located on the lower front of each PB-II. Then, make sure that on the PB-Is, the "Main Input" breaker, the "Main Output" breaker and the "Output" breaker(s) are open.

Make sure that the "Main Input Switches" located on the blue cabinet on the south wall as well as the "System 0" breaker in the Generater Room are closed.

Powerung up PB-I

1) Close "Main Input" breaker

2) Clear out erroneous alarms (ALARM TEST & AUDIO RESET)

3) Check incoming power by pressing "Input Voltage A-B", "...B-C","... C-A". Results should be 480 volts, plus or minus 10%.

4) Close "Main Circuit Breaker"-(this doesn't do anything as far as the system is concerned).

5) Get output power - press and hold "transfer initiate",

while rotating the "transfer switch" to "Bypass" (BYP). Then release both switches. The "load on Bypass" alarm will be on. Clear this with "alarm test and "audio reset".

6) Check "output volatage" by pressing "Output Voltage A-N", "... B-N", and "... A-N" one at a time while observing the digital panel meter. Values should be 227vac +/- 2v.

7) Get rid of error conditions - use "Alarm Test Reset" and "Audio Reset".

8) To get "critical bus" power, close "Main Output" breaker located on the upper right as you face the controls, etc.

9) Check output power - A-B etc., as in 6) above. This value should be 277vac +/- 2v.

To start MG

1) Press "motor start" and wait from about 90 seconds to 2 minutes for contactors to close.

2) Observe "Generator Run Time Meter". If the red flag (located in the small square at the upper right of this meter) is moving, it's ready to be transferred. Wait for contacters to sound.

3) Press and hold "Transfer Initiate". Rotate "Transfer Switch" to MG; the system will tranfer itself. Hold the "Transfer Switch" until you hear contacters click. This transfers from "Bypass" (BYP) to "MG".

4) Check generator ouput voltage. It should be 277vac +/-2v.

5) Observe "Bypass" light - clear it using the "Audio Test Reset" and "Audio Reset". When MG is up to normal speed, it should be suppling the critical bus with power.

6) Close "Output" breakers located below "Main Output breaker. (System #3 has a "Spare" output breaker which usually stays open).

7) See if you made any mistakes by checking the "status panel".

Powering up PB-II

At this point, PB-II has logic power.

METERING SECTION

1) Clear out erroneous alarms using "Alarm Test Rest" and "Audio Reset"

2) Close "Main Rectifier/Charger" breaker. This starts the "Charger". At this point, the "DC Volt Meter" should show 414V and the AC amp meter should momentarily show 20a. The 20 amp surge shows that the "DC Capacitor Deck" is being charged.

3) Close "Battery Disconnect"s on wall cabinets.

4) Close individual "Battery Disconnect" breakers on battery cabinets. As each cabinet breaker is closed, the "AC Amp Meter" should jump. You can't see this if your'e alone.

5) Clear Alarms using "Audio Alarm Reset" and "Audio Reset".

6) The "DC Volt" meter should show 400+ DC volts.

The entire Power Bloc(s) should now have power.

Powering down Power Bloc system

Power down PB-II first

1) Take batteries offline by opening "Battery Disconnect" breakers on individual cabinets, and then the "Battery Master Disconnect"s on the wall panels. "Battery Breaker Open" alarm will go on. This is normal.

2) Reset "alarm test reset" and "audio reset".

3) Open the "Charger/Rectifier" breaker. The "DC bus" indicator should start dropping very slowly.

4) Reset "Battery Discharge" alarm

Power PB-I down

5) Depress "transfer initiate" and hold. The "non sync lamp" should start to go off and on. However, the off and on states occur very slowly. This light should come on and off about 2 times per minute.

6) After the "non sync light" dims, rotate "transfer initiate" to "Bypass" (BYP). You should hear contactors open. The MG is still running.

7) Reset alarms using "alarm test reset" and "audio reset".

NOTE: * * * * Never press "Motor Stop" unless PB-I is bypassing the Motor Generator (Load on Bypass). * * * *

8) After "Load On Bypass" indicator is on, press "Motor Stop".

UNIX Filesave

UNIX files, specifically, /e, /a and /f need to be saved about once per month and most of the others, every other month.

The UNIX filesave can be run either from the DECwriter near the TU80 on the second floor or from another terminal.

UNIX filesave cannot be run with users on the UNIX system.

A new, unitialized tape can be used for the filesave. Soon, however, older filesave tapes will be re-used.

When asked "Which Host?", use XAAD (from a terminal)

Addr?   use "9999"

login: user types "filesave"

password: user types "dogood" in lower case

source /e filesave/setup

Typing "df" will show the different file systems

Typing "u" will tell which users on are the system.

If any users are signed on, send a message asking them to signoff.

To send a message, type "write name-of-user"

To find out more about who the user is, type "finger name-of user"

An example run follows-

```
login:   type "filesave"
password: dogood   (use lower case)
df    (shows different file systems)
source /e/filesave/setup
dodump 0 /e
ok to do a level 0? y
ok to do backup /e? y
Doing file system consistency check ...
** /dev/rhp1b
**Last mounted on /e
** Phase 1 - Check Blocks and Sizes
** Phase 2 - Check Pathnames
** Phase 3 - Check Connectivity
** Phase 4 - Check Refence Counts
** Phase 5 - Check Cyl Groups
4421 files, 50272 used, 12763 free (219 frags, 1568 blocks,
0.3% fragmentation)
OK to dump Onu /dev/rph1b? y
dump Onu /dev/rhp1b
  DUMP: Date of this level 0 dump: Wed Jan 28 17:51:12 1987
  DUMP: Date of last level 0 dump: the epoch
  DUMP: Dumping /dev/rhp1b (/e) to /dev/rmt8
  DUMP: mapping (Pass I) [regular files]
  DUMP: mapping (Pass II) [directories]
  DUMP: estimated 54144 tape blocks on 1.3D9 tape(s).
  DUMP: dumping (Pass III) [directories]
  DUMP: dumping (PASS IV) [regular files]
  DUMP: 20.93% done, finished in 0:18
  DUMP: 38.75% done, finished in 0:15
  DUMP: 56.24% done, finished in 0:11
  DUMP: Tape rewinding
  DUMP: Change Tapes: Mount tape #2
  DUMP: NEEDS  ATTENTION: Is the new tape mounted and ready
to go?:  ("yes" or "no")
  DUMP: NEEDS ATTENTION: Is the new tape mounted and  ready
to go?:  ("yes" or "no")
  DUMP: NEEDS  ATTENTION: Is the new tape mounted and ready
to go?:  ("yes" or "no") yes
  DUMP: TApe 2 begins with blocks from ino 8435
  DUMP: 70.81% done, finished in 0.10
  DUMP: 89.59% done, finished in 0:03
  DUMP: DUMP: 54144 tape blocks on 2 tape(s)
  DUMP: DUMP IS DONE
  DUMP: lelvel 0 dump on Wed Jan 28 17:51:12 1987
  DUMP: Tape rewinding
remounting file system
%
```

IBM ES3090/300J Water Temp./Flow (17Mar92)

Note: The water flow can influence the temperature of the IBM water even if "customer" water is at ideal temperature.

IBM water	20.9 - 22.5	Celsius
Customer Water	4 - 11	Celsius
Flow Rate	70 - 80	Litres per minute

Usually, if the water temp. is way off, the system will call home. If IBM water temp. is bad, call IBM service after checking customer (chiller) water temp. If customer water temp. is bad, call the A/C people.

Xerox Page Printer Notes

Powering On and Software Loading Procedures

1. Press the LOCAL button on the system controller panel.

2. Press the system POWER ON button on the system controller panel.

3. If the user interface screen is blank, press the ON/OFF toggle switch at the back of the user interface console to the ON position.

   The POWER indicator lights on the system controller and system diagnostics appear on the screen for approximately 30 seconds.

   A "$" and a blinking yellow cursor appear on the display. (If the system is configured so that the user interface is not directly powered by the 9790, only a blinking yellow cursor appears.)

4. Press the CAPS LOCK key so that the red light turns on and enter B for BOOT on the keyboard to load the operating software.

   The warm-up cycle is complete only after operating software loads.

   When loading is complete, the display shows:

```
       LASER PRINTING SYSTEM
  VERSION XXXXXX
       DISK I.D.XXXXXX
       ENTER DATE(MM/DD/YY):
```

5. Enter the date with the numerals such as 02:14:18. You may use a space, a colon or a slash (/) between the month, day and year. Once you have entered the date, by pressing the enter key, the display should show it to you in a written version, e.g., 1/23/86 becomes January 23, 1986.

6. The display will request that the time be entered.

   ENTER TIME (HH:MM:SS):

   • Enter the time usint the 24-hour system, i.e., 2:30 P.M. is 14:30. Both the date and time will now be displayed.

7. Display asks:

ARE THE DATE AND TIME CORRECT AS DISPLAYED (Y/N)?

• Enter Y if correct.

If incorrect, enter N and reenter the date and time correctly.

8. Finally, the screen asks if you want the time/status displayed in the upper right corner of the display:

LOAD AND ENABLE TIME-OF-DAY/STATUS DISPLAY (Y/N)?

In most cases, you should answer yes (Y). If you respond "yes" to this question, the system displays the time and system status in the upper right hand corner of the screen. If, at any time you wish to disable this feature, enter TIM[E] OFF. To re-enable the feature, enter TIM[E] ON.

• If you select "no" at this time, you cannot add the time/status display without re-booting the system.

9. Allow 4 to 7 minutes for the printer to completely warm up.

10. When the printer is ready, a multicolor screen appears on the UI 90 display with printing options. You may select these options with the mounse or use the keyboard to enter commands.

• Refer to the "Xerox UI 90 Operator Guide" for instructions on using the UI 90 screens to operate the 9790.

11. The time and/or date can be changed without re-booting the system. The system must be off-line and idle with no background tasks running and the logon level must be at a class "two" or higher. to make a change, enter:

SET[TIME] MM/DD/YY HH:MM[:SS]

You may enter the time or date or both. Follow the same procedures explained in steps 5 and 6.

If the time-of-day/status display option was selected, when the software was loaded, the time change appears in the upper right corner of the display. SETTIME is available whether or not the time-of-day/status display is enabled. The system records of the date change, and the command line displays the time change.

Removing PR belt

1) Put on plastic gloves

2) Open covers

3) Cam down developer module using Xerox tool

4) Swing out laser shield using screw driver

5) Remove edge exposure lamp at left

6) Remove detrack cortron at top of belt

7) Remove laser lamp at bottom left

8) Raise and latch "C" cortron

9) Remove "drive" and "support shaft" nuts

10) Screw in short extension shaft to the drive roll at upper right

11) Screw in the long extention shaft to the drive belt roll at lower left

12) Slide photo receptor belt module onto extension shaft unitl it latches

13) Wash and polish PR belt

Cleaning the PR belt

1) Use pumice, solvent (43p63) or eraser - eraser is for small scratches

2) Turn belt counter clockwise only using "belt advance crank"

3) Pour small amounts of pumice or solvent on foam pad (white sponges)

4) Apply with light to heavy pressure in small circular motions

5) If cleaning the entire belt, turn crank while applying pumice or solvent until entire surface is covered

6) Lightly buff haze from belt using clean foam pad--you may use "Film Remover" if you like - it dries faster

Replacing PR belt

1) Push PR belt back in machine

2) Unscrew extension shafts - place them back in the machine

3) Replace "drive" and "support shaft" nuts

4) Replace "Laser lamp"

5) Cam up developer module (belt)

6) Replace "Edge Exposure" lamp

7) Replace "Detrack" cortron

8) Lower and latch "C" cortron

9) Let machine warm up

10) Type "offline" on the display on top of the machine

11) Press "Continue"

12) Print 250 samples - "SAM TEST.FRM, 250"

13) Check samples to see if the problem is solved; if it is

14) Return machine to normal operations and run more samples

If this doesn't work, call XEROX