

GENERAL NOTES

MTS Distribution 6.0

April 1988

Note: Installations receiving this distribution of MTS should obtain approval from the University of Michigan Computing Center before distributing any part of the distribution to any installation other than one of those listed at the end of these notes. In addition some parts of the distribution are copyrighted, either by the University of Michigan or by other organizations. Conditions on the use of copyrighted material vary, but distribution or redistribution to installations other than those listed at the end of these notes is often restricted.

In addition to the general information contained in this writeup, there are two other writeups which give more specific information about installing MTS, one for new and one for existing installations. A hardcopy of the appropriate writeup is included with your distribution tapes; both are available on the tapes as components 461/34 and 461/35.

MTS, the **M**ichigan **T**erminal **S**ystem, is distributed using three types of tapes: (1) Dump/Restore tapes for a single-pack MTS system designed to be used as the base system for new installations or for testing and conversion for existing installations, (2) a tape with several utility programs that can be used with the Dump/Restore tapes to build a single-pack system at new installations where no working version of MTS is available, and (3) tapes generated by the MTS *FS program which contain the source, object, command, data, and print files for the system. The Dump/Restore tapes are unlabeled, the utility tape uses standard labels (VOL=MTSUTL), and the *FS tapes are "Volume Label Only" tapes which must be mounted using the keyword LBLTYPE=VLO (VOL=6.0T1, VOL=6.0T2, ...).

Throughout the distribution, reference is made to the components of the distribution. Generally these references consist of a 3- or 4-digit component number, usually followed by a slash and a subcomponent number. For example, the MTS accounting system has been assigned component number 104. However, the accounting system actually has many "pieces" and so it consists of well over 100 subcomponents, beginning with number 104/1. From distribution to distribution, a component will almost always have the same number, but subcomponent numbers may be changed. Thus, for example, something may have been distributed on D5.1D with the number 104/15, while on D6.0 it may be 104/16 (this could happen if a subcomponent containing an update were inserted between source and object).

New component numbers are assigned by each MTS installation in ranges. The number of a component doesn't necessarily indicate which installation is responsible for its maintenance (there is an installation code for that), just which installation originally assigned the number.

The installation assignment ranges are as follows:

Range	Code	Installation Name
0001-0999	UM	University of Michigan
1000-1199	UBC	University of British Columbia
2000-2199	UNE	University of Newcastle upon Tyne
2200-2399	UD	University of Durham
3000-3199	UQV	University of Alberta
4000-4199	WSU	Wayne State University
4200-4399	RPI	Rensselaer Polytechnic Institute
4400-4599	SFU	Simon Fraser University
4800-4999	RIO	CNPq/CBPF/GPD (Rio de Janeiro)
9000-9999	UM	Used for redistributions only

The *FS tapes are generated by the MTS *FS program based on information contained in a data file (known as a driver file), each line of which describes a component (or subcomponent) of the system. Each component has a name, an optional subcomponent name, an installation code, a type code (source, object, MTS commands, etc.), a location (the file name or tape from which it was obtained), a contact person at the installation principally responsible for its maintenance, a locally responsible person (optional), an optional disk name (which gives the file name, if any, on the Dump/Restore pack where this component is also located), and an optional revision level.

If a file name in the disk name field has the string "@E" appended, an empty file is moved to the test pack. This is handy for log files which are usually not included on the *FS tapes, but which must be present on the test pack. File names in the disk name field may also include suffixes of the form "@UM" to indicate that the file is to be included on the UM test pack, but is not normally sent out on the standard distribution test pack.

Revision level "C" denotes subcomponents which have been changed in the UM production system since they were last distributed. Subcomponents with revision level "N" are new components that have never been distributed. Revision level "U" denotes subcomponents that have been distributed since D5.1 (either on D5.1A, D5.1B, D5.1C, D5.1D) but are unchanged since distributed. Subcomponents with no revision level are those that went out on a regular distribution (D5.1) but haven't changed since then.

The driver file editor program, *DEDIT (461/7), may be used to manipulate a driver file. Indexing information is kept in the driver file, which makes the line numbers in the driver file and its associated comment file **very** important. Care should be used when copying or changing the file. Finally, only a D5.1 or later version of *DEDIT should be used to change the D6.0 driver file.

In the distributed driver file, the local persons, when given, are

UM people for non-UM components; other installations should use *DEDIT to fill in its own local names for components assigned to other installations. In this way, a printout may be produced (using the *DEDIT PLIST command) for each programmer, showing the components for which he or she is responsible. This is also a convenient way to inform other installations of who is currently responsible for various components at your installation.

Some of the components in the driver file have the letters "UNSP" in the field that indicates the person who is responsible for the component. The ID UNSP exists on the University of Michigan system to provide a common location from which unsupported programs and subroutines can be made available. Most UNSP software is not actively supported by the University of Michigan Computing Center. This means that there are no guarantees about its reliability, performance, or continued availability. UNSP software has received a minimal amount of testing to insure that it operates correctly for most common cases.

A number of "dummy" driver file entries have been added for components from other MTS installations that are not installed as part of the production system at UM. These entries are simply an aid in assigning component numbers.

As *FS generates the distribution tapes, it adds additional information to each line in the driver file, such as the name of the distribution tape on which it has written the component (tapes are named 6.0Tn), the file number on the distribution tape, a unique name for the component (known as the FS name), information about the file size or tape blocking information if the component was obtained from a regular (non-FS) tape, and the time and date when the file was saved.

Components may be obtained from the *FS tapes using the *FS RESTORE command, either by reference to the FSname or to the file number. In distribution mode, if the RESTORE command is given with the parameters "(m) (n) filename", where **filename** is the name of a driver file which was used to construct the FS tape, that RESTORE command and all subsequent RESTORE commands which have "(m)" or "(m) (n)" as parameters will use as the new name the original name given in the driver file in columns 59-117. For example, to restore a file with the UM name FDEV:MAKEFILE (on tape 6.0T6, file 1), run *FS in distribution mode from the ID FDEV on the tape 6.0T6. Assuming that you have previously restored the D6.0 driver file to file name DIST:DRIVER, the following command will cause the file to be restored to the name specified in the driver file location field:

```
RESTORE (1) (1) DIST:DRIVER
```

Starting with the D5.0 version, *FS generates checksum information as it saves files on the distribution tape. Previous versions of *FS should accept this checksum information even though they don't normally generate it themselves.

Beginning with D5.1, *FS writes tape blocks longer than 4096 characters. However, the D5.1 *FS tapes were generated using an old version of *FS and so do not have long blocks. D6.0 was generated using the D5.1 version of *FS and has long blocks which cannot be read by

pre-D5.1 versions of *FS.

A printed copy of the driver file listing is included in the distribution. Additional copies of this listing may be printed using the LIST command in *DEDIT. The listing is ordered by component number and includes descriptive comments about each component. The driver file and its associated comment file are on the *FS tapes and on the Dump/Restore pack (in the files DIST:DRIVER and DIST:COMM).

A printed copy of the driver file index is also included. It provides an alphabetized list of the components, facilitating use of the driver file listing (which is in numerical order). Additional copies of the index may be printed by copying the appropriate file to *PRINT*. The index is component 461/30 (DIST:INDEX6250 on the Dump/Restore pack).

The following components are new since D5.1D (June 1986):

539, 641, 705, 930-976
1122-1127, 1140-1146, 1177
4259-4264

HIGHLIGHTS OF SIGNIFICANT CHANGES

ACCOUNTING-RELATED ITEMS

The following fields have been added to the accounting record since D5.1:

- accounting management-level ID
- cumulative sums for network, remote message, special forms, and phototypesetter usage, surcharges and royalties full-words
- Julian times changed to clock units and epoch.

The ACCOUNTING subcomponent ACCCONVERT, which calls the ACCCNVRT subroutine, can be used to convert to the current version accounting records.

Accounting management has been changed to use the extended accounting file *ACCOUNTINGX instead of *PROJECT and *PRJDIRECTORY (which are still used). The extended accounting file can be generated with the ACCOUNTING subcomponent LVLCONVERT.

The statistics records have been changed to allow multiple records per signon. The files created and destroyed information has been deleted. special forms, remote messages, surcharges and royalties, and phototypesetter have been added. The units for VMI have been changed from page-milliseconds to page-seconds. The BILLING subcomponent STACNVRT subroutine can be used to convert statistics records to the current version.

TEXTFORM

Only the object of the version of TEXTFORM currently running at UM is included in this distribution. This does not include the POSTSCRIPT OD. Complete source and updates are distributed directly from UQV.

***FS**

The new version of *FS uses variable length blocking on tapes. The current *FS program formats tapes as U(4096), that is, blocks of 4096 or fewer bytes. The new version formats tapes depending on the density of the tape used. The formats are:

- 8 page blocks (U(32767)) for 6250 bpi tapes
- 4 page blocks (U(16384)) for 1600 bpi tapes
- 1 page blocks (U(4096)) for 800 bpi tapes

CATALOG REFORMAT

The file system catalog was changed after D5.1 in two significant ways.

1) The time and date of the last change to a file's data was added to the information kept in the File Descriptor in the catalog. Previously, only the date of the last change (whether that was a catalog change or a data change) was kept. Users can now use \$Filestatus to determine when data last changed: \$Filestatus FILE Lastdatachange

The entire catalog had to be reformatted in order to make this change. The file FILE:FDMOD (482/55) can be used to do the reformatting.

2) A flag was added to the File Descriptor to indicate whether the file was to have a Program Product Charge calculated whenever the file was used. (See the Program Product Charge, component 882.) This change did not require a catalog reformat, but one bit was designated for this function in the File Descriptor.

XA

MTS can run under the 370 and 370-XA architectures. Under 370-XA, each task is limited to an address space of sixteen megabytes, but all real storage on the machine will be used to support virtual storage. MTS does not (yet) run under IBM's ESA-370 architecture.

NAMED ADDRESS SPACES

MTS provides Named Address spaces, a facility similar to IBM's Discontiguous Saved Segment support under VM, to allow tasks to share preloaded programs and data. A program may be loaded into a Named Address Space (or NAS) at system IPL-time and then added to or deleted from a task's virtual storage as needed.

EXPANDED STORAGE

MTS will use expanded storage as a high-speed cache to minimize I to DASDI. The cache is store-through; all writes go immediately to disk. However, disk reads will be satisfied from the cache if possible, thereby cutting down the number of real disk I operations required.

VECTOR FACILITY

User programs running under MTS can make use of IBM's vector facility. MTS saves and restores the vector registers for each task, if necessary, and makes the vector instructions available to a job as part of the standard 370 instruction set.

*MESSAGES and FSM

A change was made to the structure of *MESSAGES. Previously, it was not

possible to have a message number greater than 2,147,482. By reorganizing the format of *MESSAGES, the limit is increased by a factor of 1000 to well over 2 billion. Distribution 6.0 includes a new version of the Message subroutines and utilities which examine *MESSAGES so that they use the new format.

For similar reasons, archive files produced by FSM were constructed in such a way that they could not archive messages numbered beyond 2,147,482. A change was made so that FSM could archive much larger message numbers. FSM now recognizes old files and asks the user to translate such files before allowing access to an archive file.

MTS ENVIRONMENT

1) Command macro processing is now active by default. To disable, one must \$SET MACROS=OFF. The system command macro library, *CMDMACLIB, contains one macro, named MACROLIB, which can be used to easily attach other macro libraries to the MTS environment. To attach a command macro library contained in a file named "CMDMACLIB", issue >MACROLIB CMDMACLIB.

2) Notices of incoming electronic mail are posted on the terminal of the recipient. These notices are called MAILCALLS. This feature can be deactivated by issuing \$SET MAILCALL=OFF.

3) The "execution begins" message and the "execution ends" message now give the same information in interactive and batch mode.

4) Most CLSs (Command Language Subsystems) accept a question mark as a wildcard character in the same way that \$FILESTATUS does. For example, you can now:

```
$DESTROY A?B
$EDIT DOC.?? :ALTER "1987"1988"
$RENAME OLD.?? NEW.??
```

New CLSs were added: \$FILEMENU (component 876), \$MAKE (component 869), \$FTP (File Transfer Protocol - component 4263), \$DUPLICATE (component 855), \$FSM (Full Screen Message - component 1106), \$HELP (component 845).

6) The Editor defaults to PATTERNS=ON now. Use the Editor command SET PATTERNS=OFF to get the old behavior.

7) Some additional \$SET options:

ERRPROMPT=ON/OFF determines whether MTS should prompt for replacement when it encounters an error.

DISPATCH=ON/OFF determines whether immediate messages (dispatches) will be accepted.

MAILCALL=ON/OFF determines whether the user should be notified when a new message arrives in his/her mailbox.

HELPMODE=Line/Screen determines whether information printed from the \$HELP CLS should come out in line mode or whether it should attempt to use full screen mode.

NewFileAccess(NFA)=OFF/<permit-string> determines what access a newly-created file should have. For example:
 \$SET NFA="UNLIMITED W010, READ OTHERS"

FILESAVE

The FileSave process now uses labelled tapes and the Mount subroutine. Changes to *RESTORE were also made to accomodate this change.

CODING CONVENTIONS

The writeup in COPY:CC*PX (1054/6) describes the current MTS Coding Conventions and gives the history of the other conventions previously in use.

All the resident system components which were previously using the old coding conventions have been upgraded to use the current MTS Coding Conventions (most notably, the Disk Manager, the Subtasking Monitor, the Unit Check routines). Most of the post-IPL loaded components which were using the old coding conventions have also been upgraded to use the current conventions (the Resource Manager and all the Plus CLSs for example). Some notable exceptions are the Edit and Messagesystem subroutines.

RESOURCE MANAGER

The distributed MTS system now contains the Resource Manager which is distributed from UBC. The distributed system does not use the Resource Manager as a spooler for printing, but does use it for Bitnet Support.

NETWORK SUPPORT

The distributed MTS system has support for both the Merit network and the UBCNet network. The installation into the distributed MTS system of the Resource Manager, the Message Multiplexor (MM), the DSPDSR, and the DSP layer of the RM provided the hooks for the UBCNet network.

The Mount subroutine, however, is set up to work with the Merit Network, though it has hooks for the UBC HIM. Future plans call for Mount to support both networks; a Mount which works with UBCNet may be obtained from the University of British Columbia, in the meantime.

Note that many components of the Merit Network are included in D6.0. However, the Merit folks did not review all their components so this cannot be considered to be an official distribution of the Merit Network.

HIM SUPPORT

The Host Interface Machine (HIM) has been installed at the University of Michigan. The HIM was developed at the Univeristy of British Columbia. The MTS-side software which supports the HIM was developed jointly by the

Rensselaer Polytechnic Institute and the University of British Columbia. The 6.0 distribution of MTS contains support for the HIM, which includes the HIM I package, the UDP, TCP, and TLNT DSPs, and the FTP CLS and server.

MTS STARTER SYSTEM

The MTS Starter System is comprised of the basic components of a beginning MTS system. Some of the things done for this Starter setup include:

- 1) Special accounting records have been set up for necessary shared user IDs and for a set of 100 private user IDs.
- 2) CKID has been set up for the shared IDs.
- 3) An initial set of Autostart jobs have been scheduled.
- 4) An initial userdirectory with "Systems Group Fire Fighters" has been created.
- 5) A set of tables which can be used to configure a virtual machine under VM for MTS is included.

The DIST:NEWSYS*WF (461/34) writeup describes the Starter System in more detail.

NEW DISK SUPPORT

Changed DASDI to use "Read Device Characteristics" command to determine the size of the disk being formatted. This gives us support for any disk supporting this command including:

3380s (Ds and Es have been tested; Js and Ks should work but have never been tested)

CMS minidisks (untested)

The old FBA disk support has been rejuvenated.

MTS now supports internal 9370 disks (9335s have been tested; we have not tested 9332 support).

EXPANDED STORAGE SUPPORT

UMMPS and CONFIG now support IBM's expanded storage.

The PDP has been modified to use expanded storage as a primary paging device. This was tested but never used in the production system.

The disk manager was modified to use expanded storage (instead of real storage) for the disk cache. TABLES entries can be used to control both the type (XSTORE or REAL) and sizes (min, max, default) of the cache.

DSR CHANGES

MTS no longer has a device list following the DSECT. Many DSRs had to be changed to handle this. The biggest change is that the FDUBLN field now points to an FCB instead of containing the LDN for the device. The LDN is obtained from the FCB as are most of the flags in the old device list.

PPC

MTS now contains a program product charging mechanism (component 882) that allows programs to be "surcharged" based on any GUINFO item. The activation of PPC is based on a bit in the file descriptor and a table of charges (in *PPCHARGES at UM). No changes to the program product are required to utilize this feature.

NETWORK SERVERS

MTS now supports a third connection type - Servers (the other two types are Terminal and Batch). Network connections become Server connections when the DSR returns the server name from WAITFOR.

For PUBLIC servers, this name is looked up in a table (in the file *NETSERVERS (944/3) at UM). The table contains the name of the server command file (\$SOURCED after the project and user sigfiles), the name of the CCID to charge (if any), and a number of flags.

For PRIVATE servers, the system expects a LOGON record containing the above information.

See the new write-up on Servers (944/8) for more information.

NO SEPARATE PAGING AND SPOOL PACKS

Both HASP and the PDP use files on regular disk packs instead of separate packs. This removes the need for TSS DASDI and SAM packs.

MOUNT COMMAND

The D6.0 version of \$MOUNT has minor changes to cope with the new UBC FSUB rewrite, major changes to work with the UNE tape catalog support, and related changes to the format of the PDN table for the tape catalog support. Also, this version has changes related to the No Device List version of MTS. Support for mounting HIM devices and "intertask" pipes has also been added.

STARTER SYSTEM ERRORS

The following files appear on the test pack but shouldn't:

```
*PDP8RTN
FILE:FILERTNS.REA
RSTR:FSRELAB*OA - this is not on the tapes and doesn't work
```

The Driver File says that the following files are on the test pack but they aren't and they don't need to be.

```
ETC:RMQIF
SYS:JSTCMD
SYS:HASPRECEIPTS
UNSP:PAS.SB.LOG
```

The following files appear on the test pack but their contents are outdated. However, the versions on the FS tapes are correct.

```
DIST:GENNOTES*WF (461/33) - the document that you are reading
DIST:OLDSYS*WF (461/35)
```

BUG FIXES

Bugs have been fixed in the following files after the tapes were written:

```
*SIGSETUP
ETC.:CNVTEREP
TRAK
```

The fixes are included in Appendix A. Since this is part of the file DIST:GENNOTES*WF (461/33) on D6.0, they can also be found there. Note that the printed output in Appendix A is not quite right (underscores cause problems, for example, and some lines are truncated) but the file mentioned above contains the correct source for the updates.

DESCRIPTION OF THE DRIVER FILE LISTING

The following is a description of the driver file listing produced by the LIST and PLIST commands in *DEDIT (461/7). A printed copy of the listing is included in the distribution. For each component (or subcomponent), two or three lines of output are printed, followed by any comments associated with the component.

LINE 1

Num - component number and subcomponent number (if any)
R - revision level (if any) of the component or subcomponent:
 C, N, U or a blank for D6.0
Component Name - name of the component
Subname - name (if any) of the subcomponent
Type - component type as follows:

- B - Binary (BTK=TeX-generated DVI file)
- C - MTS Commands
 - CM=Command Macros
 - CML=Command Macro Library
 - CE=Commands Editor
- D - Data
 - DF=\$MAKE Dependency File
 - DQL=Definitions Library for Plus
- H - Help Information
 - HC=CLParser format
 - HF=FSHelp format
- L - Listing (Same extensions as used for Source)
- LE - Linkage Editor commands
- LML - Load Module Library
- M - Messages (program message file)
- O - Object
 - OC=unlinkedited
 - OE=linkedited
 - OL=library
 - OV=MVS load modules in VSS format
- P - Printed Output
 - PL=Line Printer-ready
 - PP=Page Printer-ready
- S - Source code
 - SA=Assembler
 - SAL=Assembler Source Library
 - SB=Web
 - SC=C Language
 - SE=Reduce
 - SF=Fortran
 - SF6=*FTN (FORTRAN 66)
 - SF7=VS/FORTRAN (FORTRAN 77)
 - SG=GOM
 - SI=ICON
 - SJ=AlgolW

SL=LISP
 SN=Cobol
 SP=PL1
 S3=PL360
 SQ=PLUS
 SQL=Plus Source Library
 SR=CLparser Grammar
 S4=Ratfor
 SS=Spitbol
 SS4=Snobol4
 S*=Snostorm
 SW=Pascal
 SX=XPL
 SY=YACC
 SZ=Prolog

U - Update Deck
 UC=*CDUPDATE
 UE=\$EDIT
 UI=*IEBUPDAT
 UU=*UPDATE
 UB=WEB Update

W - Writeup Input
 WF=FORMAT
 WT=TEXT360
 WLK=LaTeX
 WTK=TeX
 WX=TEXTFORM

X - Xerox 9700
 XF=form definition
 XJ=Job Description Library
 XT=Font Definition

G - "Goodness" code
 G=Good, O=OK, S=Shaky (has bugs), B=Bad (needs rewrite)

S - Save control
 Blank means normal
 "#" means not distributed
 "@ " means information incomplete
 "=" means temporary hold
 ">" means very large component

Location - file or tape from which the component was obtained. For tapes, the first parameter is the rack number, the second and third are the volume name (if labeled) and the tape ID (if different from the volume name), followed by an optional parameter "VLO", denoting a volume-label-only tape. Then follow keywords for the label type, blocking format, and DSNAME (if any).

File - the file number if the component was obtained from tape (optional for labeled or *FS tapes).

LINE 2

6250 Tape - name of 6250 BPI distribution tape on which the component was saved
6250 File - *FS file number on the 6250 BPI distribution tape
FS Name - FS name assigned to component as it was saved
Ver - *FS version number assigned to component as it was saved
Ftype - the file type (LINE or SEQ)
LRECL - the maximum record length of the component
Size - the size of the component (in pages if the DevT field is PAGE, in tracks if DISK)
DevT - the device type from which the component was obtained (PAGE for non-FS tapes and files, DISK for items obtained from older (before the page-formatted file system) *FS tapes)
Inst - installation code for the installation responsible for maintenance
Person - the person responsible for the component at the installation given in the SHARE field
Local Per - the person responsible for the component at the local installation
Date and Time - date and time component was saved

LINE 3

1600 Tape - name of 1600 BPI distribution tape on which the component was saved
1600 File - *FS file number on the 1600 BPI distribution tape

NOTE: Currently there are no MTS installations requiring 1600 BPI tapes, so no 1600 BPI version of D6.0 was written when the original version was sent out. A 1600 BPI version will be generated later if it becomes necessary.

Disk Name - name of the file (if any) on the Dump/Restore test pack where the component is located (the save control field controls whether it is also on the *FS tapes). One of four modifiers may be added to this field: @E indicates that the file is to be empty when it is moved to the Dump/Restore pack; @UM indicates that the file is to go be included on the UM Dump/Restore test, but is not normally sent out on the standard distribution pack; @MN indicates the file is a Merit Network file; and @ST indicates that the file is part of the MTS Starter System which includes special accounting files and user documentation, such as *GENDOC, for example.

USE OF DASDI and DISKCOPY

The disk format in use in MTS for the MTS file system is VAMX. It is slightly different than the TSS VAM2 format that was used in the MTS file system for many years or the old VAMX format which was used by several MTS installations when they first began using 3350 type disks. The SAM disk format can be used by HASP for its spool pack(s), though MTS 6.0 HASP has been upgraded to use MTS files as spool extents.

The MTS DASDI and DISKCOPY programs are provided to initialize and restore disks. The MTS DASDI program (component 598) may be used to initialize VAM2 and VAMX format disks. The program uses RDC (read device characteristics) so that it should initialize any VAM2 or VAMX format disks. It has been tested with the following devices: 2301, 3330-I, 3330-II, 7330, 3340, 3344, 3350, 3370, 3375, 3380, 6280, 9332, 9335, and VM minidisks.

The MTS DISKCOPY program (component 724) may be used to dump, restore, or copy VAM2 and VAMX format disks. DISKCOPY can be used on any disk which can be DASDI'd with the following exceptions: 2311 and 2314 disks require proper unit check routines to be developed. Diskcopy does not write track 0 (the label) for FBA devices.

These programs are located in the files FILE:DASDI*OA (598/3) and FILE:DISKCOPY*OA (724/3) and are included on the *FS distribution tapes, the Starter System Dump/Restore tapes, and the utility tape. They require a working version of MTS to run. For existing MTS installations, this should be no problem. New installations must use a special version of MTS that will work without a disk subsystem to run these programs (see the instructions for new installations for details (461/34)).

MTS file system volumes are normally labeled MTS001, MTS002, etc. and must be VAMX (VX) format. The public volume number for the first pack in the system must be 1 and go up by one for each additional pack. No two packs at an installation should have the same volume label **and** the same public volume number. If you are running DASDI or DISKCOPY on your production system, the SLOW option may be used to keep the program from monopolizing the disk system.

The MTS DASDI program will initialize a pack in either VAM2 or VAMX format. Directions for running DASDI are given in the MTS Operator's Manual (592) and in the writeup in FILE:DASDI*WF (682/6).

The following example initializes the pack on D001 as a VAMX pack with volume label MTS501 and public volume 1. The underlined lines are input.

```

Execution Begins
MTS DASDI PROGRAM (VERSION).  ENTER INPUT LINE:
Dddd llllll Vx #/PAGING/PRIVATE pars ...
D001 CURRENTLY LABELED AS "NEW001".  PLEASE CONFIRM.
PAT TO BE WRITTEN ON PAGES X'009178' THRU X'00918A'.
NEXT?

```

Execution Terminated

The MTS DISKCOPY program may be used to copy VAM2 and VAMX disk data from pack to pack as well as to and from tapes. The program will not convert VAM2 format data to VAMX format or vice-versa. It will copy from one disk type to another (3330 to 3350 or 3330-11 to 3330-1 for example) as long as the "to" volume is large enough to hold all of the data and there is sufficient space in the PAT for any relocation entries needed.

Instructions for running DISKCOPY may be found in the writeup in FILE:DISKCOPY*WF (724/4). The following example shows a tape-to-disk restore. The underlined lines are input.

```
$run file:diskcopy*oa+copy:sysdefs prot=off
Execution begins
Enter "FROM" device type (DISK/TAPE):
tape
Enter tape device or pseudo-device names(s):
>T908
Enter "TO" device type (DISK/TAPE):
Enter device name and volume label (Dxxx MTSyyy):
Enter options (SLOW, SWAP, IPL):

Volume copied: 29453 data pages copies, 2 relocations
Enter "FROM" device type (DISK/TAPE):
Execution terminated
```

The SLOW option keeps the program from monopolizing the disk system. The SWAP option causes the volume label on the "TO" device to be replaced with the volume label from the "FROM" device (in a disk-to-disk copy both labels would be changed). The IPL option causes any IPL records to be copied. IPL records are always included on a disk-to-tape copy, but are not normally included on a tape-to-disk or disk-to-disk copy. These IPL records should not be confused with the IPLAREA data used by the IPLREADER, both types of IPL data are needed. More than one tape device may be specified at a time, but this is not required even when the Dump/Restore data spans more than one tape reel.

PATCHING THE SYSTEM

Patching Shared Memory

Find out where the deck to be patched is loaded by checking a current map, by checking the file SEG2:S2MAP, or by using the LOADINFO CLS (enter the MTS command "\$INFO" from a privileged user ID, i.e., an ID that can set PROT=OFF).

Use the System Status Routine (SSRTN) command

```
DISPLAY loc[+disp[+disp ...]] [n]
```

to display memory and the SSRTN command

```
MODIFY loc[+disp[+disp ...]] value[,value ...]
```

to change it. For example

```
DISPLAY 218500+6DBA
```

```
MODIFY 218500+6DBA 47F0
```

These SSRTN commands are legal from the 3270 operator's console when prefixed with a slash (/), as input to the JOBS job which may be run from any operator's console, or as input to the \$SYSTEMSTATUS command when signed on using a privileged MTS user ID. For a complete description of these and the other privileged SSRTN commands see the MTS Operator's Manual (component 592).

SDS or PEEK may also be used to modify shared memory when used from a privileged ID. In addition, the supervisor commands DIS and MOD may be used to display and modify routines loaded into unpagged memory (segment 0), but these commands are not legal when issued from a 3270 operator's console.

The IPLREADER has commands which allow a system to be patched at IPL time. See the IPLREADER description (1021/13) for more information.

Patching the System Object Deck

Using RAMROD (MTS:RAMROD), CREATE a new system from the current system or GET a previously created system which has not yet been made current, but which will become the current system after the patches are made. RMRD:RAMROD*PF (1019/23) contains a description of the use of RAMROD. Enter enough comments so that other people can tell what you've been up to. You will have been prompted for comments if you created a new system, otherwise you may use the ANNOTATE command to add comments.

Add REP cards to the deck(s) using the PATCH command.

RENAME the new system with the correct version name and make it CURRENT.

For example:

```

# $Run mts:ramrod
# Execution begins
  Using file "MTS:ROD"
  Proceed.
~ list current
  UG117 created from UG057 23:06:01 08-11-77
    08-11 23:00 REPLACED UMMPS TO FIX BUG IN GETSTK//SETSTK (MTA)
~ create newsys from ug117
~ Enter comments :
? 08-24 21:05 Just an example (DLB)
?
~ Done.
~ patch taskstat
~ Address Esdid Text <comments> :
? 30 01 58F0 just an example patch (DLB)
?
~ Enter comments :
? 08-24 just an example patch
?
~ REP 000030 0158F0 example patch 21:15:25 08-24-77 W163
~ OK ? ok
~ Done.
~ rename newsys ug247
~ "NEWSYS" is a system.
~ ** RENAME system "NEWSYS" as "UG247" :
~ OK ? ok
~ Done.
~ current
~ There are 3 IPL files with prefix "*IPL.  "
~ Loading system "UG247"
~ LOAD: Resident: 1000-27D80 UMLoad Psect: 58000
  Pageable: 6C000-ACD90 End: AFFFF
  Contents of IPL file "*IPL.2":
  UL177 ENTRY=26E08 PSECT=100008 VIRTUAL=58000...ACD90
  WRITTEN BY ID MTA. AT 06:24:38 07-17-77
  COM 07-14 20:00 CHANGED CONFIG.CARD TO GIVE MORE SPACE TO
    SEGMENT 0 FOR BIG MACHINES
  COM 07-14 21:59 REPLACE TASKSSTAT, MINOR CHANGES.
  COM 07-15 13:35 NEW TABLES TO ADD MORE JOB TABLES.
  COM 07-17 06:20 REPLACED PDP WITH WHAT I HOPE IS THE D4.0
    VERSION.
~ ** Write system "UG247" to IPL file "*IPL.2":
~ OK ? ok
~ System "UG247" has been written to IPL file "*IPL.2"
~ IPL file stacked has been pushed.
~ System "UG247" is now the current system.
~ "UG247" : Released.
~ System "UG247" has been Frozen.
~ Done.
~ stop
# Execution terminated

```

Remember to Produce an Update to the Source

It seems almost silly to mention this, but

Patching Decks Loaded by PISTLE

Decks loaded into shared VM by PISTLE (the post-IPL system loader) can be patched in memory as described above. REP cards are added to the files from which PISTLE loads the object, generally using *OBJUTIL's PATCH command. It is a good idea to use PISTLE without specifying PAR=NOTEST to make sure the patched version will load.

PISTLE can also be used to load complete new versions of a deck into shared VM so long as all references to the module are made using a low core symbol table rather than external references that are already resolved, i.e., if the IPL option is not used to load it. PISTLE allows the automatic replacement of symbols in the low core symbol table LCSYMBOL; other low core symbol tables must be patched by hand.

The PISTLE writeup in UMPS:PISTLE*WP (component 559/14) describes how to use this facility.

PRINTED DOCUMENTATION IN THE DISTRIBUTION

The following lists all of the paper-copy documentation included in D6.0. Of course, all of the documents listed below are available on the *FS tapes except for those marked with an asterisk (*), which indicates that only a paper copy was shipped (no machine readable copy is available). In addition, there are many more writeups on the *FS tapes for which paper copies have not been shipped. Using the MTS editor on the driver file to match for the letters "W" or "P" in column 45 will produce a complete list of all the machine readable documentation.

Printed copies of many items have been omitted from this distribution for existing installations when machine readable versions are available on the *FS tapes or when the material has not changed from that included in the previous distribution of MTS. If your installation needs a printed copy of any of the omitted items, send a request to:

Suzan Alexander
The University of Michigan
Computing Center
535 W. William Street
Ann Arbor, MI 48103
USA

Printed copies of the following items are included with D6.0 for both **new** and **existing** installations.

461/30	Driver file index (6250 BPI ; a listing sorted by component names which serves as an index for 461/15.
461/33	General Notes, what you are reading now
461/34	Installation instructions for New Installations.
461/35	Installation instructions for Existing Installations.
711/1	List of current CCMemos.
* 461/38	UM Machine Configuration Chart.
* 711/15	Computing Center Newsletter (v15n7 through v3n9).
* 711/16	Permission to reproduce Computing Center publications.

Printed copies of the following items are included with D6.0 for new installations only, but are available to existing installations upon request.

004/14	TABLES 370 writeup.
004/16	TABLES XA writeup.
387/44	HASP Operator's Guide.
387/47	HASP Notes.
* 387/92	HASP Remote 360/20 Operator's Guide.
* 387/93	HASP Remote 360/non-20 Operator's Guide.
* 387/94	HASP Remote 1130/1800 Operator's Guide.
* 387/95	HASP Remote System/3 Operator's Guide.
* 387/96	HASP 2780 Remote Workstation Operator's Guide.
* 461/39	TSS DASDI instructions.
468/15	Description of D6.0 System Object Deck.

- 592 MTS Operator's Manual.
- * 673/23 Documentation to supplement the machine readable documentation for the FLECS FORTRAN preprocessor.
- 711/2 List of Computing Center Publications and the publications themselves.
- * 711/17 MTS Reference Summary.
- 1021/13 Documentation for the IPLREADER and friends.
- 1019/23 Documentation for the RAMROD System Maintenance Utility.

The following items are not automatically included with D6.0 of MTS for either existing or new installations, but copies are available upon request.

- * 583/06 Documentation for the KWIC (583) program.
- * 584/04 Documentation for the QUIC (584) program.
- * 646/02 Audio tapes of 33 system lectures given by the UM staff during 1973.

COMPONENTS WHICH DEPEND ON THE SYSTEM CONFIGURATION

1. The TABLES assemblies (468/8-11) depend almost completely on the hardware configuration being used. It is described in separate write-ups (004/14 and /16).
2. There are two disk file backup processes available to save files on tape: the weekly FILE SAVE and the daily online FILE SAVE. There are command files associated with the weekly file save which will have to be changed at each installation so that they indicate correctly which disk volumes are to be saved. See the driver file comments for these components (067).
3. The distributed system pages to disk through the Disk Manager. If it is desired to add 2305 or 4305 paging devices, then INIT:INITCMD (097/5) should be changed to run INIT:FMT2305 (97/36) to initialize 2305s (or 4305s).
4. The PDP (044) assumes by default that paging to disk is done through the Disk Manager. If you intend to use dedicated disks as paging disks, the PDP must be reassembled. If you intend to use more than four 3805 devices or if you intend to use 2305s, you must also reassemble the PDP. The PDP will automatically grab any 3805 that is online and use it when it is started so if it is not to be used, OFFLINE it before the PDP is started (before giving the reason for reloading). The PDP can be reassembled to automatically use any 3330 disks that are labeled with the prefix "PAG" (e.g., PAG001) and that are labeled as PAGING packs. The PDP has also been assembled to assume two 2305s but they will not work properly since the number of slots per device is set to 8. Sites using 2305s will need to reassemble the PDP with NOSLOTS set to 3 (the only number of slots for which the 2305s will work properly). For this reason, all 2305s should be offline before the PDP is started. Note: The PDP has an assembly parameter which determines whether 2301s or 2305s are to be used; the distributed version assumes 2305s. It can be reassembled for 2301s. The PDP can also be reassembled with the 3805 support removed, making it a bit smaller.
5. MOUNT (101) includes support for mounting paper tape readers and punches, audio response units, floppy disks, intertask pipes, HIM devices, and connections on the Merit Computer Network in addition to magnetic tapes. Assembly parameters allow support for these other (non mag tape) devices to be deleted. Support for the Adage Graphics terminal may be included, but UM doesn't have one so this code has never been fully tested.
6. In MTSBATCH (957) the subroutine NEXTJOB will do a binary read from a 2540 or 2501 when it is looking for the next job in a batch stream. Since a 2540 without binary feature ("card image" feature) will accept the command and treat it as an EBCDIC read, NEXTJOB will never find a job. This affects only non-HASP batch (rarely, if ever, used).
7. HASP (387) contains several assembly parameters that depend on the machine configuration. A separate description of these, which wasn't changed for D6.0 and which is somewhat out of date, is

included in component 387/47. Both HASP and the HASPLING (388) have assembly parameters related to RJE support. The HASP master source and distributed object contains no RJE support.

8. TAPERTN (135) - the magnetic tape routines - has several assembly parameters (see comments in the source). In addition, the local system name used in the data set labels generated by MTS is obtained from the CIINAME field in the CNFGINFO table described above.
9. The 2741 Device Support Routine (038) has a built-in table of the device names of hardwired 2703/1270 lines. All other lines are assumed to be dial-up. This table should be updated and *2741RTN (038/3) reassembled as appropriate.
10. The file COPY:SECTIONS*SAL (4215/3) has global set symbols for various hardware and software features. In most cases, this file is copied by system components which are dependent on these features.
11. The 3270 DSR (629) has several global set symbols that may be used to tailor the DSR for use at a given installation.
12. The GRAB3270 table (629/17) has entries for a fixed number of displays. It will work for fewer, but must be reassembled if more displays are allowed to GRAB and FLIP.

LISTINGS IN THE DISTRIBUTION

Listings have been included on the *FS tapes for the following components of the system.

<u>Comp. Name</u>	<u>5.1 Comp. number</u>	<u>5.1 Tape, file #</u>	<u>6.0 Comp. number</u>	<u>6.0 Tape, file #</u>
MTS	0042/4	5.1T1 #22	0042/6	6.0T1 #88
UMMPS MP	0045/4	5.1T1 #30	0045/21	6.0T6 #968
UMMPS 370	0045/4	5.1T1 #30	0045/22	6.0T6 #969
UMMPS 4K	0045/4	5.1T1 #30	0045/23	6.0T6 #970
UMMPS 64	0045/4	5.1T1 #30	0045/24	6.0T6 #971
CONFIG 370	0046/4	5.1T1 #38	0046/5	6.0T1 #117
CONFIG XA	0046/4	5.1T1 #38	0046/6	6.0T1 #118
GUINFO	0198/4	5.1T1 #208	0198/7	6.0T1 #744
LLXU	0354/4	5.1T1 #265	0354/4	6.0T1 #1114
PLIMIT	0464/4	5.1T1 #324	0464/4	6.0T1 #1307
CMDSTAT	0531/4	5.1T1 #550	0919/4	6.0T4 #1426
RSF	0578/4	5.1T1 #636	0578/4	6.0T2 #639
FSUB	0635/4	5.1T1 #759	0635/4	6.0T2 #930
CMD5	0636/3	5.1T1 #762	0636/4	6.0T2 #934
DSRS	0637/4	5.1T1 #765	0637/4	6.0T2 #938
USUB	0638/4	5.1T1 #768	0638/4	6.0T2 #942
DSRI	0639/4	5.1T1 #771	0639/4	6.0T2 #946
TBLS	0829/4	5.1T2 #694	0829/4	6.0T3 #1368
PROFORT	0847/2	5.1T2 #882	0847/2	6.0T4 #208
GATE	4000/4	5.1T3 #317	4000/4	6.0T6 #851

PEOPLE LIST FOR THE UNIVERSITY OF MICHIGAN COMPUTING CENTER

The following is a list of UM Computing Center people whose names appear in the "person" field of the driver file listing. All correspondence should be addressed to:

The University of Michigan
 Computing Center
 535 W. William Sreet
 Ann Arbor, MI 48103
 USA

ALEXANDER	Alexander, Suzan
AUPPERLE	Aupperle, Kathy
BOLLETTINO	Bollettino, Laura (MERIT)
BOSLEY	Bosley, Kevin
BURLING	Burling, Steven R.
B MOORE	Moore, Brian
CASHMAN	Cashman, Brian
CONTO,	Conto, Richard (MERIT)
C MOORE	Moore, Chris
DONNELLY	Donnelly, Stephen M.
DWB	Boettner, Donald W.
D BODWIN	Bodwin, Diane L.
EMERY	Emery, Allan R.
ENGLE	Engle, Charles F.
FLANIGAN	Flanigan, Larry K.
FLOWER	Flower, David S.
FRONCZAK	Fronczak, Edward J.
GLUSKI	Gluski, Kari
GOLD	Gold, Steve
GOODRICH	Goodrich, Andy
HARDING	Harding, Leonard J.
HYDE	Hyde, Dan
J BODWIN	Bodwin, James M.
KNOPPER	Knopper, Mark (MERIT)
LANG	(Language Group - See SWARTZ)
LEVER	Lever, Chuck
LIFT	Lift, Gail H.
MANUAL	(see SALISBURY)
MARTZ	Martz, Paul
MTA	Alexander, Michael T.
MTS	(see BURLING, J BODWIN, BOSLEY, MTA)
PIRKOLA	Pirkola, Gary C.
PLUMMER	Plummer, Chris
RAMANUJAN	Ramanujan, Chitraleka
RIOLO	Riolo, Rick
ROTHWELL	Rothwell, Steven
SALISBURY	Salisbury, Richard A.
SCHNEIDER	Schneider, Eric (MERIT)
SELL	Sell, Jon
SNYDER	Snyder, Dave
SUN	Sun, Dave
SWARTZ	Swartz, Fred G.
TAYLOR	Taylor, Eric

TIFFANY	Tiffany, L. Bernard
TOPOL	Topol, Susan
UNSP	(see SELL)
VALERIO	Valerio, Thomas
WHIPPLE	Whipple, David
WOLFSON	Wolfson, Genie R.
YOUNG	Young, Howard B.

Correspondence related to the administration of the UM Computing Center should be sent to the director:

Dr. Carolyn Autrey-Hunley, Director
The University of Michigan
Computing Center
535 W. William Street
Ann Arbor, MI 48103
USA

Telephone (313) 936-2773

DISTRIBUTION 6.0 INITIAL MAILING LIST

The following is a list of persons to whom the initial shipments of MTS Distribution 6.0 have been sent.

RIO	LCC - CNPQ Attn: Jayme S. P. Goldstein R. Lauro Muller, 455 22290 - Botafoga - Rio de Janeiro R.J. - BRAZIL	6250 BPI
RPI	Office of Computer Services ATTN: Garance Drosehn Rensselaer Polytechnic Institute Troy, NY 12181	6250 BPI
SFU	Computing Centre ATTN: Peter Howard Simon Fraser University Burnaby, B. C. V5A 1S6 CANADA	6250 BPI
UBC	Computing Centre ATTN: Ron Hall 6356 Agricultural Road University of British Columbia Vancouver, B. C. V6T 1W5 CANADA	6250 BPI
UD	Computer Unit Science Laboratories ATTN: Mike Ellison South Road University of Durham Durham DH1 3LE ENGLAND	6250 BPI
UNE	Computing Laboratory ATTN: John Law The University Newcastle upon Tyne NE1 7RU ENGLAND	6250 BPI
UQV	Computing Services ATTN: Garry Jackson The University of Alberta Edmonton, Alberta T6G 2H1 CANADA	6250 BPI
WSU	Computing Services Center ATTN: Tom Stevenson Wayne State University 5925 Woodward Ave., Room 284 Detroit, MI 48202	6250 BPI

Goddard Space Flight Center 6250 BPI
ATTN: Jordan Alpert
Code 616
Greenbelt, MD 20771

Hewlett-Packard CCC
ATTN: Randy Rumbaugh
3000 Hanover Street 20CH
Palo Alto, CA 94304

McGill University Hospitals 6250 BPI
ATTN: Clifford Kuhl
C/O ST Systems
Southfield, MI

MM Computer Laboratory 6250 BPI
ATTN: Rich Wiggins
Michigan State University
East Lansing, MI 48824

APPENDIX A

Bug Fix for *SIGSETUP

. . New (D6.0) base for SIGSETUP*SQ created on Wed Apr 1388. egin sensedatatype. . Inst.: UM. By : Steve Burling. Date : Mon Apr 2588. . sensedatatype:. Getting SenseDataType out of MTS*SQL now, and SnsForTerminal. is bigger than 255 bytes. The result of this is that sigsetup. was providing a 255 byte buffer, but claiming the buffer was longer, and this screwed up MTS, since it would clobber the. following storage, and if \$LOGging was in effect you'd get a. PGNT in LOG. Provide a suitably sized buffer. .

```

y '%Include(ReturnControlBlockType);'
y ' ' %Include(Fill);
y 'definition Primary;' .....elete ' variable SenseInfo
is FixedString,' variable SenseInfo is character(ByteSize(SenseDataType,
Substring(B255,          0,          ByteSize(SenseInfo));'          Fill(SenseInfo,
ByteSize(SenseInfo),          "          ");          end          sensedatatype.
=====
==== Rolled to here at UM on Mon Apr 2588 by Steve Burling ====.
```

Bug Fix for TRAK

. . New (D6.0) base for TRAK*SA created on Thu Apr 1488. sourcemargin=1-72egin CLEARFLG. . Inst.: UM. By : James M. Bodwin (for MTA). Date : Wed Apr 2788. . CLEARFLG: Clear the type field before ORing the new one in.

```

y 'RTHREE USING BCB,R3' NI BCBFLAG,X'FF'-BCBTYPMASK Clear the old
type end CLEARFLG.
=====
==== Rolled to here at UM on Wed Apr 2788 by James M. Bodwin ====.
```

Bug Fix for ETC.:CNVTEREP

. . New (D6.0) base for CNVTEREP*SA created on Fri Apr 2288. sourcemargin=1-72egin MDRType. . Inst.: UM. By : Mike Alexander. Date : Tue Apr 2688. . MDRType: Get the MDR record type code from the device characteristics correc.

```

y ' IF ATYPE,EQ,25 Then look for device char.'
y ' IF R2,LT,R0'

LRBHSW2(1),2+40(R2)      MBR      RECORD      IDENT      end      MDRType.
=====
egin CHPIDV. . Inst.: UM. By : Mike Alexander. Date : Tue Apr
2688. . CHPIDV: The IBM manual had the sense of the "CHPID Valid"
bit backwards, app.
y 'OBRREC MVI HDRTYP,HDRUC0 OBR RECORD' .elete ' OI LRBHSW2,LRBOCPV
CHPID valid'
y ' STC R1,LRBOSCUA' ELSE , OI LRBHSW2,LRBOCPV CHPID not valid
```

```

y 'GENERICOBR MVI HDRTYP,HDRUC0 OBR RECORD' elete ' OI
LRBHSW2,LRBOCPV CHPID valid'
y ' STC R1,LRBOSCUA' ELSE , OI LRBHSW2,LRBOCPV CHPID not valid
y 'OBRTYPE EQU TAPEOBRSVE+4*(R0-R14+16)+3 TYPE CODE IN R0' elete '
OI LRBHSW2,LRBOCPV CHPID valid'
y ' STC R1,LRBOSCUA' ELSE , OI LRBHSW2,LRBOCPV CHPID not valid elete
' OI LRBHSW2,LRBOCPV CHPID valid'
y ' STC R1,LRBOSCUA' ELSE , OI LRBHSW2,LRBOCPV CHPID not valid end
CHPIDV.
=====
==== Rolled to here at UM on Tue Apr 2688 by Mike Alexander ====
. egin
Addr. . Inst.: UM. By : Mike Alexander. Date : Thu Apr 2888. . Addr:
Get some addressibility without using register 13 as a base, since that.
doesn't work. .
y 'CNVTEREP CSECT' .....elete ' USING SA,R13'
y "RDSTTIME SERCOM ' Enter starting time (e.g. 8:20 Jan 13)'" elete
" PMSG ' The part following ',(IN,FLEN=TDLEN),' was ignored'" L R2,=A(IN)
PMSG ' The part following ',(0(R2),FLEN=TDLEN), @ ' was ignored'
y "RDEND SERCOM ' Enter ending time'" ...elete " PMSG ' The part
following ',(IN,FLEN=TDLEN),' was ignored'" L R2,=A(IN) PMSG ' The part
following ',(0(R2),FLEN=TDLEN), @ ' was ignored' elete " MVI IN,C' ' " L
R2,=A(IN) MVI 0(R2),C' '
y 'YORN GUSER IN,LEN,EXIT=END' .....elete ' TR IN(5),0(R1)'
" CLC =C'OK',IN" TR 0(5,R2),0(R1) CLC =C'OK',0(R2) elete " CLI
IN,C'Y'" CLI 0(R2),C'Y' .....elete " CLI IN,C'N'" CLI
0(R2),C'N' .....elete ' USING SA2,R13' .
elete ' USING SA2,R13'
y 'FNDBASE2 LH R0,BASEPATH GET BASE ADDRESS' elete ' TRT
ASENSE+4(1),ONEBIT IS THERE A GOOD ID?' L R2,=A(ONEBIT) TRT
ASENSE+4(1),0(R2) IS THERE A GOOD ID?
y 'BASEOFLO DS X =1 IF TABLE HAS OVERFLOWED' elete "ONEBIT TRTAB
(X'80',X'40',X'20',X'10',X'08',X'04',X'02',X'01',1)"
'*'
y 'TAB3420 EQU *' .....elete ' SPACE 5'
'IN DS XL256 INPUT AREA'
y 'FMTVSTUFF DS 4H'
y ' SPACE 5' IN DS XL256 INPUT AREA SPACE 5
y 'BASETBL DS (BASECNT)XL(BASELEN) TABLE FOR BASE ADDRESSES'
y " " DC X'FE'" * ONEBIT TRTAB
(X'80',X'40',X'20',X'10',X'08',X'04',X'02',X'01',1) end Addr.
=====
==== Rolled to here at UM on Thu Apr 2888 by Mike Alexander ====

```

INSTALLATION INSTRUCTIONS FOR EXISTING INSTALLATIONS

MTS Distribution 6.0

April 1988

In the five years between the MTS/5.1 Distribution and the MTS/6.0 Distribution, a number of incompatible changes have been made to the MTS system internals. (All these changes are upward compatible in the user environment.) This makes updating from a MTS/5.1 system to a MTS/6.0 system more involved than usual because the incompatible changes must be made.

There are two basic ways to consider updating to MTS/6.0. One way is to serially make the required changes to the MTS/5.1 system you are running. The second is to save all the user files currently in use on your system, install the MTS/6.0 system, and restore the files. This latter method is most likely the best for sites which have no local changes to the distributed MTS system and either have relatively few users or use MTS for primarily one purpose (such as conferencing).

The most significant incompatible changes to the resident system are discussed in the General Notes (461/33). They include

- ® The file system catalog format has changed to allow for last data change times.
- ® The accounting record format has changed.
- ® The statistic record format has changed.
- ® The names of system files have changed to conform to new naming conventions.
- ® Components have been changed from using the old Coding Conventions to using the current Coding Conventions.
- ® More MTS commands have been made into Command Language Subsystems (CLS's).
- ® Named Address Spaces are used to load many components.
- ® The Online and Full filesaves now use labelled tapes.
- ® Tapes are now cataloged in an analogous way to files.
- ® HASP no longer needs separate disks for spooling; MTS files can be used as spool files.
- ® The PDP (Paging Drum Processor) no longer needs separate disks for paging; MTS files can be used as paging files.
- ® The *MESSAGES file format has changed to accomodate longer message numbers.

Other system changes (not necessarily incompatible) are described in the comments for the update files (...*Ux files). Changes to the MTS Job Program assemblies are listed in the file MTSS:MTSCHANG*WX (42/10).

A number of initial MTS/6.0 systems are supplied in different IPL files, so you must decide which is appropriate for your site. See the writeup on the Resident System, UMPS:RESSYS*WF (468/15), for descriptions of the following

available IPL files:

```
*IPL.MP      (468/3)
*IPL.370     (468/4)
*IPL.4K      (468/5)
*IPL.64      (468/6)
```

The following discussion describes some basic steps to take in order to save user files, bring up a MTS/6.0 system, and restore the user files onto it. At installations where

where it is not desired that all of this distribution be installed at once, this procedure will have to be modified accordingly. As this is a rather rough outline, please enlist the help of one of the other MTS sites if you have questions or concerns about the upgrade.

These first steps can be done while your production system is still running.

1. Restore FILE:DASDI*OA (598/3) and FILE:DISKCOPY*OA (724/3) to your current MTS system from the MTS/6.0 distribution tapes (the MTS/5.1 versions of these programs might work, but these programs have been upgraded in the 6.0

file index (461/30) for the locations of these components. The index will indicate on which tape each is located and what the file numbers are.

2. Use DASDI to initialize a VAMX disk pack as public volume 1 with a unique volume name; MTS601 might be a good choice. See the General Notes (461/33) for instructions on using DASDI. The following commands illustrate the input to this program:

```
$Run File:Dasdi*oa+Copy:Sysdefs prot=off
Dxxx MTS601 VX 1 IPL SLOW
$ENDFILE
```

where "Dxxx" is the MTS device name of the disk to be initialized and "MTS601" is the volume name to be used for the disk pack.

3. Use DISKCOPY to restore the MTS/6.0 Starter System from the three distributed dump/restore tapes. You should specify the IPL option. You may wish to specify the SLOW option to lessen the interference with your normal operation. See the General Notes for information on running DISKCOPY. The following commands illustrate the input to this program:

```
$Run Diskcopy*oa+Copy:Sysdefs prot=off
TAPE
*1* *2* *3*
DISK
```



```
Dxxx MTS601
IPL SLOW
$ENDFILE
```

where *1*, *2*, *3* are the pseudodevice names associated with the three mounted MTS/6.0 dump/restore tapes.

4. You will need a set of TABLES for your MTS/6.0 Starter System. You can build these on your production system and write them to tape where the Starter System will access them in the next step. The TABLES macros (4/3) should be used to construct the TABLES which will work for the MTS/6.0 Starter System using the devices at your site. You should consult the MTS/6.0 TABLES description (4/4) for details.

Maintaining a set of TABLES using these macros is simpler and much more reliable than using DECKGEN, so we do not recommend using DECKGEN to build TABLES anymore. (In fact, we have not continued to support it so it may not even work.) Be sure to assemble your new TABLES using the 6.0 version of COPY:FILE*SAL (482/45) and COPY:SECTIONS*SAL (4215), in addition to the new TABLES macros. This version of TABLES should only contain one MTS volume, MTS601, though it should contain information on all disk devices. You may want to take a look at the UM TABLES deck (4/1) or the default Starter System TABLES (468/8) to see an example of the form of the TABLES source. In fact, you should probably start with the default tables as a basis for your TABLES, since it is already set up for a one-pack system.

5. Using either a real or a VM guest virtual machine, IPL from the MTS/6.0 Starter System pack and reply "NO" when asked if you want the current system. This time enter the commands:

```
LOAD NAME=*IPL.ttt
REPLACE TABLES.ttt FROM xxx
(Reply to the prompt for a printer address for a map)
```

where xxx is the address of a simulated tape corresponding to the object deck for your assembled TABLES and ttt is the suffix corresponding to the kind of machine architecture you desire, such as "XA" or "370". Finally enter

```
START
```

The MTS/6.0 Starter System should now be running. The following steps should be done on the Starter System.

6. When system initialization finishes (when the message "*NAL complete" appears), you should signon to the userid MTS and \$Run RAMROD (1019), create a new system from the D6.0SYS.ttt system, and replace the deck TABLES.ttt in

that new system with the Starter System TABLES created in step 4. Make this new system current so that you can IPL on the Starter System easily in the future. Instructions on running RAMROD can be found in the writeup in RMRD:RAMROD*WF (1019/22).

7. Use this Starter System pack to get the new version of MTS working to your satisfaction. This will require making any local modifications to the resident system in the RAMROD file in the Starter System. (Remember to change RNBR.) The Starter System contains almost all of the public and semi-public files, but not all of the utility programs you may want. If additional programs are needed, restore them from the MTS/6.0 distribution tapes using *FS.
8. Add any local modules to *LIBRARY. Do not simply replace the new *LIBRARY with your old one, but rather merge the two.
9. Add any local messages to *SMDS. Again, do not just replace the new file with your old one, as there may be new messages in it. Generate a new object module from *SMDS using the utility program (MTSS:GEN_SMDS*OG (197/3)) provided for this purpose and replace the DECK SMDSTBL with the new version using RAMROD.
10. Add any local changes to *SYSMAC. Again, these must be merged with the distributed version.
11. Change the MTS/6.0 SEG2:S2FILES (559/7) and SEG2:NASFILES1 through SEG2:NASFILES7 (891/17 - 891/24) files so that all the components represented in your production system SEG2:S2FILES (and SEG2:NASFILES, if you have that) are accounted for.
12. Build a set of TABLES which will describe your MTS/6.0 production system. Remember that spooling and paging packs are no longer necessary, so your TABLES must specify which volumes the Spooling Extents and Paging Extents will be on.
13. \$Run MTS:RAMROD again, create a new system from the current system and replace TABLES.ttt with the production tables you just made. This is now the system you will load when you come up on your production MTS/6.0 system. **Do not** current this system, but use the Ramrod commands FREEZE, LOAD, and WRITE to write this system to *IPL.PROD.

Now that the basic MTS/6.0 system is ready, you will need to do some special work on your production system in order to finish up the merger. All users must be off the production system for the following steps (except perhaps for the first one).

14. Disable *PRINT* and *BATCH* on your production system and

empty the print and batch queues by printing/running all the spooled jobs. This is necessary since the MTS/6.0 HASP is new and must be started anew. (Actually, this step could be done with users still on the system; they just couldn't use batch or printing facilities). All batch jobs which are under a HOLDUNTIL restraint will have to be cancelled and resubmitted by users on the MTS/6.0 system.

15. Save your current RSTR:FILEDIR.MAS file onto a *FS tape, then empty RSTR:FILEDIR.MAS. Run a special version of the Full Filesave on your production system that saves all files - even those marked as NOSAVE. You can change the current FileSave code to do this by changing the "if Fd_No_File_Save" clauses in the routines Match_File and Write_Save_Info which are in the file RSTR:FS_MAIN#SQ (67/6). The empty RSTR:FILEDIR.MAS file will be filled with the list of all the files saved.
16. In this step you will construct lists of files to be restored later onto your production MTS/6.0 system.

i) Determine which files from the full filesave need to be restored onto the MTS/6.0 system. Go through the new RSTR:FILEDIR.MAS list and find all the user files, licensed products, and files which are **not** distributed as part of MTS/6.0 that you desire on your system. Make a list of those files. Add to that list the files which are site dependent such as *Messages, the Userdirectory database, accounting files, statistic files, autostart schedules, ckid lists, *Forum databases, etc. which you will want your own versions of rather than the Starter System versions of. (You might check for files in the driver file with names of TSTP:? since those are Test Pack/Starter System files.)

ii) Make a second list of files which you use in your production system which **are** distributed as part of MTS/6.0 but are not on the Starter System pack. (The list of MTS/6.0 files already on the Starter System is in the file DIST:MTS6.0.FILES (461/44).) This list will serve as input to *FS to use in conjunction with the MTS/6.0 driver file (461/27) to restore files from the MTS/6.0 distribution tapes. Therefore, the list should be separated by distribution tape and should be of the form

```
Restore (m) dist:6.0driver
Restore (m) (n) dist:6.0driver
```

where "(m)" is a file number and "(m) (n)" is a file number range on one distribution tape of the files you want and "dist:6.0driver" is the distributed driver file.

Save both of these lists on a *FS tape so they can be used later on.

17. \$Run *STA to save your statistics files to tape.

At this point you are ready to get rid of your MTS/5.1 production system. Your MTS/6.0 Starter System should be IPL'd (if it isn't already up) before the following steps are taken.

18. Totally shutdown your production system.
19. You can optionally save a copy of your MTS/5.1 system by copying all the production packs onto tape. Run DISKCOPY from the Starter System to do this. If you only have a few production packs, this shouldn't take too long and might be worth the time (about 20 minutes per pack).
20. Run DASDI and reinitialize all of the production MTS volumes. Make their labels the same as they were previously. Remember that you do not have to initialize any paging or spooling packs anymore.
21. Rebuild the production system catalog by using the CCATL (Create Catalog) program (442). A description of how to do this is in FILE:CCATL*WF (682/3). Be sure to use a version of FILE:FILERTNS which describes the production volumes.
22. The Starter System files now need to be "smeared" or evenly distributed onto the production volumes. Use the list of Starter System files given in DIST:MTS6.0.FILES (461/44) and DIST:FMGEN*O* (461/43) to build a list of files which FileMove (1020) will use to move the files onto the multiple production volumes. You will need to modify the list to specifically put the Paging and Spooling files onto the volumes on which they must reside (consistent with the designation in your production system TABLES assembly). If you have specified more Spooling or Paging extents in your production TABLES than exist on the MTS/6.0 Starter System, you should modify the list to move more of those extents. You must also be certain that the file *IPL.PROD is moved to the file *IPL.0. You may want to explicitly assign files to volumes in order to balance the load.
23. Use FileMove to copy the files onto the production volumes.
24. IPL your multi-volume MTS/6.0 system.

At this point your MTS/6.0 system is ready for the final merger. You should not need the Starter System anymore. The following steps are to be performed on the multi-pack MTS/6.0.

25. Signon to the MTS userid in order to restore files from tape onto the multi-pack MTS/6.0 system.
 - i) \$Run *FS and restore RSTR:Filedir.mas and the lists of desired files made from the MTS/5.1 system.

ii) \$Run *Restore and feed it the list of files (user files, licensed products, site dependent databases, etc.) which are not part of the MTS/6.0 distribution. You should restore your *ACCOUNTING_n files to new names, such as MTS:ACC1, MTS:ACC2, etc. so that they can be reformatted (see below). Be sure to restore *Accounting, *Project, and *Projdirectory.

iii) \$Run *FS and restore the MTS/6.0 driver file to the file DIST:6.0DRIVER. Mount each MTS/6.0 distribution tape and \$run *FS and feed it the list of files to be restored from that tape. (This is the list or lists constructed in step 16ii.)

26. Changes to the accounting records must be made. Restore W013:ACONVERT*OG (104/112), ACC:ACCLIB (104/74), and W013:LVLCNVRT*OG (104/52) from the MTS/6.0 distribution tapes.

i) Convert existing accounting files to MTS/6.0 format. You should do this from a userid which has write access to the Accounting files, such as ACC.

```
$Run Aconvert*og+Acc:ACCLIB scards=MTS:ACC1
      spunch=*ACCOUNTING1 sprint=*dummy*
```

...

```
$Run Aconvert*og+Acc:ACCLIB scards=MTS:ACCn
      spunch=*ACCOUNTINGn sprint=*dummy*
```

ii) Make sure an empty *Accounting_x (sic) file exists. Use the following utility to initiate multi-level accounting:

```
$Run Lvlcnvrt*og+Acc:ACCLIB
```

27. Nothing special needs to be done to accommodate the statistic records format change. The MTS/6.0 programs and utilities which use statistic records have been upgraded to work with MTS/5.1 format (version 0) statistic records through MTS/6.0 format (version 3) statistic records.

28. *Messages must be reformatted. Use the program in MAIL:CONVERT*OQ(798/6) to do this. A description of how to use it is in MAIL:CONVERT*WP (798/7).

The following is a list of public, semi-public, and major private files on the test pack which have been added or changed since the MTS/5.1 distribution. A word of caution is needed here. The following list is based on the revision levels in the driver file. While we have tried to set the revision levels correctly for all components, doing this for a distribution is a big job and we have certainly made some mistakes. Therefore the following list will include some files that haven't changed and some files that have changed will be omitted. We've tried to make the first type of mistake more common than the second.

*ACCINIT	new	*COMPARE	changed
*ACCRESTORE	changed	*CSMP	changed
*ADL	changed	*CSMPEXEC	changed
*ANALYSIS	changed	*CSMPLIB	changed
*ARC	new	*CSMPTRAN	changed
*ASCEBCD	changed	*CTD	changed
*ASMH	changed	*C87	new
*ASSIST	changed	*C87INCLUDE	new
*AUTOREPLACE	new	*C87LIB	new
*AUTOSTART	changed	*DDL	changed
*BDL	changed	*DEDIT	changed
*BLO	changed	*DLO	changed
*BNCHRTN	changed	*DMP	changed
*CBELL	new	*DSD	new
*CBELLINCLUDE	new	*DUMPSYSPGNTS	changed
*CBELLLIB	new	*EBCDASC	changed
*CCMEMOS	changed	*EDL	changed
*CCP	changed	*ELO	changed
*CCPUBLICATIONS	changed	*FILESCAN	changed
*CCQUEUE	changed	*FORTRANVS	new
*CLB	new	*FS	changed
*CLK	changed	*FSC	changed
*CLN	changed	*FSM	changed
*CLPARSEGEN	changed	*FSMCONVERT	new
*CLPARSELIB	changed	*FTPSERV	new
*CLR	changed	*GFINFODSECT	changed
*CLSACC	changed	*HLG	changed
*CLSEEDIT	changed	*HPGLPLOT	changed
*CLSFILEMENU	changed	*IF66	new
*CLSFSMESSAGE	changed	*IF77	new
*CLSHHELP	changed	*IG	changed
*CLSLIST	changed	*IG.SAVE	changed
*CLSMESSAGESYS	changed	*IG.TX4113	changed
*CLSMOUNT	changed	*INIT	changed
*CLSNET	changed	*I8080ASR	changed
*CLSPMF	changed	*LABELS	changed
*CLSSSTA	changed	*LBH	new
*CMD	changed	*LBL	changed
*CMDMACLIB	changed	*LIBRARY	changed
*CMS	changed	*LINK11	changed
*CMSTAPELOAD	new	*LOCALPLOT	changed
*CMSTAPESCAN	new	*LOCALPRINT	changed
*CNFGINFODSECT	changed	*MACBINHEX	new
*COLLATE	new	*MACPAINT4045	new

*MACRTNS	changed	*PTPRRTN	changed
*MCS650XASR	changed	*PTS	changed
*MDP	changed	*PWR	changed
*MESSAGES.LOCK	new	*RATFOR	changed
*MESSSUBS	changed	*REDUCE	changed
*MGR	new	*RESTORE	changed
*MGRSTAT	new	*RG	changed
*MMS	new	*RIP	changed
*MNETRTN.O	changed	*RM.MSG	new
*MNS	changed	*RMS	new
*MSG	changed	*RST	changed
*MSGUTIL	changed	*SAV	changed
*MUP	changed	*SCRIPTPTRFILE	changed
*M6800ASR	changed	*SCRIPT1	changed
*M6809ASR	changed	*SDL	changed
*NAL	changed	*SDLLIB	changed
*NAL1	changed	*SFSAVSUB	changed
*NAL2	changed	*SKEYSUB	changed
*NAL3	changed	*SLIDEQ	new
*NAL4	changed	*SNOSTORM	changed
*NAL5	changed	*SORT	changed
*NAL6	changed	*SORTLM	changed
*NDL	changed	*SPITBOL	changed
*NETCOPYMAC	changed	*SPITLIB	changed
*NETMAILSCHEDULE	changed	*SPRIVSUB	changed
*NETMAILSITES	changed	*SRK	new
*NETSERVERS	new	*STA	changed
*NEWCLSSSTA	new	*STATUS	changed
*NLO	changed	*STE	new
*OVERDRIVE	changed	*SYSMAC	changed
*PASCALJBINCLUDE	changed	*S2P	new
*PASCALJBLIB	changed	*TANGO	new
*PASCALJBSYSLIB	changed	*TANGOLIB	new
*PASCALTIDY	changed	*TAPERRETRIEVE	new
*PASCALVS	changed	*TAPERTN	changed
*PASCALVSINCLUDE	changed	*TAPES	new
*PASCALVSLIB	changed	*TAPESUBMIT	new
*PASCALVSSYSLIB	changed	*TAXIR	changed
*PDP11ASR	changed	*TCM	new
*PDP8RTN	changed	*TEXTFORM	changed
*PERSUB	changed	*TEXT360	changed
*PEXIT	changed	*TNR	changed
*PGF	changed	*TOUCHSUB	changed
*PIL	changed	*TPR	new
*PLOTSEE	changed	*TPS	changed
*PLOTSYS	changed	*TYPEQ	changed
*PLUS	changed	*UDP	new
*PLUS.ENDJUNK	changed	*UDROUTINES	changed
*PLUS.OBJLIB	changed	*UNEDIT	changed
*PLUS.SOURCELIB	changed	*UNIXSTAR	new
*PLUS68	new	*UNLINKER	changed
*PL1F	changed	*USERDIRECTORY	changed
*PL1FLIB	changed	*UTILISP	changed
*PL1LIB	changed	*VALIDATEFILE	changed
*PL1TIDY	changed	*VMP	new
*PROLOGW	new	*VSS	changed

*VSSPDSBUILD	changed	BNCH:BN02	changed
*VSSPDSCOMPRESS	changed	BNCH:BN03	changed
*VSSPDS CONVERT	changed	BNCH:BN04	changed
*XEDIT	new	BNCH:BN05	changed
*XLISP	new	BNCH:BN06	changed
*Z80ASR	changed	BNCH:BN07	changed
*1052RTN	changed	BNCH:BN08	changed
*2501RTN	changed	BNCH:BN09	changed
*2741RTN	changed	BNCH:BN10	changed
*3270RTN	changed	BNCH:BN11	changed
*3284RTN	changed	BNCH:BN12	changed
ACC:ACCDISPLAY	changed	BNCH:BN13	changed
ACC:ACCDISPSHORT	changed	BNCH:BN15	changed
ACC:ACCFILCHARGE	changed	BNCH:BN16	changed
ACC:ACCLIB	changed	BNCH:BN17	changed
ACC:ACCMANT	changed	BNCH:BN30	changed
ACC:ACCRETREIVE	changed	BNCH:BN31	changed
ACC:ACCSAVE	changed	BNCH:BN32	changed
ACC:DISLEVEL	changed	BNCH:BN33	changed
ACC:FILEINFO	changed	BNCH:BN34	changed
ACC:LEVELMAINT	changed	BNCH:BN35	changed
ACC:PROJECTIDS	new	BNCH:BN36	changed
ACC:PROJECTMAINT	changed	BNCH:BN37	changed
APLI:MAKEFILE	new	BNCH:BN38	changed
APLI:MAKEFILE>SD	new	BNCH:BN39	changed
ARC.:ARC*O	new	BNCH:BN40	changed
ARC.:ARC*OE	new	BNCH:BN41	changed
ARC.:MAKE.UNIX	new	BNCH:BN42	changed
ARC.:MAKESIGFILE	new	BNCH:BN43	changed
ARC.:MAKEFILE	new	BNCH:BN44	changed
ASMH:ASMH*SAL	new	BNCH:BN50	changed
ASMH:ASMTIDY.MAKE	new	BNCH:BN51	changed
ASMH:CMDMACLIB	new	BNCH:CONSOLELOG	new
ASMH:MAIN*OA	changed	BNCH:DRIVER	changed
ASMH:MAKEFILE	new	BNCH:DSR*WF	changed
BNCH:ASSIGNVOL#SS	changed	BNCH:LISTFILES#SA	changed
BNCH: BATCH.LOG	new	BNCH:MACROS	changed
BNCH:BM00	changed	BNCH:MIDAS3DATA	new
BNCH:BM01	changed	BNCH:MONITOR*WF	changed
BNCH:BM04	changed	BNCH:MTS.MACROS	new
BNCH:BM05	changed	BNCH:SEG2*OA	changed
BNCH:BM06	changed	BNCH:SEG2#SA	changed
BNCH:BM08	changed	BNCH:STATSUM*OA	changed
BNCH:BM09	changed	BNCH:STATSUM#SA	changed
BNCH:BM11	changed	BNCH:SUBMIT*OF	changed
BNCH:BM12	changed	BNCH:SUBMIT#SF	changed
BNCH:BM13	changed	BNCH:S2FILE	new
BNCH:BM14	changed	BNCH:TAXIR	new
BNCH:BM18	changed	CLIB:CBELL.MAKE	new
BNCH:BM19	changed	CLIB:MAKEFILE	new
BNCH:BM20	changed	CLS:MAKEFILE	new
BNCH:BM21	changed	CMS:MAKEFILE	new
BNCH:BM23	changed	COPY:CLP*SQL	new
BNCH:BM26	changed	COPY:DMGR*SAL	changed
BNCH:BN01	changed	COPY:DSP*SAL	new

COPY:FECP*SAL	new	DMGR:MAKEFILE>SD	new
COPY:FILE*SAL	changed	DSP:MAKEFILE	new
COPY:FILE*SQL	changed	DSP:MAKEFILE>SD	new
COPY:FILE*UQL	changed	DSR.:BNCH*UA	changed
COPY:FILE#SQL	changed	DSR.:BNCH#SA	changed
COPY:HOST*SQL	new	DSR:CMDMACLIB	changed
COPY:INTERNET*SQL	new	DSR:MAKEFILE	new
COPY:I8080*SAL	changed	DSR:SSRTNS.MAKE	new
COPY:MCH*SAL	changed	DSR:3270.MAKE	new
COPY:MCS650X*SAL	new	DUMP:DUMP.CMDS	changed
COPY:MISC*SAL	changed	DUMP:DUMPER*OA	changed
COPY:MNTR*SAL	changed	DUMP:MAKEFILE	new
COPY:MNTR*SQL	changed	DUMP:SNARKLOG	new
COPY:MTS*SAL	changed	DUMP:TASKDUMP	new
COPY:MTS*SQL	changed	DWB:MAKEFILE	new
COPY:M6809*SAL	changed	EDIT:CMDMACLIB	changed
COPY:NETDATA*SQL	new	EDIT:MAKEFILE	changed
COPY:RJE*SAL	new	EDIT:MAKEFILE#SD	new
COPY:RJE*SQL	new	ETC.:\$HELP*HFL	changed
COPY:RMGR*SAL	new	ETC.:APC*H	new
COPY:RMGR*SQL	new	ETC.:FORUM.LOG	new
COPY:SECTIONS*SAL	changed	ETC.:FORUM*HFL	changed
COPY:SL*SQL	changed	ETC.:FS*HF	changed
COPY:STAT*SAL	new	ETC.:FSDISP*HFL	changed
COPY:STOR*SAL	new	ETC.:FSHELPER*HFL	changed
COPY:SUPER*SQL	changed	ETC.:FSMCONV*HC	new
COPY:TABLES*SAL	changed	ETC.:FSMESSAGE*H	new
COPY:TABLESXA*SAL	new	ETC.:FTP*HCL	new
COPY:UC*SAL	changed	ETC.:KERMIT*HFL	new
COPY:UMMPS*SAL	changed	ETC.:KERMIT*LOG	new
COPY:UTIL*SAL	changed	ETC.:LOG.HELP	changed
COPY:VM*SQL	new	ETC.:NET*H	changed
COUN:GENDOC.DOCS	changed	ETC.:RESOLV.CONF	new
COUN:GENDOC.DSTAT	changed	ETC.:RESTORE*HF	new
COUN:GENDOC.GENS	changed	ETC.:TAPES*HCL	new
COUN:GENDOC.NUMS	changed	ETC.:TAPESUB*HFL	new
COUN:GENDOC.ONL.W	changed	ETC.:TAPESUBMIT	new
COUN:GENDOC.OUTPT	changed	ETC.:UNIXTAR*HFL	new
CREP:MAKEFILE	new	ETC:\$ACCHELP	changed
CREP:MAKEFILE>SD	new	ETC:AMDEREP6.0	new
CREP:SDL.MAKE	new	ETC:CNVTEREP	changed
C87:MAKEFILE	new	ETC:CREPHELP*H	changed
DIFF:APC.MAKE	new	ETC:C87ERRORLOG	new
DIFF:COMPARE.MAKE	new	ETC:EREPCMDS	changed
DIST:CMDMACLIB	new	ETC:EREPLIB	changed
DIST:COMM	changed	ETC:FILEMENU*H	changed
DIST:DRIVER	changed	ETC:FINGER*D	new
DIST:FM.GEN	changed	ETC:IF.ELMQQSV	changed
DIST:GENNOTES*WF	changed	ETC:LISTHELP*H	changed
DIST:INDEX	changed	ETC:MACRO.H	changed
DIST:INDEX.GENL	changed	ETC:ONELINER	changed
DIST:MAKEFILE	new	ETC:PASVS*EXPL	changed
DIST:MTS6.0.FILES	changed	ETC:PASVS*MESS	changed
DIST:NEWSYS*WF	changed	ETC:PASVS*OPT	changed
DIST:OLDSYS*WF	changed	ETC:PASVS*TRAN	changed
DMGR:MAKEFILE	changed	ETC:PEEK*H	changed

ETC:RDC3.FAPLIB	changed	FILE:DISK*WF	changed
ETC:RDC3.IMAGE	changed	FILE:DISKCOPY*OA	changed
ETC:RDC3.IMAGE.2	changed	FILE:DISKCOPY*UA	changed
ETC:RDC3.RLISP	changed	FILE:DISKCOPY#SA	changed
ETC:RDC3.SLISP	changed	FILE:DSK*WF	changed
ETC:RDC3.XMPLIB	changed	FILE:DSKMAN*UQ	changed
ETC:RMPLOTALPHA	new	FILE:DSKMAN*UR	new
ETC:RMQIF	new	FILE:DSKMAN#SQ	changed
ETC:SDS*HFL	changed	FILE:DSKMAN#SR	new
ETC:SNOSTORM	changed	FILE:FDMOD*OA	changed
ETC:TAPEFREE	new	FILE:FDMOD*UA	new
ETC:TAPELOGFILE	new	FILE:FDMOD#SA	changed
ETC:TAPERETRIEVE	new	FILE:FILE*UCML	new
ETC:TAXERRORS	changed	FILE:FILE#CML	new
ETC:TXTFENVIR	changed	FILE:FILEMOVE*OQ	new
ETC:TXTFENS	changed	FILE:FILEMOVE*OR	new
ETC:TXTFHYPHDICT	changed	FILE:FILEMOVE*UR	new
ETC:TXTFAYOUTS	changed	FILE:FILEMOVE#SR	new
ETC:TXTFLOG	changed	FILE:FILERTNS	changed
ETC:TXTFMSG	changed	FILE:FINT*OL	changed
ETC:TXTFNUCLEUS	new	FILE:FINT*UA	changed
ETC:TXTFODS	changed	FILE:FINT*UQ	changed
ETC:TXTFWIDS	changed	FILE:FINT#SA	changed
ETC:UD.HELP	changed	FILE:FINT#SQ	changed
FDEV:MAKEFILE	changed	FILE:FIXCAT*OA	changed
FICS:MAKEFILE	new	FILE:FIXCAT*UA	new
FILE:ACATLFAST*OA	changed	FILE:FIXCAT#SA	changed
FILE:AMALCOMP*WF	changed	FILE:FIXEH*UA	new
FILE:CATALOG*OA	changed	FILE:FIXEH#SA	changed
FILE:CATLIST*O	changed	FILE:FIXSD*UA	new
FILE:CATLIST*UQ	changed	FILE:FIXSD#SA	changed
FILE:CATLIST#SQ	changed	FILE:FM*WF	changed
FILE:CATSCAN*OQ	new	FILE:FSTEST*OA	changed
FILE:CATSCAN*UA	new	FILE:FSTEST*UA	changed
FILE:CATSCAN*WF	changed	FILE:FSTEST*WF	changed
FILE:CATSCAN#SA	changed	FILE:FSTEST#SA	changed
FILE:CATSCAN#SQ	new	FILE:GTUNIT	changed
FILE:CCATL*OA	changed	FILE:MAKEFILE	changed
FILE:CCATL*UA	new	FILE:OPEN*OA	changed
FILE:CCATL*WF	changed	FILE:PM*WF	changed
FILE:CCATL#SA	changed	FILE:PPCSET*OA	changed
FILE:CHKFILE*OA	changed	FILE:PPCSET*UA	new
FILE:CHKVTOC*OA	changed	FILE:PPCSET#SA	changed
FILE:CHKVTOC*UA	changed	FILE:RECATALOG*UA	new
FILE:CHKVTOC*WF	changed	FILE:RECATALOG#SA	changed
FILE:CHKVTOC#SA	changed	FILE:STATAREA*OA	changed
FILE:CHONID*OA	changed	FILE:STATAREA*UA	new
FILE:CHONID*UA	new	FILE:STATAREA#SA	changed
FILE:CHONID*WF	changed	FILE:STRPACK*WF	changed
FILE:CHONID#SA	changed	FILE:TABLMOD*WF	changed
FILE:CMDMACLIB	changed	FILE:TABLMDM*OA	changed
FILE:DASDI*OA	changed	FILE:TABLRTN*OA	changed
FILE:DASDI*UA	changed	FILE:TRAK*OA	changed
FILE:DASDI*WF	changed	FILE:TRAKSTAT*OA	changed
FILE:DASDI#SA	changed	FILE:TRAKSTAT#SA	changed
FILE:DISASTER*WF	changed	FILE:TSTABLRTN*OA	changed

FILE:VALIDATEF*UA	changed	HOST:MAKEFILE	new
FILE:VALIDATEF*WF	changed	HOST:NSTEST	new
FILE:VALIDATEF#SA	changed	HOST:NSTEST*HCL	new
FILE:VAMREC*OA	changed	HOST:RESOLVER	new
FILE:VAMREC*UA	changed	HOST:TCMCMD	new
FILE:VAMREC*WF	changed	HOST:TLNTMNGR	new
FILE:VAMREC#SA	changed	HOST:X25*SQL	new
FILE:VNTD*WF	changed	IF66:CMDMACLIB	new
FILE:VNTD#SA	changed	IF66:MAKEFILE	new
FILE:VOLGET*OA	changed	IF77:CMDMACLIB	new
FILE:VTOCUTIL*O	changed	IF77:MAKEFILE	new
FILE:VTOCUTIL*UQ	changed	IG:DDR.MAK	new
FILE:VTOCUTIL*UR	changed	IG:DI.MAK	new
FILE:VTOCUTIL*WF	changed	IG:IAIG.MAK	new
FILE:VTOCUTIL#SQ	changed	IG:MAKEFILE	new
FILE:VTOCUTIL#SR	changed	INIT:ACCINITL	new
FORT:FTNTIDY.MAKE	new	INIT:ACCINIT1	new
FORT:LOGIC.MAKE	new	INIT:ACCINIT2	new
FVS.:MAKEFILE	new	INIT:ACCINIT3	new
GDOC:GENDOC.DBPER	changed	INIT:ACCINIT4	new
GDOC:GENDOC.DOCS	changed	INIT:ACCINIT5	new
GDOC:GENDOC.DSCLO	changed	INIT:ACCINSCN	new
GDOC:GENDOC.GENS	changed	INIT:AMDHIST	new
GDOC:GENDOC.MPER	changed	INIT:CLEAN	changed
GDOC:GENDOC.NUMS	changed	INIT:DATECHK	changed
GDOC:GENDOC.ONL.W	changed	INIT:DATECHK.MAKE	changed
GDOC:GENDOC.OUTPT	changed	INIT:IBMHIST	new
HASP:CMDMACLIB	new	INIT:INITCHK	changed
HASP:CONSOLE	changed	INIT:MAKEFILE	changed
HASP:CONSOLE.D	changed	INIT:WWVCHK	new
HASP:DA	changed	KERB:MAKEFILE	new
HASP:DA.D	changed	LBT:ASA.MAKE	new
HASP:DA*H	changed	LIB:STAFF	changed
HASP:HASP.D	changed	LSPX:XLISP.MAKE	new
HASP:HASP.D*SAL	new	MAC:MAKEFILE	new
HASP:HASP.T	changed	MAIL:CMDMACLIB	changed
HASP:HASP*SAL	changed	MAIL:EXPLAINFILE	changed
HASP:HASPSTAT	changed	MAIL:MAKEFILE	changed
HASP:HASPSTAT.D	changed	MAIL:MCLS.SKELS	changed
HASP:MAKEFILE	new	MAIL:MESSAGEDUP	changed
HASP:QUESCAN	changed	MAIL:MGR	new
HASP:QUESCAN.D	changed	MAIL:MGRSTAT	new
HELP:CMDMACLIB	new	MAIL:MMGR	new
HELP:FSHELPER	changed	MAIL:MMGRSTAT	new
HELP:MAKEFILE	new	MAIL:MSGUTIL*O	changed
HELP:MAKEFILE>SD	new	MAIL:MSUB.SKELS	changed
HOST:CMDMACLIB	new	MAIL:PURGE.LOG	new
HOST:COPY*SQL	new	MCP:TRS13.DOC	changed
HOST:DSPENV*SQL	new	MINI:GENASMUSELOG	changed
HOST:ECHO	new	MINI:MAKEFILE	new
HOST:ECHOSERV	new	MLDB:CMDMACLIB	new
HOST:FTP*SQL	new	MM:MAKEFILE	new
HOST:HIM*SQL	new	MM:SEHLMACLIB	new
HOST:IO*OL	new	MNET:AUTH	changed
HOST:IO*SQL	new	MNOP:CONSOLE(1,999)	changed
HOST:IP.MAKE	new	MNOP:MACLIB	changed

MNOP:11LOAD(1,999)	changed	PCP:ANMINOS	changed
MNTR:DUMP*OA	changed	PFIO:HPGLPLOT.MAK	changed
MNTR:ENVSW*OA	changed	PLOG:MAKEFILE	new
MNTR:GETFREE*OA	changed	PLUS:CLP.MAKE	changed
MNTR:MAKEFILE	changed	PLUS:CMDMACLIB	changed
MNTR:MAKEFILE>SD	new	PLUS:CSL.MAKE	new
MNTR:MNTR*OA	changed	PLUS:LIBGEN	changed
MNTR:ONUNIT*OA	changed	PLUS:LIBGEN.MAKE	new
MNTR:PRVI*OA	changed	PLUS:LIBLIST	changed
MNTR:RESLCSPR*OQ	new	PLUS:MAKEFILE	changed
MNTR:SERVICE*OA	changed	PLUS:MISC*OL	changed
MNTR:SYNCWAIT*OA	changed	PLUS:MISC*SQL	changed
MNTR:UCRTNTAB*OA	changed	PLUS:NEWCCENDJUNK	changed
MNTR:UNITNODE*OA	changed	PLUS:OBJRTNS*OL	changed
MNTR:WAIT*OA	changed	PLUS:OBJRTNS*SQL	changed
MTA:CLEANMSG*O*	changed	PLUS:OBJ68AMI	new
MTA:COPYMAC	changed	PLUS:OBJ68MDS	new
MTA:DMGRSTAT.DIF	new	PLUS:OBJ68MPW	new
MTS.:BROADCAST	new	PLUS:PLUS*PTK	changed
MTS:BUFSTAT*OA	changed	PL1:DWB*CML	new
MTS:CKID.MAKE	changed	PL1:MAKEFILE	new
MTS:CMDMACLIB	changed	PMF:MAKEFILE	new
MTS:CONSOLE	changed	POST:CMDMACLIB	changed
MTS:CREP	changed	POST:MAKEFILE	changed
MTS:LISTSTAT	changed	POST:NMARCHIVE	new
MTS:MAKERULES	new	POST:SCHEDULE	changed
MTS:MAKERULESU	new	POST:SITEBUILD*O	changed
MTS:MAKEFILE	new	PSEE:MAKEFILE	new
MTS:MAKEMACLIB	new	PSYS:MAKEFILE	new
MTS:MTS#CML	changed	RATF:MAKEFILE	new
MTS:PRINTDUMP	changed	REPL:MAKEFILE	new
MTS:TTABLES#SA	changed	RMGR:CMDMACLIB	new
MTSS:CMDMACLIB	changed	RMGR:INSTMAN*WP	new
MTSS:MAKEFILE	changed	RMGR:MAKEFILE	new
NASD:MAKEFILE	changed	RMGR:MAKEFILE>SD	new
NET:CODE.MAKE	new	RMGR:MTSMSGLING	new
NET:CODE*O	new	RMGR:OLG.MAKE	new
OPER:CMDMACLIB	new	RMGR:OPR*WP	new
PAGE:CCTDEFS#SD	new	RMGR:RFC82X*SQL	new
PAGE:CCTBUILD	new	RMRD:MAKEFILE	new
PAGE:CMDMACLIB	changed	RMRD:RAMROD*PF	changed
PAGE:FNTCMDMAC	changed	RSTR:AUTOREST*OA	changed
PAGE:FRMBUILD	new	RSTR:DEADSAVE*OA	changed
PAGE:M.GR*DQL	changed	RSTR:FASTREST*OA	changed
PAGE:M.PAGEPR*O	changed	RSTR:FASTRSTR*OA	changed
PAGE:M.PAGEPR*SQL	changed	RSTR:FS*SAL	changed
PAGE:MAKEFILE	new	RSTR:FSMAIN*OQ	changed
PAGE:P.GR*DQL	changed	RSTR:FSRELAB*OA	changed
PAGE:PAGEPR.LCS	changed	RSTR:FSTAPCOP*OA	changed
PAGE:PAGEPR*O	changed	RSTR:LFSRESTAR*OA	changed
PAGE:PAGEPR*SQL	changed	RSTR:LISTER*OA	changed
PAGE:X97CMDMAC	changed	RSTR:MAKEFILE	changed
PCP:AAMINOS	changed	RSTR:MERGE*OA	changed
PCP:ABMINOS	changed	RSTR:OLFSTEST*OA	changed
PCP:ADMINOS	changed	RSTR:ONLINEFS*OA	changed
PCP:AEMINOS	changed	RSTR:REGENERAT*OA	changed

RSTR:RST	changed	SEG2:LCSLCS	new
RSTR:RSTFMT0*OA	changed	SEG2:LCSPRLCS	new
RSTR:RSTMOUNT*OA	changed	SEG2:LOWSYM	changed
RSTR:SCANFILE*OA	changed	SEG2:MACRTNS	changed
RSTR:TAPEOUT	changed	SEG2:MESSSUBS	changed
RSTR:TAPEOUT*OA	changed	SEG2:MESSSWS	changed
RSTR:TAPEVIEW*OA	changed	SEG2:MM	new
RSTR:TSNCMD	changed	SEG2:MM.TBLS	new
RSTR:VTOCREAD*OA	changed	SEG2:MMGRSUBS	new
SCP:SCpload	changed	SEG2:MTACTNG	new
SCP:SCPTABLE	changed	SEG2:MTSSERV	new
SDS.:MAKEFILE	new	SEG2:NALCMDMACLIB	new
SEG2:ASMH.V2	changed	SEG2:NALMAP	changed
SEG2:CCTTABLE	new	SEG2:NALMAP	changed
SEG2:CHECKANS	new	SEG2:NALMAP	changed
SEG2:CLSCOPY	changed	SEG2:NASFILES.RM	changed
SEG2:CLSCREATE	changed	SEG2:NASFILES1	changed
SEG2:CLSDESTROY	changed	SEG2:NASFILES2	changed
SEG2:CLSDISPLAY	changed	SEG2:NASFILES3	changed
SEG2:CLSDUMP	changed	SEG2:NASFILES4	changed
SEG2:CLSDUPLICATE	changed	SEG2:NASLOAD	changed
SEG2:CLSEdit	changed	SEG2:NASTRTNS	new
SEG2:CLSEmpty	changed	SEG2:NETENV	new
SEG2:CLSFILEMENU	changed	SEG2:NEWCLSLOCKST	changed
SEG2:CLSFILES	changed	SEG2:OBJUTIL	changed
SEG2:CLSFSMESS	changed	SEG2:ONESHOT	changed
SEG2:CLSFTP	new	SEG2:PAGEPR	changed
SEG2:CLShelp	new	SEG2:PASCALJB	changed
SEG2:CLSINFO	new	SEG2:PASCALJBLIB	changed
SEG2:CLSlist	changed	SEG2:PASCALVS	changed
SEG2:CLSLOG	changed	SEG2:PASCALVSLIB	changed
SEG2:CLSMaKe	changed	SEG2:PERSUB	changed
SEG2:CLSMESSAGESY	changed	SEG2:PIPEDSP	new
SEG2:CLSMOUNT	changed	SEG2:PLUS	changed
SEG2:CLSNET	changed	SEG2:PLUS68	new
SEG2:CLSPeek	changed	SEG2:PL1FLIB	changed
SEG2:CLSPERMIT	changed	SEG2:PRINTMAP*OA	changed
SEG2:CLSRENAME	changed	SEG2:PRINTNMAP*OA	new
SEG2:CLSSET	changed	SEG2:RMCLSSSTA	new
SEG2:CLSTRUNCATE	changed	SEG2:RMGRLOAD	new
SEG2:CMDSERV	changed	SEG2:RMOPR	new
SEG2:CMS	new	SEG2:RMSRVLOAD	new
SEG2:C87	new	SEG2:SCREENRTNS	changed
SEG2:DSPDSR	new	SEG2:SITEROUTINES	changed
SEG2:DSPLOAD	new	SEG2:SYMLOAD	new
SEG2:EDITSUB	new	SEG2:S2L	changed
SEG2:FIX	changed	SEG2:TAPECATRTNS	new
SEG2:FMTable	new	SEG2:TAPERTN	new
SEG2:FNTRTN	new	SEG2:TIMERTNS	changed
SEG2:FORUM	changed	SEG2:TLNTDSP	new
SEG2:FRMTable	new	SEG2:TRMTYP	new
SEG2:FSHELP	changed	SEG2:TRTABLES	changed
SEG2:HIMDSPS	new	SEG2:UDROUTINES	changed
SEG2:IF66	new	SEG2:VIRTFILE	changed
SEG2:IF77	new	SEG2:VSF2COMP	new
SEG2:IG	changed	SEG2:VSS	new

SEG2:VSSCMS	new	TSTP:\$PRJDIRECTOR	new
SEG2:WTABLE	new	TSTP:\$PROJECT	new
SFIL:DEFINITIONS	new	TSTP:%ACCOUNTING	new
SLID:MAKEFILE	new	TSTP:%ACCOUNTINGX	new
STRT:MMS	new	TSTP:%ACCOUNTING1	new
STRT:MNS	changed	TSTP:%ACCOUNTING2	new
SYS:AMXEREP	new	TSTP:%ACCOUNTING3	new
SYS:AUTOHASP	changed	TSTP:%ACCOUNTING4	new
SYS:BADDEVS	changed	TSTP:%ACCOUNTING5	new
SYS:CCP	changed	TSTP:%PROJECT	new
SYS:CCPINIT	changed	TSTP:CKID.ACCESS	new
SYS:CHCHK	changed	TSTP:CKID.LOG	new
SYS:CLOCKWATCHER	changed	TSTP:CKID.PURGE	new
SYS:CMDTAPE	changed	TSTP:CKID.SIG	new
SYS:DWBLOCK	new	TSTP:CLK.MASTER	new
SYS:EREP.MAC	new	TSTP:CMDMACLIB	new
SYS:FIXFEP	changed	TSTP:DCK	new
SYS:HLOG	changed	TSTP:FORUM.MASTR1	new
SYS:HUH	changed	TSTP:FORUM.MASTR2	new
SYS:ITX	changed	TSTP:HSP	changed
SYS:JSTCMD	new	TSTP:HS88010100	new
SYS:MSDM*OG	changed	TSTP:IPL.MP	changed
SYS:MSSCN	changed	TSTP:IPL.370	changed
SYS:OLG	new	TSTP:IPL.4K	changed
SYS:OLG*H	new	TSTP:IPL.64	changed
SYS:PIDENT	changed	TSTP:LAS	changed
SYS:PLOTSTAT	new	TSTP:MAIL.FIREFIG	new
SYS:PLT51	changed	TSTP:MAKEFILE	new
SYS:RMOPR	new	TSTP:MESSAGES	changed
SYS:RMSCMD	new	TSTP:MESSAGES	new
SYS:SDM	changed	TSTP:MMGR.LOCK	new
SYS:STATSAV*OG	changed	TSTP:MMGR.LOG	new
SYSU:CSTP2XA*OA	new	TSTP:NAL7	changed
SYSU:MAKEFILE	new	TSTP:NASFILES5	changed
SYSU:NAS.MAKE	changed	TSTP:NASFILES6	changed
SYSU:NASTEST	changed	TSTP:NASFILES6.UM	changed
SYSU:PRMBLK	new	TSTP:NASFILES7	changed
SYSU:PRMBLK*OA	new	TSTP:NETLOG.BIT	new
SYSU:SUMMPRT#SG	changed	TSTP:NETLOG.EXP	new
TAPE:CROSS		TSTP:NETLOG.IMP	new
		TSTP:RATEFILE	changed
.#SQ	new	TSTP:REMOTE.DIR	new
TAPE:MAKEFILE	new	TSTP:SITETABLE	changed
TAPE:TPR.LOG	new	TSTP:SVWCMD	changed
TAR:MAKEFILE	new	TSTP:S2FILES	changed
TMTS:LOADMTS	changed	TSTP:S2FILES	new
TMTS:POSTIPL	changed	TSTP:S2FILES	new
TMTS:POSTIPL2	new	TSTP:S2FILES.UM	changed
TNGO:MAKEFILE	new	TSTP:S2FILES.UM	new
TSTP:\$ACCOUNTING	new	TSTP:S2FILES.UM	new
TSTP:\$ACCOUNTINGX	new	TSTP:S2L	changed
TSTP:\$ACCOUNTING1	new	TSTP:TAPEMASTER	new
TSTP:\$ACCOUNTING2	new	TSTP:TPTABXA#SA	new
TSTP:\$ACCOUNTING3	new	TSTP:TPTAB370#SA	new
TSTP:\$ACCOUNTING4	new	UD\$.:MAKEFILE	changed
TSTP:\$ACCOUNTING5	new	UD\$.:UDMASDUPLOG	new

UMPS:CLRPROF	new
UMPS:CMDMACLIB	new
UMPS:DBJP	changed
UMPS:IPLGTUM*OA	changed
UMPS:IPLINIT*OA	changed
UMPS:IPLREAD*OA	changed
UMPS:MAKEFILE	new
UMPS:PDPBUG	changed
UMPS:PISTLE*WF	changed
UMPS:PRINTDUMPA	changed
UMPS:PRINTDUMP370	changed
UMPS:PRNTCNTS	changed
UMPS:PTRACEXA*OA	changed
UMPS:PTRACE370*OA	changed
UMPS:RESSYS*WF	changed
UMPS:SAVEPROF	new
UMPS:SVC*PX	changed
UMPS:UBTSTTAB*OA	new
UMPS:UMLOAD*OA	changed
UMPS:UMTABLES#SA	changed
UMPS:UMTSTTAB*OA	new
UNSP:COUNTW	changed
UNSP:HASPLOG	changed
UNSP:TAPEDRIVS*OA	changed
UNSP:TESTIO	changed
VMXA:MAKEFILE	new
VMXA:VMCMD*OQ	new
VMXA:VMP	new
VMXA:VMPRINT	new
WSCV:MAKEBINHEX	new
WSCV:MAKEPAINT	new
W013:MAKEFILE	new
W030:CREBATEX	changed
W030:MAKEFILE	new
W030:PLOTSTAT	new
W052:IEHMOVE.MAKE	new
W052:MAKEFILE	new
W062:LBSNIFF.MAK	new
W062:MACSCAN.MAKE	new
W062:MACUTIL.MAKE	new
W062:SRCHSORT.MAK	new
W930:MAKEFILE	changed

Many resident system object decks have been changed, renamed, added, or deleted relative to the MTS/5.1 system. The following is a list of all the decks in the MTS/6.0 XA Starter System (in *IPL.MP) and an indication if they have been changed, renamed, or added since MTS Distribution 5.1. MTS/5.1 decks which have been deleted in the MTS/6.0 resident system are also indicated. Decknames ending in ".XA" will be different in the 370 version of MTS.

ACCSUB	new	DSRS	changed
ASTATSUB		DSTYPE	changed
BJPMOD	changed	DYS.....	deleted
BLOKLETR	changed	E_ATT	new
BLSTDEV		E_IO	new
BROADCST	changed	E_QUIT	new
BUFALOC	changed	E_TIMER	new
CACHTABL	new	ECORE	new
CARDUC	changed	ENDSEG0	new
CAT	changed	ENDSEG2	
CCDEFS	changed	EXIT	changed
CCHSUBR.....	deleted	FAKE_DISK_MANAGER	
CCSYMBOL	changed	FBJTJL	changed
CCTB		FBUILDPM	new
CFDUB		FEPI	new
CHARGE.....	deleted	FESNIFF	changed
CHECKFE	changed	FIDCQ	changed
CLOCKFIX	new	FLINE	changed
CLOCKSET.XA	new	FNAMETRT.....	deleted
CLPARSER	new	FNDJTJL	changed
CLPLCS	new	FSUB	changed
CMDBXLE	new	GATE	changed
CMDS	changed	GETD	changed
CMDSTAT	changed	GETFINF	changed
CONFIG.CARD.COMMENTS	new	GETRATES	changed
CONFIG.CARD.MP	change,renamed	GUINFO	new
CONFIG.MP	changed,renamed	GUINFYBL	new
CONSIO.XA	changed,renamed	G11	
CONSOLE.XA	new	ICOSTTAB	changed
COST.....	deleted	IGFC60.....	deleted
CVTOMR		IGFC70.....	deleted
DEVCTRL	new	IGFC80.....	deleted
DJDE9700	changed	IGF00580.....	deleted
DMGR_DISKXF	changed	IGF01470.....	deleted
DMGR_DISPATCH	changed	IGF01580.....	deleted
DMGR_ENDJUNK	new	INFO.....	deleted
DMGR_ITSKXF	changed	INITJE	changed
DMGR_LOCK	changed	INTERPROC	new
DMGR_MAIN	changed	INTERTSK.XA	changed,renamed
DMGR_MISC	changed	ITBRDCST	new
DMGR_MNTR	changed	ITGETMSG	new
DMGRSTAT	changed	ITJOB	changed
DSRDUMMY	new	ITNET	new
DSRI	changed	JBRP	changed

JMSGTDR	new	PUNUC	changed
JOB_DUMP		P11	
JOBLIST	new	QJOBENTR	new
JOBLST.ENTRIES.....	deleted	QJOBLIST	new
JOBLST.HEAD.....	deleted	QN.....	deleted
JOBLST.TAIL.....	deleted	RATEVEC	changed
JOBS	changed	READ	changed
JOBSTA	changed	REDL	changed
JULGRG_RTNS		RESLCSPR	new
KWIC	changed	RMJOBLIST	new
LDT.CARD		RNBR	changed
LLXU	changed	RSF	changed
LOWCORE.DEFS.MP	change, rename	RWSE	changed
MCHCCH.MP	changed, renamed	SDUMP	changed
MCHCRW	new	SEGMENT.CARD	
MCHSUBR.....	deleted	SEG0.NCA	
MONITOR_DUMP	new	SEG0_HIGH_ADDRESS	new
MONITOR_ENVSW	new	SEG0.NCA.TABLES.....	deleted
MONITOR_GETFREE	change, rename	SEG1.NCA	
MONITOR_LCSPPR	new	SEG1_HIGH_ADDRESS	new
MONITOR_MNTR	changed	SEG1_LOW_ADDRESS	new
MONITOR_MTSPGNTU	new	SEG1.NCA.TABLES.....	deleted
MONITOR_ONUNIT	change, renamed	SIGNONM	
MONITOR_PGNTUNIT	new	SMDSTBL	changed
MONITOR_PRVI	changed, renamed	SOFTCHK	
MONITOR_SERVICE	changed	SPELLCHK	changed
MONITOR_SYNCWAIT	new	SSCN	new
MONITOR_UCRTNTAB	changed	SSRTN.XA	changed, renamed
MONITOR_UNITNODE	change, rename	STARTUP/SHUTDOWN	changed
MONITOR_WAIT	changed	STATBUFF	
MOVE		STATJOB	changed
MSG	changed	STATSUB	changed
MTS	changed	STATSW	changed
MTSBATCH	new	STDDMP	changed
MTSCAT	new	STOPJOB	changed
MTSINIT	new	STOR	changed
MTSMACI	new	SYSDEFS.XA	changed, renamed
MTSSTOR	new	TABLES.MP	changed, renamed
MXFRTN		TABLMOD	changed
NASRTNS	new	TABLRTN	changed
ONLINE/OFFLINE.XA	change, renam	TASKS	changed
OPEN	changed	TASKSTAT	changed
OPERATOR	changed	TBLS	chnaged
PATH_DEVICE.XA	change, renamed	TIME	new
PATRN	new	TIMECONV	changed
PDP.DM	changed, renamed	TIMT.....	deleted
PER.XA	new	TIMZONTB	
PICTURE	changed	TN.....	deleted
PLIM	changed	TPMOVE	changed
PLUS_LINKAGE	new	TRAK	changed
PLUS_RESLIB	new	TRTABLES_RES	
PLUS_TIME	new	TR3270	changed
PN/QN/TN	changed, renamed	TTTABLES	changed
PRINTLAW	new	T11	
PTRUC		UCDISK	changed

UCTAPE	changed
UCTVSYNC	changed
UC3270	new
UC3284	
UC3330	changed
UMLOAD	changed
UMLOADTV	
UMMPS.MP	changed, renamed
UNITS.XA	changed, renamed
USUB	changed
UTIL	new
VOLG	changed
WRIT	changed
XASCEBC	new

INSTALLATION INSTRUCTIONS FOR NEW INSTALLATIONS

MTS Distribution 6.0

April 1988

1. The starter system requires a minimum of one 3270 terminal, one 3380 disk drive, and one 9-track tape drive. Each device may appear at any of the addresses for the corresponding device type in the following default configuration (for XA machines, "address" indicates the device number):

<u>Device Type</u>	<u>Address</u>	<u>Device name</u>
3287	0000	PTR1
3270	0001-0007	DS01-DS07
3270	0009	CON1
2540R	000C	RDR1
2540P	000D	PUN1
1403	000E	PTR2
3270	001F	CON2
3287	0100	PTR3
3270	0101-011F	DS21-DS3F
3420	0180-018F	T900-t90F
3330	0200-020F	D100-D10F
3350	0220-022F	D200-D20F
3370	0240-024F	D300-D30F
3380	0260-026F	D400-D40F
3420	0C70	T920
9335	0D00-0D03	D500-D503
9335	0E00-0E03	D600-D603

MTS refers to devices by name rather than by address (or device number, in XA mode). You'll need to know the device addresses only for purposes of defining your configuration and IPLing the system; once MTS is running, use the "Device Name" column to determine the name of any device.

The starter system will select any one of CON1, CON2, DS01, or DS21 as the operator's console (whichever it finds first). At least one of these devices must be included in the configuration. DS01 is preferred, because the IPLREADer (see step 2) will talk to that device without any special prompting.

In addition, you must determine which architecture your machine supports (XA or 370) and, for the 370 architecture, whether your machine supports 2K or 4K storage protect keys, and whether your machine supports one-megabyte segments. Each possible combination of options is identified by a code; you will use the code later to

specify which version of MTS to run. For XA machines, the code is "MP"; for "standard" 370 machines (2K keys and 1M segments) the code is "370"; for 370 machines with 4K keys and 1M segments, the code is "4K"; for 370 machines with 2K keys and 64K segments, the code is "64".

2. You are now ready to load MTS. Mount the Dump/Restore tape and IPL from it. The message "Do you want to run the current system" will appear on the console (the first usable terminal in the sequence DS01, CON1, CON2, DS21). Enter the command:

```
RUN NAME=*IPL.xxx.NDSK ADDR=yyy
```

where xxx is the code indicating the type of machine and yyy is the address of the tape drive where the utility tape has been mounted. The no disk version of MTS should now start running.

3. The system may prompt for the current time and date if you are running in a real machine and if the TOD clock is not set. For example, entering the time and date as For example, entering the time and date as

```
14:41:00 5/1/88 EDT
```

will set the time and date to 14:41 Eastern Daylight Time on 1 May 1988. MTS keeps GMT in the TOD clock. Although this agrees with the standard set by IBM for the use of the TOD clock, some IBM systems do not do this. This means that even if the clock has been set by another system, it may be off by several hours and you will want to reply "NO" when asked if the time is correct. You can then enter the correct **local** time, which MTS will use to load the TOD clock with GMT.

4. At this point you will be told that MTS600 couldn't be found (which is reasonable since you haven't restored it yet) and you should respond by entering CANCEL. This will result in a fairly serious looking message about an error in catalog initialization which can be ignored.

5. Start the job status master job by typing

```
JOBS MAS
```

at the operator's console.

6. The following steps can be done from the operator's console or from a 3270 terminal. If you are going to work from the operator's console you'll need to start an MTS job by

typing

```
MTS NDSK OPER
```

at the console. If you are going to work from a terminal you'll need to start an MTS job on that terminal by typing

```
MTS NDSK xxxx
```

at the operator's console, where xxxx is the MTS device name of the terminal to be used. The NDSK parameter tells MTS to avoid using the file system.

7. Next, signon by entering

```
SIGNON MTS.  
MTS.      (the password)
```

If you are working at the operator's console you won't be asked to enter a password. If you are working from a terminal you will need to enter a password and because you are using the no disk system the password will be the same as the ccid used on the signon command (MTS. in this case).

8. Next attach the MTS Utility tape to your job by entering

```
$GET >Txxx  
$CONTROL *AFD* VOLUME=MTSUTL
```

where Txxx is the MTS device name of the tape drive where the MTS Utility tape is mounted.

9. Next DASDI the public volume to be restored by entering:

```
$CONTROL *AFD* POSN=DASDI  
$RUN *AFD* PROT=OFF  
Dxxx MTS600 VX 1 IPL  
$ENDFILE
```

where Dxxx is the name of the device where the pack to be restored is mounted. See the General Notes (comp. 461/19) for further instructions on using FILE:DASDI.

10. Use DISKCOPY to restore the D6.0 system from the distributed dump/restore tape(s). You should specify the IPL option (SLOW and SWAP aren't necessary). See the General Notes for information on running DISKCOPY.

```
$CONTROL *AFD* POSN=DISKCOPY  
$RUN *AFD* PROT=OFF  
TAPE
```

```

>Tnnn
DISK
Dyyy MTS600
IPL
$ENDFILE

```

where >Tnnn is the name of the MTS device on which the DZ dump/restore tape is mounted and Dyyy is the MTS device name of the disk drive to which the data will be restored.

11. Stop your machine, and IPL from the new disk. Enter the command:

```
RUN NAME=*IPL.xxx
```

where xxx is the code indicating the your machine type. The D6.0 version of MTS should now be running.

12. Proceed with the start-up procedures described in the MTS Operators' Manual (comp. 592). This time you shouldn't be told that MTS600 couldn't be found.

After initialization is complete you should do the following from the operator's console or a terminal

```

SIGNON MTS
password (see below)
$RENAME *IPL.xxx *IPL.0

```

where xxx is the code indicating your machine type. This sets up the version of MTS appropriate for your machine as the "current" system; this version will be selected by the IPLREADer each time you reload and either just hit ENTER or type "YES" in response to "Do you want to run the current system."

There are two classes of userids on the distributed test system: shared system userids and private userids. The shared system userids are those userids like MTS. or FILE.

for the shared system userids is "AARDVARKS". The passwords for the STnn userids is the same as the userid. (Note that if you are using the operator's console as your terminal, a password is not required for the shared system userids, but is required for the STnn userids.) The shared system userids are protected by the program *CKID. If you attempt to sign on to one of those userids from a terminal other than the operator's console, you will be prompted for a password. After entering AARDVARKS, you will be prompted for a second userid and password. As distributed, the userids ST00 through ST09 have access to the shared system userids, so a typical signon to one of the shared system userids might look like:

```

signon mts
aardvarks      (password for userid MTS)
st01           (second level userid, in response to the
               "ID?" prompt)
st01           (second userid's password, in response
               to the "PW?" prompt)

```

You should note that this set of passwords provides essentially no security, and you should change the passwords for both the shared system userids and the private userids before beginning any kind of production use of the system.

13. You now have a working D6.0 version of MTS. You can start to run user programs after you've created some user IDs. To do this, see the description of the accounting maintenance procedures (104/137) and (104/138), which you should obtain from the *FS tapes.

There are several things that you will probably want to do to clean up a few loose ends before going much further. Some of these are:

- A. Fix HASP for whatever local options you desire. See the description of these options (comp. 387/26). You may also want to change some of the commands in STRT:HSP, the command file for *HSP, which can be used by the operators to issue the appropriate HASP \$START commands after starting HASP.
- B. You will also want to change the contents of STRT:LAS, the command file for *LAS, which can be used by the operators to start up the terminal lines.
- C. Fix up the command statistics directory file SYS:CMDDIR for the appropriate tapes. If you don't do this, the system will occasionally mount a tape with rack number CMDTP (which must be labeled CMD001) to dump the command statistics data. Eventually this tape will fill up with command statistics data and the system will get mad unless you have fixed up SYS:CMDDIR to give it more tapes to use. A description of the structure of this file is available in the Operators' Manual on the *FS tapes (comp. 592).
- D. Install ASMH. Once you have a signed and paid license agreement with IBM for ASMH contact UM for a starter version of ASMH (ASMH is an IBM program product and as a result cannot be included on the standard MTS distribution tapes).
- E. Build a new set of tables which describe your configuration. Building TABLES is described in the XA or 370-mode TABLES macros writeup.
- F. If you are planning to run with more than one disk

pack, you should use the utilities under the CCID BNCH to "smear" the one-pack system across several disk volumes to improve system performance (as opposed to just adding several empty volumes to the system).

- G. If you add additional disk volumes, you should allocate additional spooling and paging extents on some of the volumes to further improve system capacity and performance. See the TABLES writeup for details.

14. General things new installations should know.

- A. How to sign on using the operator's console:

1. Enter "MTS OPER" on the operator's console
2. It will come back immediately for input.
3. Enter the signon command.
4. You are now signed on if the ID exists. No password is required for signon from OPER for "staff" IDs and SIGFILES are not processed. Also the "last signon" message and the "signed on at" message are not printed.

- B. 2305 paging devices must be named FF00 to FF07 for the first one, FF08 to FF0F for the second one, FF10 to FF17 for the third, etc. 3805s and 3825s should be named FB00, FB01, etc.

- C. A "rich" ID is one with X'80' on in the second byte (byte 1) of the accounting record (set by accounting maintenance program on request). If the ID is rich then no checking for maximum money, etc., is made. The expiration date and maximum file space, however, are checked. (Just to confuse things, the accounting utilities refer to a rich ID as "privileged.")

- D. A "PROT=OFF" ID is one with X'02' on in the third byte of the accounting record. This bit allows the user to run programs with PROT=OFF, and to use other privileged facilities, including privileged SYSTEMSTATUS commands. (The accounting utilities refer to a prot=off ID as a "PROT" ID.)

- E. A "public file privileged" ID is one with X'08' on in the second byte of the accounting record. This allows the user to create a public file.

- F. A user with either ACCTLB or ACCPLB on in his accounting record can set LSS (Limited-Service State) off even if the load is too high.

- G. Certain IDs are used automatically during the start-up sequence (INIT, SEG2, SYS., STRT). These IDs must be part of project WOPN on the test pack system so that they can signon even if the "in use" bit in the

accounting record is set. The project number used for this test in your production system may be changed by using a different project or project substring (or set of them) on the &IPLPROJ SETC symbol in COPY:GLOBALSETS and reassembling MTS.

- H. MTS makes certain checks to be sure that only IDs that belong to individuals that work for the computing installation can perform certain "privileged" operations. This check is made by checking the project number associated with the ID. At UM only projects that start with "W" are assumed to belong to installation staff. This may be changed by altering the setting of the SETC symbol &SYSPROJ in COPY:GLOBALSETS and assembling and installing a new version of MTS.
- I. The distributed system contains a dummy version of the rate number subroutine (RNBR) which gives zero rates. You will almost certainly want to replace this routine with a version that calculates rate numbers according to the rules of your installation and also update the file *RATEFILE to include your MTS rates.

MTS: The Michigan Terminal System

April 1987

The University of Michigan, acting as agent for the group of universities which have jointly developed the Michigan Terminal System (MTS), has established a policy governing the licensed use of MTS by other organizations. The policy is to make MTS available to academic institutions as a licensed program for an annual fee of \$5,000. For other non-profit organizations, the annual fee is \$10,000; for commercial organizations, the annual fee is \$25,000. In all cases, there are restrictions imposed by the license agreement including the stipulation that the licensee shall not provide access to MTS under circumstances that could be construed as providing commercial computing services. During the annual license period, the licensee will receive one copy of the full set of MTS distribution tapes, any incremental distributions prepared during the year (generally one or two), written installation instructions, and two copies of the current user documentation.

Since we do not have a large system support staff available to help organizations desiring to run MTS, we suggest that such organizations have system programmers who are able to do routine maintenance and diagnose simple problems. The existing MTS installations are each quite self sufficient, some of them with as few as two or three system programmers responsible for MTS.

Background of MTS

Development of MTS was begun at the University of Michigan in the mid-1960s specifically to provide general, interactive computing services on the IBM 360/67 computer; MTS has been in production use in Ann Arbor since 1967. The Universities of British Columbia, Alberta, and Newcastle upon Tyne (England), and Wayne State University have all been using MTS for over 10 years. In addition, Rensselaer Polytechnic Institute, Simon Fraser University, a research institute in Brazil, and Durham University (England) all use MTS as their primary system for research and instruction. MTS is currently being run on the following computing systems: Amdahl 5860, 5870, 470V/8, IBM 3090-400 (with vector facility), 3081G, 3081D, 3033N, and 4361. In addition MTS has been tested on or was used for production on the following machines: Amdahl 470V/6, 470V/7, 5890, IBM 3033U, 370/148, 370/168, 370/158, 4341, and NAS 9060, and XL.

MTS originally was designed for virtual memory interactive service only. The batch capability was added after a few years when it became clear that MTS would be used in production environments where it was necessary to be able to run large numbers of low-overhead batch jobs. Fortunately, this has meant that interactive support and virtual memory have not had to be retrofitted into the system (as in the case of some IBM systems). At the University of Michigan, interactive access is

used for over 85 percent of all jobs; batch being primarily used for those jobs which users are willing to defer to off-hours when lower charge rates are in effect.

Large private paged virtual memories (currently up to 9 megabytes for each user) have been a "way of life" in MTS since the beginning. MTS users have always been able to run large programs and programs with large arrays without being concerned with overlays, a concept which has never existed in MTS. In addition, MTS itself and many system utilities and compilers (those which are re-entrant) reside in shared, paged, read-only virtual memory (VM). Because of this, most MTS commands require less than a few dozen pages of private VM to execute, whereas a FORTRAN-G compilation may take 60 to 70 pages of VM since that compiler is not re-entrant and thus must be loaded into private VM for each user.

MTS provides a comprehensive shared-access file system which is not compatible with IBM software systems. The user-level program environment is also different from IBM systems in several respects, although many programs written to run in IBM systems will work with little or no modification. The format of object decks in MTS is essentially compatible with IBM, but the SVC instructions are completely different (and not normally used by MTS user-level programs). In addition, MTS provides two programs to emulate the OS/VS application program environment; they allow many OS/VS application programs to run without modification.

MTS supports a comprehensive set of language processors including FORTRAN (G, H, VS, WATFIV, and IF), PL/I (F, Optimizing, and PL/C), COBOL (U, VS, and WATBOL), Basic (MTS and Waterloo), Pascal (VS and several others), APL/VS, Snobol/Spitbol, Algol W, CSMP, GPSS, LISP, SIMSCRIPT 2, Reduce/3, SLIP, and assemblers for the IBM 370 (G and H) and various other computers including PDP-8, PDP-11, and many microprocessors.

MTS also provides a state-of-the-art context editor which includes full-screen editing through an interface to the MTS device-independent screen support routines which can function on several different terminal types. This screen management is also used by several other MTS components, including an online HELP facility, and is available to user-level programs.

A comprehensive message system is provided by MTS. The message system has a network interface allowing messages to be sent to and received from other host systems. Two MTS conferencing programs are available to licensees; one, *FORUM, is included as a part of the distributed system. A powerful conditional command and macro processor package was added to the system in 1983.

Much of the development of MTS is done jointly with other MTS installations. Representatives of these installations meet once each year for a workshop at which most of the discussion

relates to what new features and changes are desired and who will do the design and implementation.

Machine Requirements

MTS requires an IBM 370 compatible machine (including the 30xx machines) with at least two megabytes of memory and the following features:

- Floating Point
- Universal Instruction Set
- CPU-timer and Clock-Comparator
- Translation with 1M segments and 4K pages
- Conditional-Swapping
- PSW-Key-Handling
- Channel Indirect Addressing on all channels
- Clear I/O on all channels

It will make use of the following features if they are available:

- Extended precision floating point
- Vector facility
- Direct control (limited use)
- Branch and Save
- Fast release on channels
- Invalidate Page Table Entry
- Common Segment Facility

Other features are not used, unless user programs make use of them. MTS will likely be extended in the future to make use of the Dual Address Space feature. It has been changed to run with either 1M or 64K segments for the IBM 30xx and will operate in either 370 or XA (Extended Architecture) modes, including multiple processor support, but these changes are not yet available in the standard distribution system.

MTS can run under VM/370, but performance is poor under heavy load. As distributed, MTS will not make use of any of the features of VM to enhance performance of guest operating systems, although versions of MTS have been prepared in the past that do so.

The MTS file system normally uses IBM 3330 and/or 3350 compatible disks in any combination although it also supports older types of disks. Support has recently been added for IBM 3370, 3340, 3344, 3375, 3380 and Amdahl 6280 disks, but these are not yet available in the standard distributed system.

MTS also provides support for the Xerox 9700 page printer, the Autologic APS-5 phototypesetter, and for both IBM and ANSI standard magnetic tape labeling/blocking and the normal IBM unit record equipment.

MTS Performance at the University of Michigan

On a typical weekday afternoon at the University of Michigan with 400+ concurrent interactive terminal users and 125 batch jobs per hour, there are usually about 35,000 pages (123 megabytes) of total VM in use, of which about 31,000 pages are private and the remainder shared. An average user has about 90 pages of private VM. There are usually at least 60 users running large programs (150 to 500 pages of VM) and a few running even larger programs (500 to 1,000 pages). Our 64-megabyte IBM 3090-200 is about 50 percent saturated at this load.

For primary paging devices, we currently use one Intel 3825 and two Intel 3805s each with about 11,000 4k pages and each on a separate block multiplexor channel. These devices run in native mode, not compatible with an IBM 2305. This hardware provides about 90 megabytes of high-speed backing store and permits paging rates of approximately 450 page reads per second to be achieved easily although typical levels in actual operation are much lower. The associated page writes amount to approximately 300 additional pages per second at peak levels. In addition, MTS can also use IBM 2305s for primary paging and support for expanded storage on a 3090 is being added. Generally, a disk is also used as a secondary paging device and MTS can be run using only disks for paging if necessary.

We have a mixture of 6280s and 3380s with a total file system capacity of about 18,100 megabytes currently.

Performance data is also available for other system configurations. At Michigan, MTS also runs on a second 3090-200 and on an IBM 4361.

Terminal Access and Networking

Terminal support is provided using IBM 3270s, Memorex 1270s, and, at most MTS installations, locally developed network processors (generally PDP-11 based). At Michigan, we have five PDP-11 network nodes which are used to interface MTS to the UMnet/Merit computer network and to Telenet. These network processors allow approximately 600 concurrent terminals to access MTS. The network provides access from a diverse set of ASCII terminals (including micros and other intelligent terminals) to several computers on our campus including the central MTS systems. It is composed of PDP-11 and LSI-11 switching and terminal access nodes linked together by twisted pair and Ethernets operating at speeds up to 1.5 Mbps. It is interconnected with the regional Merit network and Telenet allowing terminals on the network to access other computers in the state and beyond.

The UMnet software can be separately licensed and the hardware purchased as can the MTS network software from some other MTS installations.

In addition, we have several dozen **local** 3278s (MTS does not support remote 3270 terminals).

Obtaining Additional Information

Further details on MTS, including information about user-level documentation and the MTS License Agreement can be obtained by contacting the following:

MTS	Michael T. Alexander,	(313) 763-4890
	James M. Bodwin,	(313) 763-3733
	Jon C. Sell,	(313) 764-5324
License policy	Leonard J. Harding,	(313) 763-6054
Distribution	Suzan W. J. Alexander,	(313) 763-4906
Address	University of Michigan Computing Center 535 W. William St. Ann Arbor, MI 48103-4943	

From MTSDist-v3-21Dec2004/ReadMe by Mike Alexander

Distribution 6.0 was similar to 5.0 and 5.1. It consisted of one utility tape, 3 dump/restore tapes, and 6 *FS tapes (in the 6250 BPI version of the distribution). All of these tapes have been copied to AWSTapes. The utility tape was originally labeled, but the labels were lost when it was copied to a file. I reconstructed the labels from a *LABELSNIFF listing I had (since they matter) and the result is an AWSTape in d6.0util.aws. The volid is MTSUTL. The *FS tapes are in d6.0tn.aws as labeled tapes with volid 6.0tn (n = 1...6). The dump/restore tapes are all unlabeled tapes in d6.0dr1.aws through d6.0dr3.aws. There were a few problems copying the second dump/restore tape, but I think it was copied ok in the end. The other tapes copied without trouble.

The documentation for this distribution (which was the last MTS distribution) is similar to the previous ones with the addition of D6.MTS_DOC which is a general description of MTS and the MTS Consortium.

Mike Alexander
1309 Gardner Ave.
Ann Arbor, MI 48104
USA
mta@umich.edu

18 December 2004