MULTICS

INSTALLATION INSTRUCTIONS

RELEASE MR12.3

PREPARED BY:

MULTICS SOFTWARE DEVELOPMENT

December 01, 1989

Worldwide
Information
Systems

Bull

CONTENTS

SECTION 1

INTRODUCTION

This document describes the MR12.3 package. Detailed instructions for installation of a system for the first time and upgrading to MR12.3 from an MR12.2 system constitute the major portion of this document.

No attempt is made to document features of MR12.3 other than those directly required for its installation.

Software Releases prior to MR12.1 are no longer supported. All sites running pre-MR12.1 software are strongly encouraged to upgrade to MR12.1 as soon as possible.

SITE SUPPORT

System Representatives who support Multics should apply for registration on the SiteSA project maintained in Phoenix on System M.

The SiteSA project was created to allow sites to communicate with Multics System Support (MSS) personnel on matters of site support. MSS maintains and pays for usage on this project. Only System Representatives who support a Multics site are authorized to use the project. They are authorized to use it only for site support activities which require communication of information to MSS personnel.

MSS would appreciate that the SiteSAs maintain the site's info segments in >udd>SiteSA>site_info (>udd>ssa>si). Information on maintaining these segments is in the info segment, >udd>ssa>si>site_info.info.

In addition to normal telephone and Multics mail communications with MSS personnel, SiteSAs are encouraged to use the unusual_crash_log forum meeting to report any unusual problems encountered at their site. SiteSAs not familiar with forum can type "help forum" for usage information. Forum is a subsystem

somewhat like the mail system in which an electronic meeting can be held. To invoke forum, type:

```
forum >udd>ssa>sa_meeting>uclog
```

Type ? to print a list of requests allowed by forum. Other forum meetings of interest to SiteSAs are:

PATHNAME	SHORT NAME
>udd>ssa>sa_meeting>install_instructions	ii
>udd>ssa>sa_meeting>hardware_problems	hwp
>udd>ssa>sa_meeting>critical_fixes	fixes

We would appreciate that sites as they complete installation of MR12.3 enter a transaction in the install_instructions (ii) forum meeting. This will allow other sites to see the amount of exposure the release is receiving and to communicate any problems encountered in the installation procedure.

MSS also suggests, very strongly, that SiteSAs DO NOT attempt to reproduce site problems on System M which are known to crash the system at their site, cause processors to loop in ring 0, etc. Such problems should be reported via the Trouble Report system, and will be verified by MSS personnel using appropriate resources, so that System M service is not interrupted unnecessarily.

Problems encountered while installing this release, or problems of a critical nature to a customer site (for a definition of critical, type: help tr.priorities) should be reported directly to MSS by mail or phone. European or Canadian sites should contact their local site support person for details on support from the Canadian, French or UK Technical Assistance Centers. Phoenix personnel include:

NAME	AREA	USER ID	PHONE
Frank Martinson	Mgr, MSS	Martinson.sm	602/862-4839
Paul Farley	TRs, C&F, Site Support	Farley.sm	602/862-4666
Joe Hirneisen	Sys. Integration Release Prep	Hirneisen.sm	602/862-4812

Note that all of the above numbers are available through HVN 862-XXXX (eg, 862-4839 for Martinson).

SECTION 2

DESCRIPTION OF PACKAGE

LIBRARY NAMING CONVENTIONS

The primary pathnames on most of the system directories are somewhat lengthy. For this document, abbreviated (added) names are used in lieu of the primary name. The following list gives the primary and abbreviated names used in this document.

>daemon_dir_dir >ddd	>system_library_1 >sl1
>documentation >doc	>system_library_obsolete >obs
>system_library_tandd >firmware	>system_library_standard >sss
>library_dir_dir >ldd	>system_library_tools >tools
>process_dir_dir >pdd	>system_library_unbundled >unb
>system_control_1 >sc1	>user_dir_dir >udd

The system directories firmware and obs, are not included in the standard system search rules. Segments in these directories must either be accessed by absolute pathnames or by changing the system search rules via the set_system_search_rules command added to the system_start_up.ec.

CONTENTS OF MR12.3 PACKAGE

MR12.3 includes this document, a set of magnetic tapes, hardcopy dump maps, and accompanying documentation. Sites not installing Multics for the first time will be able to proceed to the MR12.3 release from MR12.2. Sites upgrading to MR12.3 from MR12.2 refer to Section 4, sites installing Multics for the first time refer to Section 5. The instructions in each section will provide a procedure to guide a site through the installation.

Documentation for some of the new features and TRs resolved in this release is contained in the directory >doc>MR12.3 which is part of this release.

This is a total software release. It contains a complete set of all modules contained in the Multics standard system.

Appendix A will list modules added, modified or deleted since MR12.2.

Appendix B contains an outline of the use of MTR under Multics.

A set of master tapes was generated for this release and all dump maps reflect the contents of these master tapes. All tapes sent to the field are copies of the master tapes. Because of different lengths of magnetic tape reels, there may not be an exact correlation between a single tape and a dump map.

These differences, if any, are minimal and occur only on those sets that are multi-reel (e.g., 12.3LDD_STANDARD tapes). Site personnel may assure themselves of the contents of the tapes by visually matching the maps produced from the reload operations against the master dump maps supplied.

Tape Names	Description
12.3EXEC	Complete dump of Multics executable libraries >documentation, >obs, >sss, >tools, and >firmware.
12.3UNBUNDLED	Complete dump of Multics unbundled libraries, or portion thereof, for those sites purchasing Priced Software Products. Includes all, or portions of >unbundled, >ldd>unbundled, >ldd>mcs and >system_library_3rd_party.
12.3LDD_STANDARD	Complete dump of all standard library source, object, include files and info segs.
12.3MULTICS	Complete Multics System Tape (MST).
12.3MISC	The compout segment for this document and other supporting documentation, if any, are contained on this tape in >doc>MR12.3. Last minute changes, if any, made to software modules after generation of the above tapes is also contained on this tape. This is the last tape to be loaded.

The accompanying hardcopy listings are:

Listing	Description
12.3EXEC.DUMP.MAP	Contents of the 12.3EXEC tape.
12.3UNBUNDLED.DUMP.MAP	Contents of the 12.3UNBUNDLED tape (for those sites purchasing Priced Separate Software).
12.3LDD_STANDARD.DUMP.MAP	Contents of the 12.3LDD_STANDARD tapes.
12.3MISC.DUMP.MAP	Contents of the 12.3MISC tape.

Tape File Name	Describes
_system_book_	MR12.3 hardcore modules

SECTION 3

FCO AND FIRMWARE STATUS

FIRMWARE

The firmware identification for MR12.3 is "IFAD D.4".

IFAD (Integrated Firmware and Diagnostics) tapes, are distributed to all sites having valid Field Engineering Hardware Maintenance contracts. The IFAD tape is nolonger distributed as an FCO. It is now sent with the MR12.3 release tapes.

T&D (Test and Diagnostic) tapes, are distributed to all sites having valid Field Engineering Hardware Maintenance contracts. The T&D tape is distributed as FCO PHAFGA888, rev. B.2.

This version of IFAD, firmware, and T&D modules has been tested and exposed on System M in Phoenix. The modules are supplied as part of the MR12.3 release and can be found in >ldd>firmware as part of the LDD_STANDARD tape. When subsequent IFAD tapes become available, it is recommended that the site use the deckfile_manager, dfm, command to load the released IFAD tape into >system_library_tandd. The firmware modules should then be moved to >ldd>firmware and a new MST tape should be generated using generate_mst.

The MR12.3 release requires the following firmware revision levels.

```
Disc:  DSC191 -- V1
       DSC500 -- X1
       MSP800 -- K1

Tape:  MTC0500 -- V1
       MTP0601 -- T1
       MTP0610 -- Z2
```

Unit Record: Common -- A2
Reader/Punch -- B2
PRU1200/1600 -- M1

FIELD CHANGE ORDER LIST

The following list of FCOs should be thought of as a continuation of the similar list for MR12.2. Sites that are upgrading from MR12.2 will need to verify that FCOs for that release have been installed.

As FCOs are completed in Phoenix, they are sent out to the field sites. Some of the FCOs affect only site documentation while others may change the hardware to improve system reliability or maintainability. An example might be a change to improve the supply of cooling air to some boards so that the boards run cooler and are therefore less likely to fail. These types of FCOs are important to the long-term success of system operation. However, because these FCOs are not required to correct the results of any computer program, they are not listed in this SRB.

The purpose of this section is to identify those FCOs that are of utmost importance in the short term. This list includes only those FCOs needed to correct a program malfunction.

For those FCOs already sent to the field, the "FCO Kit Ship Date" column contains the approximate date when shipment of the FCO kits began.

One of the column headings in the tables on the following pages is "Round Robin". The implications of a "Yes" in this column for a particular FCO are as follows:

1. The FCO kit includes at least one board. The new board(s) is to replace the board(s) now in the equipment.

2. Enough FCO kits are put together for about 20% of the sites that are to take part in the Round Robin. Some FCOs result in a Round Robin for all sites. Other FCOs involve a Round Robin for some sites but not for other sites. For example, two of the factors that determine whether or not a particular site takes part in a Round Robin FCO are:

   a. The date codes of certain integrated circuit chips on a board.

   b. The board construction method -- wirewrap vs. multilayer.

3. The set of FCO kits are sent to the first set of sites. (For the remainder of this discussion, we will follow one kit as it makes its Round Robin journey.)

4. The site installs the FCO. When the site is satisfied that the FCO is satisfactory, the site returns the old board(s) to Phoenix.

5. If there is still equipment at other sites that have not had the FCO installed, the returned board(s) is reworked, tested, and another FCO kit is assembled in Phoenix and sent to the next site.

6. Since the length of time it takes a site to install an FCO and become satisfied that it is working correctly cannot be known in advance, there is no way to predict with accuracy how long it will be until the FCO has been installed at all sites.

The FCOs have been classified into the following categories:

CATEGORY 1 - The new software for this release will not run properly unless these FCOs are installed.

CATEGORY 2 - The new software for this release may or may not run properly if these FCOs are not installed. It is strongly recommended to have these FCOs installed to improve reliability of the system.

CATEGORY 3 - FCOs whose effects are program-visible, but do not fall into categories 1 or 2. Category 3 FCOs correct problems that are not uniquely related to this release.

CATEGORY 1 FIELD CHANGE ORDERS

FIELD CHANGE ORDERS VITAL TO THIS SOFTWARE RELEASE

FCO Number	Board	FCO Kit Ship Date	Round Robin	Est. Hours To Instl	FCO for Multics Release

No category 1 changes are required for MR12.3.

CATEGORY 2 FIELD CHANGE ORDERS

FIELD CHANGE ORDERS POSSIBLY NEEDED FOR THIS RELEASE

FCO Number	Board	FCO Kit Ship Date	Round Robin	Est. Hours To Instl	FCO for Multics Release

No category 2 changes are required for MR12.3.

CATEGORY 3 FIELD CHANGE ORDERS

FIELD CHANGE ORDERS INDEPENDENT OF A PARTICULAR RELEASE

FCO Number	Board	FCO Kit Ship Date	Round Robin	Est. Hours To Instl	FCO for Multics Release
-----INTEGRATED MULTIPLEXER UNIT (IMU)----------					
PHAFXS951	MCA	88 Nov	No	3.0	MR12.3
-----FIPS DISK MSU 3380/3382, MSP 3880/1/4/5----------					
PHAFFP553	Pwr Supply	89 Jan	No	1.0	MR12.3
PHAFFP556	Firmware	88 Dec	No	.4	MR12.3
PHAFFP557	Pwr Supply	89 May	No	.2	MR12.3
-----FIPS TAPE MTU 8205/06/08, MTS 8205/06/08/15/16/18-----					
PHAFFP434	Firmware	88 Dec	No	.2	MR12.3
PHAFFP436	FW Patch	89 Nov	No	.2	MR12.3
-----T&D RELEASE REV. B.2----------					
PHAFGA888	Tape	87 Mar	No	2.0	MR12.1
-----MHAT MR12.0----------					
PHAFPD998	Tape	87 May	No	2.0	MR12.0

SECTION 4

INSTRUCTIONS FOR SITES UPDATING TO MR12.3 FROM MR12.2

If problems are encountered in any of the Steps listed below, return to the last step known to be successful and retry the Steps. Pay particular attention to procedure. This entire procedure was verified on the Multics System at Phoenix Computer Operations, but some hardware/software differences may exist at a particular site.

In this section, two formats of text are used to indicate the typing of input into the system. This input usually is a command line, but could be data typed in response to a query from the operator console.

Strings of input, and output messages of importance are indicated on separate lines.

In addition, all input to the system is to be typed in lowercase, except when indicated by the occurrence of both uppercase and lowercase in the input line. User input is preceeded by an exclamation mark (!).

System display output is shown as is, except when the displayed line exceeds the page margins for this document. When this occurs, the displayed line is split into two lines, with the second line indented from the first.

SITES USING NSA IOM'S MUST INSURE OPTION WIOG008A-001 IS INSTALLED TO MAKE PAGED MODE I/O WORK.

STEP 1: PRE-INSTALLATION PREPARATION

It is recommended that sites perform a BCE SAVE. A double save is recommended to avoid any possible tape problems later.

Due to the method by which unbundled software is dumped, normal trimming during reloading of new software does not occur. To ensure that unbundled directories are clean execute the following before shutting down MR12.2 for the last time enter admin mode and type:

```
cwd >system_library_unbundled
answer yes -bf hpdl ([files **])
cwd >library_dir_dir>unbundled>source
answer yes -bf hpdl ([files **])
cwd >library_dir_dir>unbundled>object
answer yes -bf hpdl ([files **])
```

To shut the system down and continue to the next step type:

```
ame
shutdown
```

STEP 2: INSTALLATION TO BCE (EARLY) LEVEL

Place the new MR12.3 Mult tape on any convenient tape drive. Initialize and boot the MST.

The system will boot the bootload tape controller, if necessary, and continue. At this time you will be asked to input the RPV location in channel, controller, device and unit number format:

```
bootload_0: Booting t500 A 12.3. with mtc500 rev.v1 firmware.
bootload_0: Booted tape MPC.
0000.1 announce_chwm: 371. pages used of 512. in
                              wired environment.
0000.2 announce_chwm: 646. words used of 1024.
                              in int_unpaged_page_tables.
find_rpv_subsystem: Enter RPV data: M-> ! rpv a22 451 451 1
bce (early) 0012.0: M->
```

STEP 3: CHANGES TO CONFIG DECK

There are no config deck changes for MR12.3. To examine and change the config deck enter the config qedx editor by typing:

```
config
1,$p
```

to display the entire config deck as currently saved in the "conf" partition.

After having made necessary changes exit the editor by typing "w" and "q".

The "w" request will verify card syntax. The "q" request will leave the qedx editor and place the user at the bce(early) level.

STEP 4: INSTALLATION TO BCE (BOOT) LEVEL

Continue installation of BCE by typing:

```
bce (early) 0020.0: M-> ! bce
Current system time is:  Friday, June 7, 1974 00:20:46 mst
Is this correct?  no
Enter time: M-> ! 12/01/89__13:21:30
Current system time is:  Friday, December 01, 1989 13:21:30 mst
Is this correct? M-> ! yes
load_disk_mpcs:  Disk mpc(s): mspa mspc appear not to be
      operating.
Enter disk mpc names to be loaded, or "none" or "abort"
      or "all": M-> ! mspa mspc

(The operator entered the names of other disk mpcs
 to be loaded.)

hc_load_mpc:  Booting channel A20 with dsc500 Revision X1.
hc_load_mpc:  Booting channel B20 with dsc500 Revision X1.
bce (boot) 1325.5: M->
```

At this time, the operator must load firmware into all other controllers (i.e., not the bootload tape controller nor any disk controllers). bce is then considered to be fully initialized.

STEP 5: RING-1 ENVIRONMENT

The system is now ready to cross into the ring-1 environment:

```
bce (boot) 1325.5:  boot
Multics MR12.3 - 12/01/89 1328.0 mst Fri.
Command:
```

At Initializer ring-1 command level, type:

```
alv -all
```

At this point the libraries will be reloaded. The release tapes should be reloaded in this specific sequence: EXEC, LDD_STANDARD, UNB, and MISC.

To reload the libraries type:

```
reload_system_release -nosetlvid
```

When the system asks "Input tape label:" type: <reel_id>

where <reel_id> is the reel identification specified for the next tape to be reloaded. See Section 2, "Contents of MR12.3 Package," for a listing of all "Tape names". Enter <reel_id> for each library tape.

When all tapes are loaded, type:

```
standard
admin
<password>
```

STEP 6: TTF CONVERSION

The TTF is not changed for MR12.3. This step is retained as a place holder for future releases. Skip to step 7.

A new standard TTF has been provided containing additional terminal types. Sites using a modified site dependent TTF should merge these changes with their modified version, and convert the TTF to its binary version. The converted binary version must then be installed to take effect. The TTF.ttf segment in >tools contains some of the more common used terminals on Multics. Sites using the default TTF for this release must execute the following commands to perform this conversion:

```
cwd >udd>sa>a
rename TTF.ttf TTF.save
cp >t>TTF.ttf
cv_ttf TTF
install TTF.ttt
```

STEP 7: FNP CORE IMAGES AND CMF CONVERSION

A Multics Communications System (CS) core image is supplied in the >unbundled library, and is named "site_mcs". The "site_mcs" core image contains the basic support for DN6780 type FNPs with 64k of memory.

CS core images are built using the bind_fnp command in conjunction with a bindfile describing the CS modules and configurations to be used. A copy of the site_mcs.bind_fnp can be found in >ldd>mcs>info.

Sites should build their own CS core image tailored to their own FNP configuration, terminal type requirements, and use of

additional separately priced FNP software modules. Sites using the default "site_mcs" core image should skip to Step 8 after ensuring the CMF image statement points to the core image now located in >unb.

To build a new core image, the following procedure is suggested:

Sites will need to extract the communications object segments from archives located in >ldd>mcs>object. Sites should create a virgin directory under >udd>sa>a for each new core image.

The following example is for sites with the more common type Datanet and a larger memory configuration. Execute the following commands:

```
create_dir >udd>sa>a>mcs.7.6c
cwd >udd>sa>a>mcs.7.6c
qx
r >ldd>mcs>info>site_mcs.bind_fnp (or location of sites
                                   CS bind file)
.
.
.
make editing changes if any..
.
.
.
w site_mcs.bind_fnp
q
ac x ([segs >ldd>mcs>o>*.archive -absp])
bind_fnp site_mcs -list
```

Be sure the image statement in the CMF points to this newly created CS core image. The name of the CMF requires a suffix of "cmf". The following example assumes the CMF to be in the >udd>sa>a directory. This procedure will insure that the new CS image is used:

```
cwd >udd>sa>a

qx
r  CMF.cmf
.
.
1) Edit the image: statement to point to the new CS image.
2) Make any other changes needed.
.
.
w CMF.cmf
q

cv_cmf CMF
copy CMF.cdt >system_control_1>cdt -force
```

The above procedure builds a site dependent CS core image and ensures that this image is loaded in the FNP by the answering service.

Exit admin mode by typing "ame". Then issue the commands:

```
stop_mpx a          (sites with multiple FNPs execute this
                    command for each FNP)
multics

load_mpx a -check   (sites with multiple FNPs execute this
                    command for each FNP)
go
```

The load_mpx command indicates on the FNP console any configuration errors if console_man is loaded and "console: yes;" is in the bind_file. If any errors are reported they should be corrected.

Sites should assure the correct version number is reported the first time the FNP is booted.

STEP 8: ACLS AND RING BRACKETS

Check the ACLs for >sss>dm_admin_gate_, >sss>dm_daemon_gate_, >tools>installation_tools_, >tools>pnt_admin_gate_, >tools>pnt_login_gate_, >tools>pnt_network_gate_, >tools>pnt_priv_gate_, >sss>metering_gate_ and >sss>queue_admin_. The ACLs on these gates are as they appear on System M and should be restricted. The ACL for these gates are site dependent and should be changed to meet each site's needs. The dm_admin_gate_ and dm_daemon_gate_ should be restricted to data management administrators or daemons. The installations_tools_ gate should be restricted to system library maintainers. All persons on the ACL for metering_gate_ have access to the Multics metering data. All persons on the ACL for queue_admin_ are permitted to move absentee and daemon requests for themselves and other users to different queues. Users not on this ACL are only able to move their own requests. The Initializer must have access to queue_admin_. This capability is also dependent on extended access to the <queue>.ms segments. The various pnt_<name>_gate_ acls should be restricted to the following minimums for correct system operation:

```
  >t>pnt_admin_gate_
re    *.SysDaemon.*
re    *.SysAdmin.*

  >t>pnt_fs_gate_
re    *.*.*
```

```
  >t>pnt_login_gate_
re     Initializer.SysDaemon.*
re     *.SysAdmin.*

  >t>pnt_network_gate_
re     Initializer.SysDaemon.*
re     IMFT.Daemon.*
re     Card_Input.Daemon.*
re     *.SysDaemon.*

  >t>pnt_priv_gate_
re     Initializer.SysDaemon.*
r      *.SysDaemon.*
re     *.SysAdmin.*
```

STEP 9: SYSTEM CLEANUP

The system is now ready for normal operation under MR12.3. Following a reasonable period of MR12.3 operation, it is suggested that sites delete any <name.save> segments created as part of this procedure. It is also recommended that sites salvage their systems and fix quota by running Salvager.SysDaemon, executing the following command:

```
x repair salvquota > <number of salvagers> -rebuild -dcf
```

SECTION 5

INSTRUCTIONS FOR SITES INSTALLING FOR FIRST TIME

The following basic procedure must be performed when installing Multics for the first time.

STEP 1: PREPARATION

Ensure that all Multics active hardware components run error free in Multics mode using the latest T&D release. Peripheral equipment can be run in either Multics or GCOS mode and must also run error free.

Carefully check the hardware configuration (port and channel assignments, mailbox switch settings, etc.) Create and verify the configuration description on paper for later input when BCE is running. Close consultation between the SiteSA and Field Engineering representative is of the utmost importance. (Refer to Section 9 and Appendix A of the Operator's Guide to Multics, Order Number GB61, for hardware switch setting information. Refer to Section 7 of the Multics System Maintenance Procedures, Order Number AM81-03, for configuration setup.)

When selecting the storage unit for the RPV, select a disk unit with as few bad tracks as possible. For the MSS451s, T&Ds should be used to format and test the first disk to be used as the RPV (test 365, subtest 26).

The MSS500/501s are formatted at the factory, however, selection of alternate tracks is not done at the factory. It can only be done using MTR at Multics command level. (Refer to Appendix B for an outline of how MTR runs under Multics.)

STEP 2: LOGICAL VOLUME ASSIGNMENTS

Choose the logical volume assignments. Decide how many logical volumes are needed and how many physical volumes are to be in each.

Most installations have the following:

Logical Volume	Contents
root	>system_control_1 >system_library_standard >system_library_tools >system_library_unbundled >system_library_auth_maint >system_library_1 >documentation >daemon_dir_dir >dumps >system_library_tandd >system_library_obsolete >system_library_3rd_party >site >lv partitions
public	>user_dir_dir >library_dir_dir>include
ldd	>library_dir_dir

Other logical volumes may be set up for specific applications.

The assignment decision requires the system administrator to balance the costs of seek interference and breakage against the advantages of being able to define and process logically different collections of data. Data bases used for only a few hours a day or only a few days a month are natural candidates for allocation to a separate logical volume. Breaking up the system's storage into several logical volumes also allows the site to operate without all logical volumes mounted if hardware goes down. For example, an MPC or channel might go down, halving the system's disk drive capacity.

Logical volume assignments might be as follows:

Logical Volume	Contents
root	>system_library_tandd >library_dir_dir >system_library_obsolete >system_library_standard >system_library_tools >system_library_unbundled >system_library_3rd_party >daemon_dir_dir >documentation >dumps >system_library_1 >system_library_auth_maint >user_dir_dir>Daemon >user_dir_dir>SysAdmin >user_dir_dir>SysDaemon >user_dir_dir>SysMaint >site >lv partitions
Mcc	>user_dir_dir>Mcc
Multics_Pubs	>user_dir_dir>Pubs >user_dir_dir>Multics
Old_Dumps	>dumps>Old_dumps
Public	>user_dir_dir >experimental >process_dir_dir >ldd>include
list_1	>library_dir_dir>listings>hard >library_dir_dir>mcs >library_dir_dir>unbundled
list_2	>library_dir_dir>listings

These particular assignments give a wide range of flexibility and Multics can run with only the root logical volume mounted, or with one or two of the less critical logical volumes not mounted due to unavailability of disk drives. For example, logical volumes, list_1 and list_2, can easily be demounted. This frees two disk drives to be available for use with other more critical logical volumes.

Installations that wish to use the Access Isolation Mechanism (AIM) by specifying more than one access category (sensitivity level) should specify the maximum and minimum categories for one or more volumes and thus ensure that sensitive data is confined to a few packs, or that packs are not "contaminated" with information requiring special precautions.

The logical volumes that hold process directory segments must be chosen. Because of the heavy usage of process directory segments, these segments should be spread over as many physical volumes as possible. One or more logical volumes may be selected to hold process directory segments, using the set_pdir_volumes command in system_start_up.ec. In the supplied system_start_up.ec, a single logical volume, named public, is selected. This command line should be changed to select a set of publicly accessible and permanently mounted logical volumes containing as many physical volumes as possible, subject to some constraints. Site maintenance personnel are responsible for ensuring there is always enough space available on the selected logical volumes to hold the process directory segments. The process directory placement algorithm causes process directory creations to be made on each logical volume in proportion to the number of physical volumes in the logical volume.

Ensure that enough storage will be available. About 5% of each volume is used for the VTOC and volume map. In addition, some breakage is unavoidable.

Since the system handles running out of storage without crashing, and since it is possible to add physical volumes to a logical volume dynamically, logical volumes can be defined with fewer physical volumes than their maximum anticipated size.

STEP 3: RPV INITIALIZATION

Mount the Multics System Tape (MST) on Magnetic Tape Handler (MTH) nn (nn is usually equal to 01). Mount the disk pack formatted by T&D on the drive selected to be the RPV. Initialize and boot the MST. Multics will prompt with:

```
bootload_0:  Booting system MR12.3 generated 08/31/89 0000.0
bootload_0:  Enter boot tape MPC model:  ! t500
```

Normal response to this question should be "ipc", "t610", "t601" or "t500". However, on systems with an IMU configured the "Enter boot tape..." query will not appear. The system will boot the bootload tape controller, if necessary, and continue. At this time, the intention to cold boot is given. Multics will request the location of the rpv. Once this is done, the init_vol request loop will be entered to accept the layout of the rpv.

```
bootload_0:  Booting t500 A 12.3.  with mtc500 rev.u1 firmware.
```

```
bootload_0:  Booted tape MPC.
0000.1 announce_chwm: 371. pages used of 512. in
                              wired environment.
0000.2 announce_chwm: 646. words used of 1024.
                              in int_unpaged_page_tables.
find_rpv_subsystem: Enter RPV data: M-> ! query
find_rpv_subsystem: Enter RPV subsystem base channel, as Icc,
                              or "cold". M-> ! cold
find_rpv_subsystem: Booting cold will destroy all data on the RPV
   Are you sure that you want to boot cold?  M-> ! yes
find_rpv_subsystem: Enter RPV subsystem base channel,
                              as Icc. M-> ! a22
find_rpv_subsystem: Enter RPV subsystem MPC model: M-> ! 451
find_rpv_subsystem: Enter RPV disk drive model: M-> ! 451
find_rpv_subsystem: Enter RPV drive device number: M-> ! 1
find_rpv_subsystem: RPV is a model 451 drive, number 1 on MPC A22
                              (Model 3), and this is a COLD boot.
    Is this correct? M-> ! yes

Default RPV layout:  (Respond "end" to use it.)
Average seg length = 2.00
VTOC size = 2792 pages, 13920 vtoces.
27840 paging records.
Constrained by average seg length.
part hc 2792.  2500.
part conf 5292.  4.
part alt 38117.  141.          (451 disk drive only)
part dump 35847.  2000.
part log 35591.  256.
part file 35336.  255.
part bce 33136.  2200.
request: M-> ! end
```

These are the default partition assignments. Any changes to the default partitions or RPV parameters can be redefined by using the "startover" request in init_vol. The system installer should review the write-up of init_vol in the Multics Administration, Maintenance, and Operations Commands Manual, Order Number GB64-00, prior to the installation.

Sizes for the various partitions and their locations can be modified based on the needs of the site.

```
init_empty_root:  Begin rpv initialization.  This will
     take some time.
init_empty_root:  rpv initialized; 27840 records.
bce (early) 0012.0: M->
```

The list above is based on a 451 disk drive. With the exception of some default values the script will be the same for 501, 3380, or 3381 disk units.

STEP 4: CONFIGURATION

Build the configuration description as follows (user input preceeded by an exclamation mark (!):

```
! config
! 1,$d
! a
! [User types in configuration fields as defined
   in the System Maintenance Procedures, Order
   Number AM81-03]
! \f
! w
! q
```

Do not enter any part cards at this time, except for those partitions defined on the rpv. Also, make the root card specify only the rpv.

Continue booting bce.

```
bce (early) 0020.0: M-> ! bce
Current system time is:  Monday, April 29, 1985 00:20:46 mst
Is this correct?  no
Enter time: M-> ! 12/01/89__13:21:30
Current system time is:  Friday, December 01, 1989 13:21:30 mst
Is this correct? M-> ! yes
load_disk_mpcs:  Disk mpc(s): mspa mspc appear not to be
      operating.
Enter disk mpc names to be loaded, or "none" or "abort"
      or "all": M-> ! mspa mspc

(The operator entered the names of other disk mpcs
 to be loaded.)

hc_load_mpc:  Booting channel A20 with dsc500 Revision X1.
hc_load_mpc:  Booting channel B20 with dsc500 Revision X1.
bce (boot) 1325.5: M->
```

At this time, the operator must load firmware into all other controllers (i.e., not the bootload tape controller nor any disk controllers). bce is then considered to be fully initialized.

```
bce (boot) 1325.5 : M-> ! boot -cold
Do you really wish to boot cold and there by destroy the
     system hierarchy? M-> ! yes
1326.1 volume_registration_mgr_$check_volume_registration:
        Reregistered public LV root LVID 133353533031
        (Initializer.SysDaemon.z)
1326.3 volume_registration_mgr_$check_volume_registration:
        Reregistered PV rpv PVID 133353533017 in LV root
        (Initializer.SysDaemon.z)
disk_table_: New disk_table created
```

```
Multics MR12.3 - 12/01/89 1327.0 mst Fri.
Command: M->
```

Ignore the messages prefaced by disk_table_ and volume_registration_mgr_.

STEP 5: INITIALIZING ROOT VOLUMES

Initialize each new root volume except the RPV with the init_vol command.

For better performance, it is advisable to place a hardcore partition (hc) on each physical volume of the Root Logical Volume (RLV). The placement of the hardcore partition on each volume must be low. The recommended size of additional partitions is 2500 records divided by the number of physical volume used. The RPV size should remain 2500 records to allow the system to boot with only an RPV mounted.

For most volumes the command looks like:

```
init_vol PV_NAME DRIVE_NAME  -rlv {-special}

Example: init_vol root2 dska_02 -rlv -special
```

For those volumes that are to have partitions, or an average segment length other than the default of five records, add the optional "-special" as a third argument. The command then asks for instructions about the partition location. Hardcore partitions, for additional root volumes, should be specified as they are initialized. You may type one or more of the following:

```
part NAME low nrec
part NAME high nrec
avg fff.ff
list
```

complete initialization by typing:

```
end
```

An example of typing the init_vol for an MSS0451/400 with an alternate partition on a RLV drive is:

```
init_vol root2 dska_02 -rlv -special
part alt high 141     (Note: 451 disk only)
part hc low 625       (Example: 2500/4 root volumes)
end
```

when done type:

```
shut
```

STEP 6: ADDITIONAL CONFIGURATION PARAMETERS

At bce (boot) level enter the configuration deck editor by typing "config". The PART cards and ROOT card should be added to the deck. Subsequent boots divide the hardcore supervisor among all hardcore partitions.

The following script is provided as an example where a root card exists in the configuration deck and a part card does not exist. Parameters of cards will vary according to the configuration of individual sites. User input is preceeded by an exclamation mark (!).

```
bce (boot): M-> ! config
M-> ! /root/
root -subsys dski -drive 1
M-> ! s/$/ -subsys dski -drive 2/p
root -subsys dski -drive 1 -subsys dski -drive 2
M-> ! /part/
Search failed.
M-> ! a
M-> ! part bos dski 1
M-> ! part log dski 1
M-> ! part dump dski 1
M-> ! \f
M-> ! w
M-> ! q
bce (boot) 1215.2: M-> reinit
```

STEP 7: RELOAD OF EXECUTABLE LIBRARIES

Do a normal boot "BOOT". While at ring-1 initializer command level load the executable libraries. This is done as follows:

```
bce (boot): ! boot
Command: M-> ! reload -nosetlvid
```

Only the system libraries (MR12.3.EXEC) should be reloaded at this time. The -nosetlvid control argument ignores the logical volume ID on the tape when a directory is being reloaded.

```
M-> ! shut
1230.1 shutdown complete.
bce (boot) 1231.1: M-> ! boot standard
                    (ignore the messages from sc_init_.)
Multics MR12.3 - 12/01/89 1235.2 mst Fri
Ready
M-> ! admin
admin: Entry not found.  Could not retrieve admin password
  from the PNT to check admin password.  Entering admin mode.
   [NOTE: This error message is repeated each time admin is
          entered until a password has been set.]
```

Register and initialize all non-RLV volumes. For ease of typing, use of lower case names is recommended.

Use the add_volume_registration (avr) command as in the following example:

```
! avr -pv pub01 -lv public -serial 233-81 -model 451
add_volume_registration: LV "public" does not exist.  Do you
     wish to create it? M-> ! yes
add_volume_registration: Registered PV "pub01" (pvid
     100172223140) on new LV "public" lvid 100172223005).
r 14:15  1.473 8
```

to create registration entries for each logical and physical volume. The registration file for the root logical volume is created automatically by the bootload. Since the default model number is 451, use the change_vol_registration command, if necessary, to set the correct value of model number on the rpv. The serial number can also be set as follows:

```
lvr -pv rpv
cvr -pv rpv -serial 233-79 -model <model number>
ame
```

Use the init_vol for the additional logical volumes as follows:

```
init_vol pub01 dska_03 -special
part alt high 141    (Note: 451 disk only)
end
init_vol pub02 dska_04 -special
part alt high 141    (Note: 451 disk only)
end
```

After all physical volumes are registered and initialized, add them to the disk_table by typing the add_vol (av) command for all except the RPV:

```
av pvname dskX_NN
```

An example:

```
av pub01 dska_03
```

At this point add all of the logical volumes by typing:

```
alv -all
```

STEP 8: SETTING AND CHECKING ACCESS

The ACL for >lv should be set to "s" for all users. Setting initial ACLs for segments in the >lv directory is done from admin mode by typing:

```
M-> ! admin
<READY MESSAGE>
M-> ! sis >lv rew *.SysAdmin rew *.SysDaemon
<READY MESSAGE>
M-> ! sa >lv s * sma *.SysAdmin sma *.SysMaint
<READY MESSAGE>
```

Create Access Control Segments (ACS) for each logical volume. For system public volumes, create them as follows:

```
M-> ! create >lv>{lvname}.acs
<READY MESSAGE>
M-> ! cvr -lv {lvname} -acs >lv>{lvname}.acs
<READY MESSAGE>
M-> ! set_max_length >lv>*.acs 0
```

where {lvname} stands for name of each logical volume.

The ACLs of these segments are interpreted to give permission to attach the logical volume (for private volumes) and permission to modify master directory control information in the MDCS (for specific logical volumes). Specific ACL entries for Initializer.SysDaemon should be deleted at this time by typing:

```
delete_acl >lv>*.acs <READY MESSAGE>
```

This is necessary because Initializer.SysDaemon always gets default access of "rw". This would prevent the Initializer from being a volume administrator by virtue of the missing "e" access. Deletion of specific access gives the Initializer the "rew" access allowed all SysDaemons.

The ACL is now set so that all system administrators and all SysDaemons are volume administrators. The "e" bit controls executive access.

For private volumes, the ACS is in a directory controlled by the volume owner. The ACS segment must reside in a directory on a logical volume different from the private logical volume.

STEP 9: SETTING VOLUME QUOTA

Use the set_volume_quota command to give the Initializer process enough quota on each logical volume to create the necessary master directories.

```
set_volume_quota LV_NAME QUOTA
```

Example: set_volume_quota public 36000

The number QUOTA should be at least the total of the quotas of the directories to be created in the next step.

Use create_dir to create master directories. The format of the command is:

```
create_dir pathname -lv logical_volume -quota QQ
```

where QQ <262144

```
Example:  cd >library_dir_dir -lv no_backup -quota 40000
          cd >library_dir_dir>include -lv public -quota 3000
```

The acct_start_up.ec will, in step 11, create a number of project directories and assign terminal quota if the directories do not exist. They are the following with the quota that will be assigned:

>udd>SysAdmin	5000
>udd>SysAdmin>admin	2000
>udd>SysDaemon	5000
>udd>Daemon	1000
>udd>Operator	100
>udd>Terminals	10
>udd>HFED	5000

Set ACLs and additional names, as desired, on all master directories at this time.

Directory quota should be set for each master directory by those sites that wish to have disk charges for directory pages included in monthly bills. Those sites not interested in implementing this feature may type "ame" and "shutdown" then skip to the next step.

A directory quota of 1000 pages should be sufficient for all master directories with the exception of >udd. If udd is a master directory it is recommended that it be given a directory quota of 100000 pages. This provides enough directory quota for 100 projects at 1000 pages each. If the site has more than 100 active projects the 100000 figure should be adjusted accordingly. To set directory quota on each master directory execute the following command:

```
set_dir_quota PATHNAME QQ
```

This command allows a system administrator to place an arbitrary secondary storage quota for directories on a specified directory.

PATHNAME
is the name of a directory on which the directory quota is to be placed. -wd can be used to specify the working directory.

QQ
is the directory quota in 1024 word pages.

If additional directory quota is required for a master directory the quota can be reset following movement of directory quota to inferior directories.

Instructions for moving directory quota down to the project level is included in Step 18.

To shut down do the following:

```
ame
shut
```

STEP 10: RELOAD OF REMAINING RELEASE TAPES

Reboot Multics to ring-1 and reload the MR12.3.LDD_STANDARD and MR12.3.UNBUNDLED tapes with the following commands:

```
boot
alv -all
reload -nosetlvid
```

The tape labeled MR12.3.MISC must be the final tape of the MR12.3 supplied set to be reloaded.

STEP 11: RUNNING ACCT_START_UP.EC

After all the release tapes have been reloaded cross into ring-4 by executing the following commands:

```
standard
admin    [Ignore messages from admin at this time.]
```

At this time you are ready to execute part 1 of the acct_start_up.ec. To do this type:

```
ec >system_library_tools>acct_start_up cold F.ANSS
```

where F.ANSS is the channel number of the hardwired Initializer terminal.

```
 F = FNP number (a-h)
 A = Adaptor type (h = hsla)
 N = Adaptor number (0-2 for hsla)
SS = Decimal subchannel number of specified adaptor
```

The string "F.ANSS" should be replaced by "otw_" if there is no hardwired terminal and the bootload console is to be used as the Initializer terminal.

The error, "new_iod_tables_compiler: Entry not found. Accessing cdt. Channel name checks will not be performed", may be encountered during this stage. This error should be disregarded.

At this point, tapes dumped from other Multics sites can be reloaded as desired using the "reload" command with the control arguments "-noquota -notrim -nosetlvid" to avoid deletion of existing segments and resetting of quotas. If any segments are to be loaded into ring 1 then it cannot be done without exiting admin mode and rebooting to ring 1.

STEP 12: MULTICS COMMUNICATIONS SYSTEM

A Multics Communications System (CS) core image is supplied in the >unbundled library, and is named site_mcs.

The site_mcs version for this release is 7.6. The site_mcs core image contains the basic support for DN6780 type FNPs with 64k of memory.

The communication system core images are built using the bind_fnp command in conjunction with a bindfile describing the CS modules and configurations to be used. A copy of the site_mcs.bind_fnp can be found in >ldd>mcs>info.

Sites should build their own CS core image tailored to their own FNP configuration, terminal type requirements, and use of additional separately priced FNP software modules.

Sites using the default site_mcs core image should ensure the CMF image statement points to the correct default core image located in >unb. The initial CMF, which includes some sample channel entries in comments, as well as one FNP entry, should be checked. Edit this CMF to eliminate any inconsistencies with the actual configuration and add one or more entries for login channels.

Sites modifying their own CS core image are required to use the GCOS Environment Simulator which is an unbundled software product. To build a new core image, the following procedure is suggested:

Sites will need to extract the communications object segments from archives located in >ldd>mcs>object. Sites should create a virgin directory under >udd>sa>a for each new core image.

The following example is for sites with the more common type Datanet and a larger memory configuration. Execute the following commands:

```
create_dir >udd>sa>a>mcs.7.6c
cwd >udd>sa>a>mcs.7.6c
ted
r >ldd>mcs>info>site_mcs.bind_fnp (or location of sites
                                   CS bind file)
.
.
.
make editing changes if any..
.
.
.
w site_mcs.bind_fnp
q
ac x ([segs >ldd>mcs>o>*.archive -absp])
bind_fnp site_mcs -list
```

Be sure the image statement in the CMF points to this newly created CS core image. The following example assumes the default CMF to be in the >udd>sa>a directory. This procedure will insure that the new CS image is used:

```
cwd >udd>sa>a
ted
r  CMF.cmf
.
.
Locate the image: statement by typing:
```

```
/image/

Edit the image: statement to point to the CS image,
site_mcs, by making the statement read:

:image:  >udd>sa>a>mcs.7.6c>site_mcs;

.
.
Make any other changes needed.
.
.
w CMF.cmf
q
cv_cmf CMF.cmf
copy CMF.cdt >sc1>cdt -force
```

The above procedure builds a site dependent CS core image and ensures that this image is loaded in the FNP by the answering service.

STEP 13: COMPLETE ACCOUNTING STARTUP

During this step, expect many messages, some with audible alarms, reporting that certain segments do not exist and are being created. These messages would be cause for concern during normal system operation but are to be expected during accounting start up and may be ignored. Execute the following example:

```
ame
stop_mpx a  (sites with multiple FNPs execute this
             command for each FNP)
multics
load_mpx a -check  (sites with multiple FNPs execute
                    this command for each FNP)
admin    (any error messages displayed at this time, except
          hardware error messages can be ignored.)

ec >tools>acct_start_up cold2
```

This procedure will finish accounting start up. The load_mpx command indicates on the FNP console any configuration errors if console_man is loaded and "console: yes;" is in the bind_file. If any errors are reported they should be corrected.

During this stage you will encountered the error, "set_max_length: Validation level not in ring bracket. >system_control_1>mcaa.acs", which occurs if an IMU is configured. Ignore this error message. The cause of this error, validation level of >sc1>mcaa.acs, must be fixed manually after cold2 has completed successfully.

A default start_up.ec is available for use by new Multics users when they first log in. This exec_com is executed by users who login to the system without their own start_up.ec. The segment >tools>start_up.ec was copied into >sc1 by the acct_start_up.ec. Individual sites can modify this exec_com to meet their own needs. The access for the segment should be "r *.*.*" and ring brackets of 4,5,5. To start the system up for normal service type:

```
ame
word login
abs start
go
```

After typing "go" a number of messages will be returned. These messages are of the form:

```
absentee_utility_: Entry not found. Creating new <pathname>.
scavenge_vol: No volumes.
Found unexpected command_error in system start_up.ec.
```

These messages may be ignored.

STEP 14: CHECK GATE ACLS

Type "admin" and enter the new admin password you selected. Check the ACLs for >sss>dm_admin_gate_, >sss>dm_daemon_gate_, >tools>installation_tools_, >tools>pnt_admin_gate_, >tools>pnt_login_gate_, >tools>pnt_network_gate_, >tools>pnt_priv_gate_, >sss>metering_gate_ and >sss>queue_admin_. The ACLs on these gates are as they appear on System M and should be restricted. The ACL for these gates are site dependent and should be changed to meet each site's needs. The dm_admin_gate_ and dm_daemon_gate_ should be restricted to data management administrators or daemons. The installations_tools_ gate should be restricted to system library maintainers. All persons on the ACL for metering_gate_ have access to the Multics metering data. All persons on the ACL for queue_admin_ are permitted to move absentee and daemon requests for themselves and other users to different queues. Users not on this ACL are only able to move their own requests. The Initializer must have access to queue_admin_. This capability is also dependent on extended access to the <queue>.ms segments. The various pnt_<name>_gate_ acls should be restricted to the following minimums for correct system operation:

```
  >t>pnt_admin_gate_
re    *.SysDaemon.*
re    *.SysAdmin.*
```

```
  >t>pnt_fs_gate_
re    *.*.*

  >t>pnt_login_gate_
re    Initializer.SysDaemon.*
re    *.SysAdmin.*

  >t>pnt_network_gate_
re    Initializer.SysDaemon.*
re    IMFT.Daemon.*
re    Card_Input.Daemon.*
re    *.SysDaemon.*

  >t>pnt_priv_gate_
re    Initializer.SysDaemon.*
r     *.SysDaemon.*
re    *.SysAdmin.*
```

Set ACLs on the >sc1>rcp directory and on the access control segments in it (<name>.acs), to allow users to attach tape drives and any other peripherals they are allowed to use.

After all ACLs are set, type:

```
ame
x repair salvquota > 2 -dcf -rebuild
```

STEP 15: SAVE CHECKPOINT

Type "logout * *" and "shutdown". After a successful shutdown, do a BCE save. For further information see >doc>ss>bce>save.info or Appendix B of this document. Use fresh tapes for the BCE save so that the results of the above steps are not lost.

STEP 16: TAILOR SYSTEM AND REGISTER PROJECTS

Following the bce save, reboot the system for normal service by typing "boot star". The system is now ready for registration of projects and users from a SysAdmin process, such as Repair.SysAdmin, logged in from a standard terminal.

The acct_start_up exec_com created default system_start_up.ec, admin.ec, iod_tables.iodt, RTMF.rtmf, and CMF.cmf segments. These segments should be tailored by the local Site SA to meet site operational and configuration requirements.

STEP 17: SETUP VOLUME BACKUP/RELOADER

The following instructions are necessary only for those sites that intend to use the Volume Backup/Reloader facility:

The personids "Volume_Dumper", "Volume_Reloader", and "Volume_Retriever" are registered. These personids are registered on the Daemon project with the multip and daemon attributes and with a home_dir of >user_dir_dir>Daemon>Volume_Dumper. Sites using AIM must set the authorization for these personids at system_high and upgrade the home_dir at system_high.

Login Repair SysDaemon, or if running in special session using the Initializer, execute the following commands:

```
admin

ec >tools>setup_volume_reloader
```

The error message that Volumes dm001 and dm002 are not registered should be ignored. This exec_com creates all directories, segments, and message segments necessary for running the volume dumper/reloader system. This exec_com also sets suggested access on the directories and segments created. Not all the access set is required. If a site wishes, the access created for *.SysMaint.* and *.SysAdmin.* may be removed.

This exec_com resets the vtoce fields for both incremental and consolidated dumps by making a first dump pass with output to discard_. This is necessary since the first dump pass is equivalent to a complete dump on both the incremental and consolidated pass. Follow the instructions for normal use of this facility at the completion of this exec_com.

Sites need a sufficient number of tapes to accommodate the entire file system and any incremental and consolidated dumps until a subsequent complete dump is taken. This is known as a reload group. It is suggested that new sites start with 100 reels of tape or a sufficient quantity to contain two complete reload groups. A single reel of tape at 6250 bpi holds approximately 26000 Multics records.

STEP 18: SETUP DIRECTORY QUOTA

This step is necessary only for those sites that wish to charge their user projects for disk storage used by directory pages, or to obtain a more complete disk report containing additional disk usage statistics. If directory quota is not already set on >udd execute the following commands from a SysAdmin process if udd is not a master directory:

```
sac set_dir_quota > 120000
sac move_dir_quota >udd 100000
```

If udd is a master directory, then execute:

```
set_dir_quota >udd 100000
```

Then execute the commands:

```
cwd >udd
move_dir_quota ([dirs **]) 1000
```

These commands move or set sufficient directory quota on udd for 100 projects with the suggested default project directory quota of 1000. If a site has more than 100 active projects a figure in excess of 100000 must be chosen for the initial directory quota of udd. The master.ec gives each new project a default directory quota of 1000 pages by moving 1000 pages of directory quota from udd. The system administrator should make sure there is always sufficient directory quota on udd to accommodate new projects.

It is also suggested that all directories directly off the root with the exception of pdd and sl1 be given nonzero segment and directory quotas large enough to accommodate their current page usage and allowing for some growth. The purpose of this is to cause the disk report to contain complete statistical information

on these directories (directories with 0 quotas are omitted from the disk report).

APPENDIX A

MODULE CHANGES FOR MR12.3

This appendix provides information about changes to the Multics operating system on a segment basis. The following information applies to changes made to the system for the MR12.3 release only.

This appendix is provided to help sites identify, to a segment level, changes they may have applied for emergency bug fixes or site dependent modifications to current system software.

Information is catagorized as follows:

new info segments (added to >doc>info unless otherwise noted)
new segments
new or modified include files
deleted segments
modified segments

NEW INFO SEGMENTS FOR MR12.3

alm.info
calc.info
check_iacl.info
copy.info
copy_dir.info
copy_dir_.info
copy_seg_.info
display_mailing_address.info
emacs.changes.info
errata.ak50.info
errata.am82.info
errata.at58.info
errata.cj52.info
find_source_file_.info
fortran.info
manuals.gi.info
memo.info
move.info
move_dir.info
mowse_io_.info
pl1.info
pl1.info
translator_info

NEW SEGMENTS FOR MR12.3

protocol_mpx.pl1

NEW OR MODIFIED INCLUDE FILES FOR MR12.3

backup_volume_log.incl.pl1
builtin_table.incl.pl1
cmcs_entry_dcls.incl.pl1
cmtv.incl.pl1
cobol_addr_tokens.incl.pl1
cobol_fsb_type_1.incl.pl1
cobol_fsbskel.incl.pl1
create_branch_info.incl.pl1
create_branch_info.incl.pl1
disk_error_interp.incl.pl1
dn355_data.incl.pl1
language_utility.incl.pl1
mowse_io_data.incl.pl1
multiplexer_types.incl.pl1
oc_data.incl.pl1
protocol_data.incl.pl1
protocol_infos.incl.pl1
protocol_infos.incl.pl1
protocols.incl.pl1
x25_data.incl.pl1

DELETED SEGMENTS FOR MR12.3

There are no deleted modules for MR12.3.

MODIFIED SEGMENTS FOR MR12.3

PNOTICE_exec
absadr
absolute_pathname_
acc_list_
acc_name_
accept_fs_disk
accept_rpv
access_audit_
access_audit_check_ep_
access_audit_log_fault_
access_audit_r1_
access_audit_util_
access_class_check
access_mode
access_operations_
access_viol
acl
acl_
act_proc
activate
active_fnc_err_
add_key
add_mail_table_entry
add_scu
adjust_count
adjust_float_
admin_
adopt_seg
adopt_seg
aim_check_
aim_util_
alloc_
alloc_semantics
allocate_dir_ht_
alm
alm_syserr_caller
announce_chwm
ansi_tape_io_
any_to_any_
append
arc_sine_
arc_tangent_
asd_
assign_
assign_op
asu_
asum_data_
attribute_parse
audit_
authenticate_
backup_control_mgr_
backup_load
backup_load_dir_list
backup_map_
backup_util
basic_
bcd_to_ascii_
bce_abs_seg
bce_alert
bce_alm_die
bce_appending_simulation
bce_check_abort
bce_command_processor_
bce_component_to_wordnum_
bce_console_io
bce_continue
bce_copy_disk
bce_create_sstnt
bce_data
bce_die
bce_display_disk_label
bce_display_instruction_
bce_display_scu_
bce_dump
bce_error
bce_esd
bce_exec_com_
bce_exec_com_input
bce_execute_command_
bce_fwload
bce_get_defptr_
bce_get_flagbox
bce_get_to_command_level
bce_inst_length_
bce_ioi_post
bce_list_requests_
bce_listen_
bce_lock_mca
bce_map_over_requests_
bce_name_to_segnum_
bce_parse_disk_spec
bce_probe
bce_query
bce_query_af
bce_ready
bce_request_table_
bce_save
bce_save_util_
bce_severity
bce_shutdown_state

bce_state
bce_test_disk
bfp_to_hfp_
bk_arg_reader_
bk_input
bk_retrieve
bk_ss_
boot_rpv_subsystem
boot_tape_io
bootload_0
bootload_abs_mode
bootload_console
bootload_disk_io
bootload_disk_post
bootload_dseg
bootload_early_dump
bootload_error
bootload_faults
bootload_flagbox
bootload_formline
bootload_fs_
bootload_fs_cmds_
bootload_info
bootload_io
bootload_linker
bootload_loader
bootload_qedx
bootload_slt_manager
bootload_tape_fw
bound_as_mpx_.bind
bound_audit_.bind
bound_plio2_.bind
bound_priv_mpx.bind
bound_teco_.bind
bound_temp_1
boundfault
buddy_alloc_
buddy_area_
buddy_area_assign_
buddy_freen_
builtin
cache_priv
cache_tester
calc
call_bce
call_ec_
call_math_error_
call_outer_ring_
cam_cache
cancel_cobol_program
canon_for_volume_label_
canon_resource_name_
carry_dump
cat_op
change_dtem
channel_comm_meters
channel_manager
char_bit_offset_fcns_
check_entryname_
check_gate_acl_
check_iacl
check_trailer
chname
clock_
cm_compact
cmcs_create_queues_
cmcs_date_time_
cmcs_decode_status_
cmcs_expand_tree_path
cmcs_expand_tree_path_
cmcs_fillin_hdr_
cmcs_initiate_ctl_
cmcs_print_
cmcs_purge_queues_
cmcs_queue_ctl_
cmcs_scramble_
cmcs_set_lock_
cmcs_station_ctl_
cmcs_status_list_ctl_
cmcs_terminal_ctl_
cmcs_tree_ctl_
cmcs_wait_ctl_
cmtv
cobol
cobol_FILE_
cobol_FP_
cobol_IVAL
cobol_MSORT_
cobol_NUMS_
cobol_RELEASE_
cobol_RETURN_
cobol_SET_
cobol_SORTM_
cobol_accept_gen
cobol_add
cobol_add2_binary_long
cobol_add2_binary_short
cobol_add3
cobol_add_binary_gen
cobol_add_gen
cobol_addr
cobol_allo_tm
cobol_alloc
cobol_alter_gen
cobol_alter_perform
cobol_arg_descriptor

cobol_arith_move_gen
cobol_arithop_gen
cobol_bin_const_ck
cobol_binary_check
cobol_blank_stripper_
cobol_build_resop
cobol_c_list
cobol_call_gen
cobol_call_op
cobol_cancel_gen
cobol_ci_phase
cobol_close_gen
cobol_cmnio
cobol_compare_gen
cobol_compare_values
cobol_compute_bin_gen
cobol_compute_gen
cobol_control_
cobol_db
cobol_db_phase
cobol_ddact1
cobol_ddact2
cobol_ddalloc
cobol_ddst
cobol_ddsyntax
cobol_decl_gen
cobol_def_init
cobol_def_util
cobol_delete_gen
cobol_delete_tokens
cobol_diag_table_
cobol_disable_gen
cobol_display_gen
cobol_display_text
cobol_display_util
cobol_divide_bin_gen
cobol_divide_gen
cobol_enable_gen
cobol_end_gen
cobol_error_
cobol_exit_gen
cobol_exp3
cobol_expand_source_
cobol_ext_
cobol_ext_ddsyn
cobol_ext_lex
cobol_file_util
cobol_find_secdef
cobol_fix_driver_
cobol_fixup
cobol_fofl_mask
cobol_gen_driver_
cobol_gen_error
cobol_gen_ioerror
cobol_generate_report
cobol_get_index_value
cobol_get_num_code
cobol_get_size
cobol_gns
cobol_gns1
cobol_go_gen
cobol_ided
cobol_idedsyn
cobol_imp_word
cobol_init_
cobol_initstatic
cobol_insert_token
cobol_inspect_gen
cobol_io_
cobol_io_util
cobol_iocall
cobol_ioerror
cobol_ioerror_abort
cobol_iomode
cobol_ioop_util
cobol_lex
cobol_lexeme
cobol_lexerr
cobol_linage
cobol_link_init
cobol_load_register
cobol_make_bin_const
cobol_make_fsb_link
cobol_make_link
cobol_make_list
cobol_make_merge_file
cobol_make_object_map
cobol_make_reg_token
cobol_make_type3
cobol_make_type9
cobol_make_xref_
cobol_mcs
cobol_mcs_
cobol_merge
cobol_merge_gen
cobol_move_gen
cobol_mpy
cobol_mpy3
cobol_ms_handler
cobol_mst
cobol_multiply2_binary
cobol_multiply_bin_gen
cobol_multiply_gen
cobol_ntio
cobol_num_to_udts
cobol_opch_init

cobol_opch_is
cobol_opch_op_call
cobol_open_gen
cobol_open_util
cobol_operators_
cobol_output_tokens
cobol_paragraph_gen
cobol_paste
cobol_patch
cobol_pd_code
cobol_pdst
cobol_pdstax
cobol_perform_gen
cobol_pic_val_comp
cobol_pointer_register
cobol_pool
cobol_print_diag
cobol_proc_copy
cobol_proc_sort
cobol_process_error
cobol_profile
cobol_prologue_gen
cobol_purge_gen
cobol_r_initialize
cobol_read_ft
cobol_read_gen
cobol_read_rand
cobol_receive_gen
cobol_reg_manager
cobol_register
cobol_register_util
cobol_release_gen
cobol_reloc
cobol_repl3
cobol_repl3_expand
cobol_replb
cobol_report_writer
cobol_res_words
cobol_reset_r
cobol_reswd_table_
cobol_return_gen
cobol_rewrite_gen
cobol_rts_
cobol_rts_handler_
cobol_search_gen
cobol_section_gen
cobol_seginit_gen
cobol_send_gen
cobol_set_fsbptr
cobol_set_gen
cobol_set_pdir
cobol_set_pr
cobol_set_type40
cobol_short_to_longbin
cobol_sort_gen
cobol_sort_util
cobol_source_formatter_
cobol_start_gen
cobol_stop_gen
cobol_store_binary
cobol_string
cobol_string_gen
cobol_su_
cobol_subtract_gen
cobol_sym_init
cobol_syntax_trace_
cobol_trans_alphabet
cobol_unstring
cobol_unstring_gen
cobol_usrwd
cobol_version
cobol_write_gen
collect_free_core
com_err_
comm_meters_
command_processor_
command_query_
commit
compare_declaration
condition_
config_
config_data_
config_deck_data_
config_deck_edit_
config_deck_parse_
configure_test_cpu
connection_list_manager_
constant_token
context
context_processor
convert
convert_access_class_
convert_access_operation_
convert_binary_integer_
convert_date_to_binary_
copy
copy_
copy_
copy_dir_
copy_expression
copy_fdump
copy_on_write_handler_
copy_pdir_
copy_stack_0
copy_unique_expression
core_queue_man

correct_qused
countervalidate_label_
cp_data_
cplx_dec_ops_
crawlout_default_handler_
create
create_block
create_homedir_
create_hproc
create_ips_mask_
create_root_dir
create_root_vtoce
create_rpv_partition
create_vtoce
create_wordlist
cu_
cv_bin_
cv_config_card_
cv_dec_
cv_error_
cv_float_
cv_fsdisk_error_
cv_fstime_
cv_integer_string_
cv_rcp_attributes_
data_list_parse
date_name_
date_time_
dbm_man
dbr_util_
dc_find
dctl
deact_proc
deactivate
deactivate_for_demount
deactivate_segs
debug_check
dec_ops_
deckfile_manager
declare
declare_constant
declare_descriptor
declare_integer
declare_label
declare_parse
declare_picture
declare_picture_temp
declare_pointer
declare_structure
declare_temporary
decode_descriptor_
default_error_handler_
default_parse
define_area_
defined_reference
del_dir_tree
delentry
delete
delete_key
delete_record
delete_segs
delete_volume_log
delete_vtoce
demand_deactivate
demount_pv
descriptor_parse
device_control
dial_ctl_
dir_dump
dir_lock_init
disk_control
disk_emergency
disk_error_data
disk_init
disk_name_pvtx
disk_reader
disk_rebuild
disk_rebuild_caller
disk_run
disk_table_
display_access_class_
display_cobol_run_unit
display_disk_label_
display_psp
display_pvolog
display_volume_log
dm_log_
dmpr_arg_reader_
dmpr_finish_
dmpr_log_
dmpr_output_
dn355
dn355_boot_interrupt
dn355_messages
dn355_util
do_parse
do_semantics
double_arc_sine_
double_arc_tangent_
double_exponential_
double_logarithm_
double_principal_angle_
double_sine_
double_square_root_
double_tangent_
dsc_probe_requests_

dsc_requests_
dsc_sm_requests_
dsp1
dsp2
dsp3
dsp_DSA_LETTER_
dsptm_
dspuft_
dump_cdt_
dump_cmf_
dump_volume_
e_defpl1_
e_mail_pl1_
e_redisplay_
ebcdic8_to_ascii_
ebcdic_to_ascii_
edx_util_
emacs_rmail_
encode_clock_value_
enter_abs_request_
equal
error
error_
establish_config_deck
establish_temp_segs
evaluate
evict_page
expand_assign
expand_by_name
expand_cobol_source
expand_infix
expand_initial
expand_pathname_
expand_prefix
expand_primitive
exponential_
expression_parse
expression_semantics
ext_parse
fast_connect_init
fast_hc_ipc
ffop
file
file_
file_util
filemap_checksum_
fill_refer
fill_vol_extents_
fim
fim_util
find_bit_
find_char_
find_command_
find_file_partition
find_partition
find_pathname_
find_rpv_subsystem
find_source_file_
flagbox_mgr
flush_ast_pool
fnp_init
fnp_multiplexer
fnp_throughput
fnp_util
force_write
format_cobol_source
format_list_parse
formline_
fort_math_ops_
fort_version_info
free_store
freecore
fs_alloc
fs_get
fs_modes
fs_move
fs_search
fsout_vol
function
fv_iv_template
generic_selector
get_addr_
get_addr_
get_array_size
get_aste
get_at_entry_
get_control_point_id_
get_defname_
get_defptr_
get_entry_name_
get_equal_name_
get_io_segs
get_kstep
get_main
get_mcm_meters
get_pathname_
get_ppr_
get_process_id_
get_process_usage
get_ptrs_
get_pvtx
get_size
get_temp_segments_
get_tpr_
getuid
gfms_substructure_records_

grab_aste
hardcore_sct_seg
hardware_fault
hash
hash_index_
hasp_meters_
hc_device_acct_
hc_dm_util
hc_dmpr_primitives
hc_exponent_control
hc_initlzr_auxl_init_
hc_ipc
hc_load_mpc
hc_page_trace
hc_tune
hfp_to_bfp_
history_reg_save
ibm3270_meters_
if_parse
illegal_procedure
imft_receive_object_
index_set
init_aste_pools
init_bce
init_branches
init_clocks
init_disk_pack_
init_dm_journal_seg
init_early_config
init_empty_root
init_hardcore_gates
init_hc_part
init_lvt
init_partitions
init_proc
init_pvt
init_root_dir
init_root_vols
init_scavenger_data
init_scu
init_sst
init_sst_name_seg
init_stack_0
init_str_seg
init_sys_var
init_syserr_log
init_toehold
init_vol_header_
init_volmap_seg
init_vtoc_man
initial_error_handler
initialize_faults
initialize_faults_data
initialize_int_static
initialize_kst
initializer
initiate_
initiate_search_rules
install_ttt_
integer_power_integer_
interpret_info_struc_
interpret_oncode_
io_chnl_util
io_config_init
io_data_list_semantics
io_error
io_log_status_info
io_manager
io_reconfig
io_semantics
io_statement_parse
ioa_
ioam_
iod_tables_compiler
iodc_
ioi_assignment
ioi_config
ioi_connect
ioi_device
ioi_get_status
ioi_init
ioi_masked
ioi_page_table
ioi_set
ioi_suspend_devices
ioi_usurp_channels
ioi_verify_lock
ioi_wire
ioi_workspace
iom_assign
iom_connect
iom_data_init
iom_error
iom_interrupt
iom_overhead
iom_reset
iom_switches
iom_switches
iom_unassign
ios_
ios_signal_
ios_write_around_
iox_
iox_attach_name
iox_close_file
iox_detach

iox_find_iocb
iox_get_options
iox_init_
iox_measure_options
iox_open_file
iox_propagate
iox_signal_
ipc_
ipc_data_
ipc_fast_
ipc_real_
ipc_util_
ipc_validate_
ips_
isot_fault_handler_
jump_op
kermit
kermit_get_filenames_
kermit_pad_
kermit_receive_
kermit_remote_requests_
kermit_send_
kermit_xfer_modes_
kst_info
kst_util
kstsrch
lap_simplex
level
level_0_
level_error
lex
lg_ctl_
limit_covert_channel
link_man
link_snap
link_trap_caller_
linkage_error_
list
list_inacl_all
list_init_
load_disk_mpcs
load_mst
load_system
lock
lock_mca
lock_volmap
log_create_
log_data_
log_initialize_
log_initiate_
log_limit_scan_
log_list_history_
log_move_
log_name_
log_position_
log_read_
log_salvage_
log_segment_
log_wakeup_
log_write_
logarithm_
logical_volume_manager
lookup
lot_fault_handler_
lv_request_
lv_request_communicator_
m_a
mail_table_mgr_
make_branches
make_msf_
make_sdw
make_seg
make_segs_paged
makeknown_
makestack
makeunknown_
map_free_count
map_onto_disk
mask_instruction
match_arguments
match_star_name_
math_constants_
math_routines_
math_routines_end_
mca_init_
mcs_timer
mcs_timer_daemon
mcs_trace
mcs_version
mdc_check_mdcs_
mdc_create_
mdc_init_
mdc_lock_
mdc_parse_acct_
mdc_repair_
mdc_set_
mdc_set_path_
mdc_status_
mdc_util_
mdcs_util_
mdx
memo
memo_list_
merge_attributes
merge_volume_log
message_table_

meter_ast_lock
meter_response_time
mlsys_mailbox_mgr_
mlsys_misc_utils_
mlsys_nit_interface_
mlsys_parse_control_args_
mlsys_storage_mgr_
mlsys_transmit_
mode_string_
mos_memory_check
mountedp
move_non_perm_wired_segs
move_r_or_t_
mowse_io_
mrds_dsl_delete
mrds_dsl_modify
mrl_
mseg_fs_interface_
msf_manager_
multiplexer_mgr_
mvt_
new_proj
nonlocal_goto_
numeric_to_ascii_
numeric_to_ascii_base_
object_info_
object_lib_
obs_reconfigure
oc_trans_input_
oc_trans_output_
ocd_
ocdcm_
offset_adder
on_line_salvager
on_parse
ondata_
op_mnemonic_
operator_process_cmds_
operator_semantics
optimizer
other_process_info
outward_call_handler
pa_search_list
page
page_error
page_fault
page_synch
parity_check
parse
parse_error
parse_tape_reel_name_
parse_tty_name_
partition_io
pathname_
pathname_am
pc
pc_abs
pc_check_tables_
pc_deposit
pc_recover_sst
pc_signal
pc_trace
pc_trace_pl1
pc_wired
picture_info_
pipe_
pl1_decat_char_
pl1_error_print
pl1_macro
pl1_macro_error_
pl1_macro_lex_
pl1_operators_
pl1_resignaller_
pl1_signal_
pl1_signal_conversion_
pl1_snap_
pl1_symbol_print
pl1_version
plio2_signal_
plus
pool_manager_
post_purge
power_
power_integer_
prds_init
pre_link_hc
prepare_symbol_table
principal_angle_
print
print_data
print_mail
priv_channel_manager
priv_delete_vtoce
priv_hasp_mpx
priv_ibm3270_mpx
priv_mcs_trace
priv_polled_vip_mpx
priv_x25_mpx
private_logical_volume
privileged_mode_ut
probe_get_expr_
proc_info
proc_int_handler
procedure_parse
process_cobol_report
process_entry

protocol_mpx
ptw_util_
purge_volume_log
put_aste
put_field_
put_format_
pxss
qedx
qedx_
quota
quota_util
quotaw
rcp_access_kernel_
rcp_access_kernel_setup
rcp_assign_device_
rcp_attach_
rcp_attach_lv_
rcp_audit
rcp_authenticate_device_
rcp_auto_register_
rcp_cancel_id_
rcp_cancel_resource_
rcp_check_assign_
rcp_check_attach_
rcp_check_attach_lv_
rcp_comment_
rcp_compute_aim_mode
rcp_compute_bracket_mode
rcp_compute_raw_mode
rcp_control_
rcp_copy_
rcp_detach_
rcp_detach_lv_
rcp_device_info_
rcp_disk_
rcp_find_
rcp_force_unassign_
rcp_get_scratch_volume_
rcp_init
rcp_ioi_attach_
rcp_lock_
rcp_lv_
rcp_match_
rcp_match_user_name_
rcp_merge_modes
rcp_message_
rcp_mount_timer_
rcp_mount_timer_poll_
rcp_operation_access
rcp_pointers_
rcp_pre_claim_
rcp_preload_
rcp_promote_
rcp_rcse_
rcp_reconfigure_
rcp_reserve_
rcp_resource_info_
rcp_ring1_init_
rcp_set_lock_
rcp_setup_event
rcp_tape_
rcp_tape_survey_
rcp_unassign_
rcp_unload_
rcp_validate_
rcprm_find_op
rcprm_find_resource_
rcprm_journal_file_
rcprm_journalize_
rcprm_list_resource_
rcprm_registry_mgr_
rcprm_registry_util_
rcprm_verify_registries_
rdisk_
rdm_apply_request_
rdm_file_requests_
rdm_forward_request_
rdm_mailbox_interface_
rdm_mbx_requests_
rdm_message_mark_mgr_
rdm_msg_requests_
rdm_reply_request_
read_cards_
read_disk
read_disk_label
real_initializer
rebuild_dir
rebuild_pvolog
receive_file_
reclassify
reconfig
reconfigure
record_io_
record_status
recover_volume_log
ref_name_
refer_extent
reference_parse
reload
reload_volume_
reloader
relocate_instruction_
requote_string_
reserve
resource_info_
retrieve_from_volume_

retv_copy
retv_util
retv_vol_control_
ring0_init
ring_0_peek
ring_alarm
ringbr_
rldr_arg_reader_
rldr_check_pvol_
rldr_finish_
rldr_input_
rldr_label_
rldr_output_
rldr_report_
rldr_volume_map_
rldr_vtoc_buffer_
rldr_vtoc_header_
rollback
rsw_util
rtcd_util_
salv_caller
salv_check_vtoce_
salv_dir_checker_
salv_directory
salv_dump_copy
salv_err_msg
salvage_pv
salvager
save_handler_mc
sc_abort_line_util_
sc_admin_command_
sc_admin_mode_
sc_command
sc_create_sci_
sc_edit_motd_
sc_exec_request_
sc_execute_command_line_
sc_get_error_name_
sc_init_
sc_ipc_mask_
sc_process_command_line_
sc_request_table_
sc_requests_
sc_shutdown_
sc_signal_handler_
sc_signal_io_handler_
sc_stat_
scan_token_table
scas_init
scavenge_volume
scavenger
scr_util
scs
scs_and_clock_init
sct_manager_
sdm_mbx_requests_
sdm_original_requests_
sdw_util_
search_ast
search_file_
seg_fault
segment_loader
segment_mover
segno_usage
semantic_translator
set
set_alarm_timer
set_disk_table_loc
set_ext_variable_
set_kst_attributes
set_lock_
set_privileges
set_procs_required
set_smic_pattern
set_sons_lvid
set_special_password
set_stack_ptr
set_sys_audit_thresholds_
set_system_time_zone_
set_volume_log
setfaults
share_expression
shutdown
shutdown_file_system
signal_
simplify_expression
simplify_offset
sine_
slt_manager
special_messages_
square_root_
stack_header_util_
stack_oob_handler
star_
start_cpu
state_man
statement_parse
statement_type
status
status_
stock
stock_man
stop_cobol_run
stop_cpu
stop_process
structure_library_2_

structure_library_4_
sty_mpx
sub_err_
subscripter
substr
sum
summarize_sys_log
switch_shutdown_file_system
syn_
sys_log_
sys_trouble
syserr
syserr_copy
syserr_copy_wired_log
syserr_log_daemon
syserr_real
syserr_seg_manager
system
system_comm_meters
system_control_
system_startup_
tangent_
tape_checksum_
tape_dim_data_
tape_error_data
tape_ioi_
tape_ioi_activate
tape_ioi_buffer_man
tape_ioi_error_retry
tape_ioi_hardware_status
tape_ioi_io
tape_ioi_modes
tape_ioi_utils
tape_ioi_wks_man
tape_mult_
tape_mult_close_
tape_mult_detach_
tape_mult_labeler_
tape_mult_modes_
tape_mult_open_
tape_mult_parse_
tape_mult_read_
tape_mult_util_
tape_mult_write_
tape_reader
tc
tc_init
tc_shutdown
tc_util
ted4
ted_
tedaddr_
template_address_space
template_area_header
template_slt_
terminate_
terminate_proc
thread
time_defaults_
time_info_
timer_manager_
toehold
trace
trace_mc
translator_info_
trap_caller_caller_
truncate
truncate_stacks
truncate_vtoce
try_to_unlock_lock
tty_analyze
tty_area_manager
tty_dump
tty_index
tty_interrupt
tty_lock
tty_modes
tty_overstrike_canon
tty_read
tty_replace_canon
tty_space_man
tty_tables_mgr
tty_util_
tty_write
uid_path_util
uncp
uncp_boot_interrupt
uncp_multiplexer
uncp_util
unique_bits_
unique_chars_
unpack_picture_
unwind_stack_
unwinder_
update_kste_access
update_vtoce
user_wire
v2pll
vacate_pv
validate
validate_cpu_card
validate_info_seg
value_list
verify_dump_volume
verify_label_
verify_lock

video_utils_
vip7760_meters_
vm_vio
volmap
volmap_page
volmap_util
volume_cross_check
volume_registration_cmds_
volume_registration_mgr_
vrm_lock_
vtoc_attributes
vtoc_interrupt
vtoc_man
vtoc_search
vtoce_stock_man
wdx
wioctl_
wire_proc
wire_stack
wired_fim
wired_hardcore_data
wired_plm
wired_shutdown
wired_utility_
wkp_signal_handler_
write_record
ws_channel_
ws_debug_
ws_packet_dispatcher_
ws_packet_receiver_
ws_packet_transmitter_
ws_timer_
ws_tools_
ws_tty_main_
ws_tty_mgr_
ws_tty_read_
ws_tty_write_
x25_analyze_
x25_dump_
x25_mpx
x25_mpx_data

APPENDIX B

FORMATTING DISKS WITH MTR

This appendix describes a procedure for formatting disk packs using the Media Test Routines (MTRs). The procedure utilizes an annotated script which shows typical input and output. Because the formatting procedures differ for MSU0451 and MSU0500/0501 devices, separate scripts are provided. In the scripts, input typed by the user is preceded by a ⊠ (bullet).

FORMATTING MSU0451 DISK PACKS

The following script shows how to run MTR tests 6 and 3 to format and test an MSU0451 disk pack, and to assign alternates to tracks found defective during testing.

1. Enter the Total OnLine Test System (TOLTS):

```
⊠ bound_tolts_$tolts_

  ***tolts executive version 810301 on 820812 at 20.071
```

2. Enter the MPC OnLine Test Subsystem (MOLTS):

```
  ***enter "polts", "molts", "colts", "isolts", "quit", or "msg"
  ???  ⊠ molts
```

3. List the disk configuration for the disk string which formatting will be done
(because of page constraints, the following message is not an exact copy of that which is displayed by the system):

```
??? ⊠ test pcd

peripheral configuration:

dska 451 16 units; starting with device no. 1
 020xx   primary channel of 4 logical channels on mpc mspa
 026xx   secondary channel of 4 logical channels on mpc mspa
 124xx   secondary channel of 4 logical channels on mpc mspb
 122xx   secondary channel of 4 logical channels on mpc mspb
```

4. Enter MTR test 6 to format and test the MSU0451 device:

```
??? ⊠ test mmt12020t6
```

where "test mmt12020t6" is a sample of the input format "test mmtICCDDtT":

mmt identifies the MTR test package

ICC gives the IOM number (0 = IOM A, 1 = IOM B, etc) and channel number (in decimal) of a channel by which the device to be formatted can be addressed. It must be one of those shown in the output of "test pcd" in step 3. In the sample input above, "120" is IOM B, channel 20.

DD gives the device number (in decimal) of the device to be tested. In the sample input, it is device 20 (dskb_20).

T gives the number of the MTR test to be run. In this case, test 6 should be run to format/test a pack.

5. The following output describes steps taken by MTR test 6 to attach the disk drive and mount the pack for writing:

```
***molts executive versions 820601 820701 on 820812 at 19.97
**0(mmt12020) short wait, allocation queued
**0(mmt12020) start tmt65a-rmc1, ttldat 820331, phy./log. id t//04
**0(mmt12020) start tmt65b-rmc2, ttldat 820331, phy./log. id t//04
**0(mmt12020) start tmt65c-rmc3, ttldat 820401, phy./log. id t//04
**0(mmt12020) start tmt65d-rmc4, ttldat 820405, phy./log. id t//04
**0(mmt12020) start tmt65e-rmc5, ttldat 820421, phy./log. id t//04
**0(mmt12020) start tmt65f-rmc6, ttldat 820331, phy./log. id t//04
**0(mmt12020)
rmc6 is at your service to format a disk pack -
**0(mmt12020)
***** write permission granted *****
**0(mmt12020)
***** begin format pack *****
the test will format all tracks on the pack.  format will
defined by device type.  bad tracks will be marked defective
(no alt.  assigned).
**0(mmt12020)
system device code = .ds450
```

6. Answer MTR initialization questions (not a restart, normal formatting, and use 3 write patterns during testing):

```
**0(mmt12020)
is this a restart?  enter (y or n) - ⊠ n
**0(mmt12020)
select (f)ast or (n)ormal format?  (f)ast format is designed
for data security erase and/or test purposes.  (n)ormal format
is designed for disk packs that are going to be used in systems
applications.  enter (f or n) - ⊠ n
**0(mmt12020)
select from "1" to "7" write patterns?  enter (1 thru 7) - ⊠ 3
```

7. At this point, formatting of the pack begins:

```
**0(mmt12020)
***** begin disk pack format *****
```

8. After the message in Step 7 is displayed, press the BREAK key to interrupt formatting operations. When MOLTS prompts for input, set test options to: report the current cylinder/head (CCC/HH) address; display CCC/HH for transient errors; report test progress every 100 cylinders, with summary reports attached.

```
⊠ <PRESS BREAK KEY>
???  ⊠ test momt12020.r
```

where "test momt12020.r" is a sample of the input format "test momtICCDD.O":

momt	identifies request to set options
ICCDD	are the IOM, Channel and Device numbers given in Step 4.
.r	is the first option, to report current CCC/HH location.

Set the remaining options when prompted:

```
*0(mmt12020) t6 enter options:  ⊠ .i
*0(mmt12020) t6 enter options:  ⊠ .e
*0(mmt12020) t6 enter options:  ⊠ .s
*0(mmt12020) t6 enter options:  ⊠ .t
*0(mmt12020) t6 enter options:  ⊠ .go
```

9. When the .go option is entered in Step 8, MTR reports the current location being formatted and displays the defective tracks found. It then asks if you want to continue formatting:

```
**0(mmt12020)
format function current addr. = 007/00
**0(mmt12020)
format function current addr. = 007/00
**0(mmt12020)
***** rmc6 - summary report *****
no tracks were formatted defective
**0(mmt12020)
do you want the test to continue?  enter (y or n) - ⊠ y
```

10. After every 100 cylinders are formatted, MTR displays defective tracks found. For example, the final summary displayed just before formatting completes, looks like:

```
**0(mmt12020)
rmc6 has formatted tracks "000/00 thru 700/00"
**0(mmt12020)
***** rmc6 - summary report *****
no tracks were formatted defective
**0(mmt12020)
rmc6 has formatted tracks "000/00 thru 800/00"
**0(mmt12020)
***** rmc6 - summary report *****
no tracks were formatted defective
**0(mmt12020)
***** disk pack format complete *****
```

11. After formatting is complete, MTR begins testing the tracks on the formatted pack. Defective tracks are usually encountered only during the testing phase. Error summaries are displayed after every 100 cylinders have been tested.

```
start media test phase
**0(mmt12020)
rmc6 has tested tracks "000/00 thru 100/00"
**0(mmt12020)
***** rmc6 - summary report *****
no tracks were formatted defective
**0(mmt12020)
rmc6 has tested tracks "000/00 thru 200/00"
**0(mmt12020)
***** rmc6 - summary report *****
no tracks were formatted defective
**0(mmt12020)
rmc6 has tested tracks "000/00 thru 300/00"
**0(mmt12020)
***** rmc6 - summary report *****
defective - marginal data field on std track
217/10
**0(mmt12020)
***** rmc6 - summary report *****
defective - unrec. data field on std track
244/06,245/06
**0(mmt12020)
***** rmc6 - summary report *****
reclaimed - reformatted and certified
246/06
```

12. When testing is complete, termination summary reports are displayed:

```
**0(mmt12020)
***** normal termination summary reports *****
**0(mmt12020)
***** rmc6 - summary report *****
defective - marginal data field on std track
217/10
**0(mmt12020)
***** rmc6 - summary report *****
defective - unrec. data field on std track
244/06,245/06
**0(mmt12020)
***** rmc6 - summary report *****
reclaimed - reformatted and certified
246/06
```

13. MTR then asks if you want to select a new test (answer "y" for yes):

```
**0(mmt12020)
want to select a new test?
enter (y or n) - ◙ y
```

14. MTR then displays information describing how to select the next test:

```
**0(mmt12020)
rmc6 will go into waiting!
select test (t1 thru t6)
enter test no. thru standard option call (test momticcddtx) -
**0(mmt12020)
waiting
```

15. To actually select the next test, press the BREAK key and wait for the MOLTS prompt. Then select test 3, which assigns alternate tracks for those tracks found to be defective above.

```
⊠ <PRESS BREAK KEY>
???  ⊠ test momt12020t3
```

where "test momt12020t3" is a sample of the input format "test momtICCDDtT":

momt — identifies request to set options

ICCDD — are the IOM, Channel and Device numbers given in Step 4.

tT — gives the number of the next test to run.

Test 3 initialization displays the following information:

```
**0(mmt12020) start tmt65e-rmc5, ttldat 820421, phy./log. id t//04
**0(mmt12020) start tmt65d-rmc4, ttldat 820405, phy./log. id t//04
**0(mmt12020) start tmt65c-rmc3, ttldat 820401, phy./log. id t//04
```

16. Select subtest 4 of test 3, to assign alternates to all defective tracks:

```
**0(mmt12020)
rmc3 is at your service
for track and cylinder reformat -
select a sub test
a) subtst 1 - reformat 1 track (good)
b) subtst 2 - reformat 1 cylinder (good)
c) subtst 3 - reformat 1 track (defective)
d) subtst 4 - assign alternate tracks
enter (1 thru 4) - ⊠ 4
```

17. MTR then briefly describes the subtest, and asks if you want to continue (answer "y" for yes)

```
**0(mmt12020)
***** begin subtst 4 *****
assign alternate tracks on the device
a) subtst will search thru all standard tracks looking for
   tracks marked defective (no alternate assigned).
b) when a track marked defective (no alternate assigned) is
   detected, the subtst will stop and process this track.
c) the alternate track processor will go out to the alternate
   track cylinders and find the first available alternate.  it
   will mark the track as assigned alternate.  then it will
   mark the standard track as defective (alt.  assigned).
d) the search process will terminate after the last standard
   track completes testing and/or processing.
do you want the subtst to continue?  enter (y or n) - ◙ y
```

18. MTR then asks for permission to overwrite the pack's label (answer "y" for yes):

```
**0(mmt12020)
***** rmc3 - label obliterate warning *****
all sub tests in rmc3 will overwrite the system
label on track zero.
do you want the sub test to continue?  enter (y or n) - ◙ y
```

19. MTR then asks if you are restarting (answer "n" for no):

```
**0(mmt12020)
system device code = .ds450
**0(mmt12020)
is this a restart?  enter (y or n) -  ◙ n
```

20. MTR then begins displaying summary reports after every 100 cylinders are checked for alternate assignments:

```
**0(mmt12020)
rmc3 has tested tracks "000/00 thru 200/00"
**0(mmt12020)
***** rmc3 - subtst 4 summary report *****
no alternate tracks were assigned
**0(mmt12020)
rmc3 has tested tracks "000/00 thru 300/00"
**0(mmt12020)
***** rmc3 - subtst 4 summary report *****
defective - alt assigned
 def          alt          def          alt
cyl/hd       cyl/hd       cyl/hd       cyl/hd
217/10       811/00       244/06       811/01
245/06       811/02
```

21. After alternate assignments are complete, MTR displays a summary report describing all alternates on the pack:

```
**0(mmt12020)
***** normal termination summary reports *****
**0(mmt12020)
***** rmc3 - subtst 4 summary report *****
defective - alt assigned
 def          alt          def          alt
cyl/hd       cyl/hd       cyl/hd       cyl/hd
217/10       811/00       244/06       811/01
245/06       811/02
```

22. MTR then asks if you want to select a new test (answer "n" for no, and "quit" to exit from TOLTS).

```
**0(mmt12020)
want to select a new test?
enter (y or n) - ⊠ n
**0(mmt12020) normal term 1
***molts executive version 820701 off 820812 at 21.45 p.t. 119530

***enter "polts", "molts", "colts", "isolts", "quit", or "msg"
???  ⊠ quit

***tolts executive version 810301 off 820812 at 21.375
r 21:37 1107.348 1162
```

FORMATTING MSU0500/MSU0501 DISK PACKS

The following script shows how to run MTR tests 6 and 7 to format and test an MSU0500 or MSU0501 disk drive, and to assign alternates to tracks found defective during testing.

The MPC normally treats an MSU0500 or MSU0501 disk drive as two separately addressable devices. However, MTR formats and tests both logical devices during a single invocation, referring to one as the "odd device" (e.g., dskc_27) and the second as the "even device" (e.g., dskc_28). MTR refers to both logical devices as a single "head assembly" or "hda".

1. Enter the Total OnLine Test System (TOLTS):

```
⊠ bound_tolts_$tolts_

  ***tolts executive version 810301 on 820812 at 20.071
```

2. Enter the MPC OnLine Test Subsystem (MOLTS):

```
  ***enter "polts", "molts", "colts", "isolts", "quit", or "msg"
  ???  ⊠ molts
```

3. List the disk configuration:

```
  ???  ⊠ test pcd

  peripheral configuration:

  dskc 501 32 units; starting with device no. 1
   028xx   primary channel of 4 logical channels on mpc mspc
   030xx   secondary channel of 4 logical channels on mpc mspc
   130xx   secondary channel of 4 logical channels on mpc mspd
   128xx   primary channel of 4 logical channels on mpc mspd
```

4. Enter MTR test 6 to format and test the entire MSU0500 or MSU0501 device:

```
??? ⊠ test mmt12827t6
```

where "test mmt12827t6" is a sample of the input format "test mmtICCDDtT":

- mmt — identifies the MTR test package
- ICC — gives the IOM number (0 = IOM A, 1 = IOM B, etc) and channel number (in decimal) of achannel by which the device to be formatted can be addressed.It must be one of those shown in the output of "test pcd" in step 3. In the sample input above, "128" is IOM B, channel 28.
- DD — gives the device number (in decimal) of the device to be tested. In the sample input, it is device 27 (dskc_27). Always give the device number of the "odd device" associated with the disk drive.
- T — gives the number of the MTR test to be run. In this case, test 6 should be run to format/test the drive.

5. The following output describes steps taken by MTR test 6 to attach the disk drive for writing:

```
***molts executive versions 820601 820701 on 820805 at 20.08
**0(mmt12827) short wait, allocation queued
**0(mmt12827) short wait, allocation queued
**0(mmt12827) start tmt67a-mtr1, ttldat 820401, phy./log. id t//04
**0(mmt12827) start tmt67b-mtr2, ttldat 820401, phy./log. id t//04
**0(mmt12827) start tmt67c-mtr3, ttldat 820402, phy./log. id t//04
**0(mmt12827) start tmt67d-mtr4, ttldat 820405, phy./log. id t//04
**0(mmt12827) start tmt67e-mtr5, ttldat 820421, phy./log. id t//04
**0(mmt12827) start tmt67f-mtr6, ttldat 820405, phy./log. id t//04
**0(mmt12827)
mtr6 is at your service to format a physical device -
**0(mmt12827)
***** write permission granted *****
**0(mmt12827)
***** begin upgrade/downgrade hda *****
the test will format all tracks on the hda.  the format will be
defined by device type.  bad tracks will be marked defective
(no alternate).
```

6. Answer MTR initialization questions (ok to format with 512 words per sector, not a restart, normal formatting, and use 3 write patterns during testing):

```
**0(mmt12827)
device pair are configured as msu0501's
the hda will be formatted in (512) words/sector.
is this correct?
enter (y or n) -  ⊠ y
**0(mmt12827)
is this a restart?
enter (y or n) -  ⊠ n
**0(mmt12827)
select (f)ast or (n)ormal format?
(f)ast format is designed for data security
erase and/or test purposes.
(n)ormal format is designed for hda's to be used
in systems applications.
enter (f or n) -  ⊠ n
**0(mmt12827)
select from "1" to "7" write patterns?
enter (1 thru 7) -  ⊠ 3
```

7. At this point, formatting of the pack begins:

```
**0(mmt12827)
***** begin hda format *****
```

8. After the message in Step 7 is displayed, press the BREAK key to interrupt formatting operations. When MOLTS prompts for input, set test options to: report the current cylinder/head (CCC/HH) address; display CCC/HH for transient errors; report test progress every 100 cylinders, with summary reports attached.

```
⊠ <PRESS BREAK KEY>
???  ⊠ test momt12827.t
```

where "test momt12020.r" is a sample of the input format "test momtICCDD.O":

momt — identifies request to set options

ICCDD — are the IOM, Channel and Device numbers given in Step 4.

.r — is the first option, to report current CCC/HH location.

Set the remaining options when prompted:

```
*0(mmt12827) t6 enter options:  ⊠ .e
*0(mmt12827) t6 enter options:  ⊠ .s
*0(mmt12827) t6 enter options:  ⊠ .i
*0(mmt12827) t6 enter options:  ⊠ .r
*0(mmt12827) t6 enter options:  ⊠ .go
```

9. When the .go option is entered in Step 8, MTR reports the current location being formatted and displays the defective tracks found. It then asks if you want to continue formatting:

```
**0(mmt12827)
format function current addr. = 004/00
**0(mmt12827)
format function current addr. = 004/00
**0(mmt12827)
***** statistics from format of hda *****
summary for msu0501 devices (27/28)
no. of tracks with 1 defect skip  = 1
no. of tracks with 2 defect skips = 0
no. of tracks with 3 defect skips = 0
no. of new defect skips generated = 0
     total defect skips processed = 1
 odd device defective tracks      = 0
even device defective tracks      = 0
physical device defective tracks  = 0
                            total = 0
**0(mmt12827)
***** mtr6 - hda condition summary report *****
no tracks were marked defective.
**0(mmt12827)
do you want the test to continue?
enter (y or n) -  ◙ y
```

10. After every 100 cylinders are formatted, MTR displays defective tracks found. For example, the final summary displayed just before formatting completes, looks like:

```
**0(mmt12827)
mtr6 has formatted tracks "000/00 thru 800/00"
**0(mmt12827)
***** mtr6 - hda condition summary report *****
 --- msu0501 odd device report (27) ---
defective - error logging track info
006/19,028/19,284/01,370/12
**0(mmt12827)
***** mtr6 - hda condition summary report *****
 --- msu0501 even device report (28) ---
defective - error logging track info
008/19,026/19,096/05,174/16,381/06,736/09,778/19
**0(mmt12827)
***** statistics from format of hda *****
summary for msu0501 devices (27/28)
no. of tracks with 1 defect skip  = 33
no. of tracks with 2 defect skips = 2
no. of tracks with 3 defect skips = 0
no. of new defect skips generated = 0
     total defect skips processed = 37
 odd device defective tracks      = 4
even device defective tracks      = 7
physical device defective tracks  = 0
                            total = 11
**0(mmt12827)
***** hda format complete *****
```

11. After formatting is complete, MTR begins testing the tracks on the formatted pack. Error summaries are displayed after every 100 cylinders have been tested.

```
start media test phase
**0(mmt12827)
mtr6 has tested tracks "000/00 thru 100/00"
**0(mmt12827)
***** mtr6 - hda condition summary report *****
 --- msu0501 odd device report (27) ---
defective - error logging track info
006/19,028/19,284/01,370/12,816/04,818/04,832/04
**0(mmt12827)
***** mtr6 - hda condition summary report *****
 --- msu0501 even device report (28) ---
defective - error logging track info
008/19,026/19,096/05,174/16,381/06,736/09,778/19
**0(mmt12827)
***** mtr6 - hda condition summary report *****
 --- msu0501 even device report (28) ---
reclaimed - repaired data field
042/15
**0(mmt12827)
***** statistics from format of hda *****
summary for msu0501 devices (27/28)
no. of tracks with 1 defect skip  = 42
no. of tracks with 2 defect skips = 2
no. of tracks with 3 defect skips = 0
no. of new defect skips generated = 1
    total defect skips processed  = 47
 odd device defective tracks      = 7
even device defective tracks      = 7
physical device defective tracks  = 0
                            total = 14
```

12. When testing is complete, termination summary reports are displayed:

```
**0(mmt12827)
***** normal termination summary reports *****
**0(mmt12827)
***** mtr6 - hda condition summary report *****
 --- msu0501 odd device report (27) ---
defective - error logging track info
006/19,028/19,284/01,370/12,816/04,818/04,832/04
**0(mmt12827)
***** mtr6 - hda condition summary report *****
 --- msu0501 odd device report (27) ---
reclaimed - repaired data field
764/09,830/04
**0(mmt12827)
***** mtr6 - hda condition summary report *****
 --- msu0501 even device report (28) ---
defective - error logging track info
008/19,026/19,096/05,174/16,381/06,736/09,778/19
**0(mmt12827)
***** mtr6 - hda condition summary report *****
 --- msu0501 even device report (28) ---
reclaimed - repaired data field
042/15,762/09,818/08
**0(mmt12827)
***** statistics from format of hda *****
summary for msu0501 devices (27/28)
no. of tracks with 1 defect skip  = 42
no. of tracks with 2 defect skips = 2
no. of tracks with 3 defect skips = 0
no. of new defect skips generated = 5
     total defect skips processed = 51
 odd device defective tracks      = 7
even device defective tracks      = 7
physical device defective tracks  = 0
                            total = 14
```

13. MTR then asks if you want to select a new test (answer "y" for yes)

```
**0(mmt12827)
want to select a new test?
enter (y or n) - ⊠ y
```

14. MTR then displays information describing how to select the next test:

```
**0(mmt12827)
mtr6 will go into waiting!
select test (t1 thru t7)
enter test no. thru standard option call (test momticcddtx) -
**0(mmt12827)
waiting
```

15. To actually select the next test, press the BREAK key and wait for the MOLTS prompt. Then select test 7, which assigns alternate tracks for those tracks found to be defective above.

    Test 7 assigns alternates for the complete head assembly, whereas test 3 (used in the procedure for formatting MSU0451 disks) only assigns alternates for a single logical device. Thus, test 3 would have to be run twice (once for the odd device and once for the even device) to assign alternates on an MSU0500 or MSU0501 disk.

```
⊠ <PRESS BREAK KEY>
???  ⊠ test momt12827t7
```

    where "test momt12827t7" is a sample of the input format "test momtICCDDtT":

    momt — identifies request to set options

    ICCDD — are the IOM, Channel and Device numbers given in Step 4.

    tT — gives the number of the next test to run.

    Test 7 initialization displays the following information:

```
**0(mmt12827) start tmt67g-mtr7, ttldat 820405, phy./log. id t//04
```

16. Select subtest 1 of test 7, to assign alternates to all defective tracks:

```
**0(mmt12827)
mtr7 is at your service
for special physical device formatting -
select the subtst
a) subtst 1 - assign all alternate tracks
b) subtst 2 - create & write logging tracks
enter (1 thru 2) -  ⊠ 1
```

17. MTR then briefly describes the subtest, and asks if you want to continue (answer "y" for yes):

```
**0(mmt12827)
***** begin subtst 1 *****
assign alternate tracks on the physical device the subtst will
search "all" standard tracks on the hda for defective (no alt.
assigned).  if any are found, it will assign the 1st available
alternate to them.
**0(mmt12827)
do you want subtst (1) to continue?
enter (y or n) -  ⊠ y
```

18. MTR then asks if you are restarting (answer "n" for no):

```
**0(mmt12827)
is this a restart?
enter (y or n) -  ⊠ n
```

19. MTR then begins displaying summary reports after every 100 cylinders are checked for alternate assignments:

```
**0(mmt12827)
mtr7 has processed tracks "000/00 thru 100/00"
**0(mmt12827)
***** mtr7 - subtst 1 summary report *****
 --- msu0501 odd device report (27) ---
defective - alternate track assigned
 def          alt          def          alt
cyl/hd       cyl/hd       cyl/hd       cyl/hd
006/19       840/00       028/19       840/03
**0(mmt12827)
***** mtr7 - subtst 1 summary report *****
 --- msu0501 even device report (28) ---
defective - alternate track assigned
 def          alt          def          alt
cyl/hd       cyl/hd       cyl/hd       cyl/hd
008/19       840/01       026/19       840/02
096/05       840/04
```

20. After alternate assignments are complete, MTR displays a summary report describing all alternates on the pack:

```
**0(mmt12827)
***** normal termination summary reports *****
**0(mmt12827)
***** mtr7 - subtst 1 summary report *****
 --- msu0501 odd device report (27) ---
defective - alternate track assigned
 def          alt          def          alt
cyl/hd       cyl/hd       cyl/hd       cyl/hd
006/19       840/00       028/19       840/03
284/01       840/06       370/12       840/07
816/04       840/11       818/04       840/12
832/04       840/13
**0(mmt12827)
***** mtr7 - subtst 1 summary report *****
 --- msu0501 even device report (28) ---
defective - alternate track assigned
 def          alt          def          alt
cyl/hd       cyl/hd       cyl/hd       cyl/hd
008/19       840/01       026/19       840/02
096/05       840/04       174/16       840/05
381/06       840/08       736/09       840/09
778/19       840/10
```

21. MTR then asks if you want to select a new test (answer "n" for no, and "quit" to exit TOLTS).

```
**0(mmt12827)
want to select a new test?
enter (y or n) - ⊠ n
**0(mmt12827) normal term 1
***molts executive version 820701 off 820806 at 00.27 p.t. 5185916

***enter "polts", "molts", "colts", "isolts", "quit", or "msg"
???  ⊠ quit

***tolts executive version 810301 off 820806 at 00.165
r 00:16 5188.584 1038
```

Bull HN Information Systems Inc.
Corporate Headquarters: 300 Concord Rd., Billerica, MA 01821
U.S.A.: 200 Smith Street, MS 486, Waltham, MA 02154
Mexico: Hamburgo No. 64, Col. Juarez Delegacion Cuauhtemoc, 06600 Mexico, D.F.
U.K.: Great West Rd., Brentford, Middlesex TW8 9DH, England **Italy**: 32 Via Pirelli, 20124 Milan
Canada: 155 Gordon Baker Rd., North York, Ontario M2H 3P9 **New Zealand**: 14/16 Liverpool Street, Auckland 1
Asia: 4/F, Shui on Centre, 6-8 Harbour Rd., Wanchai, Hong Kong **Australia**: 124 Walker Street, North Sydney, N.S.W. 2060

SIB12.3

52066, 289, Printed in U.S.A.