MicroVMS
Release Notes
Version 4.1

January 1985

The MicroVMS Release Notes Version 4.1 contain installation information, problems resolved by Version 4.1, current problems and restrictions, and notes to published documentation.

Operating System and Version:	MicroVMS Version 4.1
Software Version:	MicroVMS Version 4.1

This document was prepared using an in-house documentation production system. All page composition and make-up was performed by TEX, the typesetting system developed by Donald E. Knuth at Stanford University.

Contents

Chapter 3   Current Problems and Restrictions

Chapter 4   Notes To Published Documentation

Appendix A   MicroVMS Version 4.1 Mandatory Update Listing

Appendix B   OTS$MOVE3 and OTS$MOVE5 Routine Descriptions

Appendix C   Additional MicroVAX I Bootstrap Support

Preface

Before installing your MicroVMS kit, read this document. The *MicroVMS Release Notes Version 4.1* contain the following sections:

- How to Install the Kit

- Problems Resolved by Version 4.1

- Current Problems and Restrictions

- Notes to Published Documentation

- MicroVMS Version 4.1 Mandatory Update Listing

- OTS$MOVE3 and OTS$MOVE5 Routine Descriptions

- Additional MicroVAX I Bootstrap Support

Conventions

This manual uses the following conventions in displaying the syntax requirements of user input to the system and in displaying examples:

- RETURN key—The RETURN key is not always shown in formats and examples. Assume that you must press RETURN after typing a command or other input to the system unless instructed otherwise.

- CTRL key—The word CTRL followed by a slash followed by a letter means that you must type the letter while holding down the CTRL key. For example, CTRL/B means hold down the CTRL key and type the letter B.

- Lists—When a format item is followed by a comma and an ellipsis (,...), you can enter a single item or a number of items separated by commas. When a format item is followed by a plus sign and an ellipsis (+...), you can enter a single item or a number of those items connected by plus signs. If you enter more than one item, you must enclose the list in parentheses. A single item need not be enclosed in parentheses.

- Optional items—An item enclosed in square brackets ([ ]) is optional.

- Angle Brackets—In examples, angle brackets enclose a syntactic element of user input, such as a key (< D>), a key sequence (< CTRL/Z>), or a parameter (< password>).

- Examples—Examples show both system output (prompts, messages, and displays) and user input. User input is printed in red.

Chapter 1

How to Install the Version 4.1 Update Kit

Version 4.1 is distributed as a maintenance update kit consisting of documentation, and patch and replacement files. These files are contained on RX50 floppy diskettes.

When you install the update kit, a Version 4.1 system is produced, provided that you install the kit on a Version 4.0 system and you have applied no other patches to that system. If you were a field test customer for Version 4.0, your system must have the official Version 4.0 update applied to it. The Version 4.1 update will not work correctly if applied to the field test Version 4.0.

1.1   The Version 4.1 Kit

The components of the MicroVMS Version 4.1 update kit are:

- Software Product Description (Order No. AE-DD57E-TE) and Cross-Reference Table (Order No. AE-DG84F-TE)

- *MicroVMS Release Notes* Version 4.1 (this document, Order No. AA-EF82A-TE)

- Update media: three RX50 floppy diskettes

1.2   Optional Software Products

The Version 4.1 kit does not contain updates to any MicroVAX optional software products. For more information about optional software products, refer to the MicroVAX Optional Software Cross-Reference Table in the Version 4.1 Software Product Description. Documentation for a specific optional software product is shipped with that product.

1.3 Before Installing the Kit

This section outlines the steps you should take before you install your Version 4.1 update kit.

1. Remove all unwanted or redundant files from the current system disk before you update your system.

   To create space on the system disk, DIGITAL recommends that you do not log in under the SYSTEM account. The SYSTEM account, which has all privileges (including BYPASS), is intended only for software installation, bootstrapping, and system problem diagnosis. You can avoid problems by creating another account and assigning it the minimum privileges you require.

   During installation, a significant number of system disk blocks are used for the work files needed to perform the installation and the files that are read from the distribution volume. The number of blocks used during the installation is called the "peak disk block utilization" and is listed in a table later in this section. DIGITAL recommends that you make enough blocks available on the system disk to cover peak utilization.

   NOTE: The installation will fail unless there are at least 7400 free blocks.

   If you choose not to change the number of disk blocks available, then VMSINSTAL will operate in an alternate mode that reduces the peak utilization.

   NOTE: If a system failure occurs while VMSINSTAL is operating in the alternate mode, the system disk is useless and you will need to restart the kit installation from the beginning.

   When the installation is complete, the system disk is returned to its normal state. However, many disk blocks will be permanently used, over and above the number in use before the installation was begun. This number is called the "net disk block utilization." The net utilization will vary, depending on whether you choose to purge old copies of system files that are replaced during the installation (see Section 1.5, step 1).

   After you reboot the system, you can recover additional disk blocks by purging system files that VMSINSTAL cannot purge during the installation (see Section 1.5, step 2).

   The next table summarizes the approximate disk block utilization requirements for Version 4.1.

Peak disk block utilization	8100
Net disk block utilization if files are purged during installation	3000
Net disk block utilization if files are not purged during installation	7400

You can confirm the free block count with the following DCL command:

```
$ SHOW DEVICE system-device-name
```

2. Back up your system disk as described in Chapter 1 of the *MicroVMS User's Manual*.

   A system failure at a critical point in the installation could result in unusable files; in addition, optional software products might delete older product versions before they install new ones. Therefore, you should always back up your system disk before you attempt any software installation.

3. Log in at the console terminal under the system manager's account, SYSTEM.

   If you are installing from an operator's terminal, you will receive a message after each mount operation if the SYSGEN parameter MOUNTMSG has been set, and after each dismount operation if the SYSGEN parameter DISMOUMSG has been set. Each message will appear within 30 seconds after its associated operation.

4. Ensure that all users are logged out and that all batch jobs are completed.

5. Prevent users from gaining access to the system by issuing the following command:

   ```
   $ SET LOGINS/INTERACTIVE=0
   ```

6. DECnet–VAX must be shut down. If you are not sure that your system includes DECnet–VAX, enter the following command:

   ```
   $ SHOW NETWORK
   ```

   If the message "Network Unavailable" appears, skip to step 7. If your system includes DECnet–VAX, shut it down by issuing the following command:

   ```
   $ RUN SYS$SYSTEM:NCP
   ```

   The system responds with the NCP prompt (NCP>). At this prompt, enter the following command:

   ```
   NCP> SET EXECUTOR STATE SHUT
   NCP> EXIT
   ```

   DECnet–VAX will perform an orderly shutdown, and OPCOM will notify you when shutdown is complete.

7. Ensure that the limits in the SYSTEM account authorization record have values that are equal to, or greater than, the following default values:

Buffered byte quota count (BYTLM)	20480
Enqueue quota (ENQLM)	30
Direct I/O limit (DIOLM)	18
Buffered I/O limit (BIOLM)	18
Open file quota (FILLM)	30
AST limit (ASTLM)	24

To check these values, run the Authorize Utility (AUTHORIZE) by entering the following commands at the dollar sign prompt ($).

```
$ SET DEFAULT SYS$SYSTEM
$ RUN AUTHORIZE
```

The system responds with the UAF prompt (UAF>). At this prompt enter the following command:

```
UAF> SHOW SYSTEM
```

The system will display the current values of the SYSTEM account's authorization record.

If necessary, you can change these values by entering the following command:

```
UAF> MODIFY SYSTEM/limit=new_value
```

For example:

```
UAF> MODIFY SYSTEM/DIOLM=18
```

To return to DCL command level issue the EXIT command.

```
UAF> EXIT
```

NOTE: You must log out and log in again for the new values to take effect.

8. Ensure that your system has at least 10 unused global sections and at least 200 unused global pages. To determine how many global sections and global pages are available, enter the DCL command

```
$ INSTALL :== INSTALL/COMM
$ INSTALL
INSTALL> LIST/GLOBAL
INSTALL> EXIT
```

Read the last line of the display and note

- The number of global sections **USED**

- The number of global pages **USED**

- The number of global pages **UNUSED**

To determine the unused global section count, invoke the System Generation Utility (SYSGEN) as follows:

```
$ RUN SYS$SYSTEM:SYSGEN
SYSGEN> USE CURRENT
SYSGEN> SHOW GBLSECTIONS
```

The number in the first column of the display is the current number of global sections. Subtract the number of used global sections (shown by the INSTALL display) from the current number of global sections (shown by the SYSGEN display). The difference between the number of used global sections and the current number of global sections is the unused global section count.

If the unused global section count is less than 10 you must increase the GBLSECTIONS parameter. To compute the new value, add 10 to the number of used global sections (shown by the INSTALL display); then use this command:

```
SYSGEN> SET GBLSECTIONS new-value
```

If the number of unused global pages (which was shown by INSTALL) is less than 200, you must increase the GBLPAGES parameter with the command

```
SYSGEN> SET GBLPAGES new-value
```

To compute the new value, add 200 to the number of used global pages (shown by the INSTALL display) if you have modified either the GBLPAGES or the GBLSECT parameter.

If you have modified either parameter (GBLPAGES or GBLSECTIONS), enter the following commands to exit SYSGEN.

```
SYSGEN>WRITE CURRENT
SYSGEN>EXIT
```

NOTE: Neither GBLPAGES nor GBLSECTIONS is a dynamic parameter; therefore, you must reboot your system in order for the new value to take effect.

1.4 Installing the Kit

After you have completed the steps in Section 1.3, perform the next steps to install the kit. At any time during the execution of VMSINSTAL, you can enter a question mark (?) for help.

Install the Common Utilities Option to Version 4.0 before proceeding. This is necessary to invoke Stand-Alone Backup on system disk support which will be updated when the kit is installed. This is also necessary to perform step 3 (recreating Stand-Alone Backup on RX50s).

During the entire process look for error and warning messages that indicate tasks you must perform manually. Many informational messages will be displayed; these messages can usually be ignored.

1. Enter the following DCL command:

   ```
   $ @SYS$UPDATE:VMSINSTAL VMS041 $FLOPPY1:
   ```

   At the start of the VMSINSTAL procedure, you need to answer the following question:

   ```
   * Are you satisfied with the backup of your system disk
   [YES]?
   ```

   If you have performed the backup of your system disk as recommended in step 2 of Section 1.3, press RETURN.

   Next, VMSINSTAL prompts you as follows:

   ```
    Please mount the first volume of the set on $FLOPPY1:.
   ```

   ```
   * Are you ready?
   ```

   Respond YES when the first volume is in the diskette drive.

   You will be prompted to mount each additional volume. After you insert each volume, enter the response YES. When all the volumes necessary for step 1 have been copied, leave the current volume in the drive while the installation continues.

2. After the first save set has been copied from the installation volumes, the following prompt appears as part of the VMSINSTAL procedure.

   ```
   1)  Apply all fixes to the system
   2)  Create a file with the descriptions of all fixes
   3)  Both of the above
   ```

   ```
   * What would you like to do [3]:
   ```

   Fixes are updates consisting either of patches to existing code or module replacements. Command procedures in the update distribution kit control the application of the updates. These command procedures contain comments briefly describing each update.

   Option 1 causes VMSINSTAL to perform the update without creating the description file. If you choose option 2, VMSINSTAL does not update your system; it simply creates the update description file.

   If you choose option 1 or 3, VMSINSTAL prompts you with the following question:

   ```
   * Do you want to purge files replaced by this installation [YES]?
   ```

3. When VMSINSTALL receives your reply to the prompt, the copy operation from the console drive will continue.

4. When the update is complete, the command procedure performs an orderly shutdown of your system. You must reboot your system manually, as described in Chapter 1 of the *MicroVMS User's Manual*.

1.5   After Installing the Update Kit

After the update kit is installed, you should perform the following tasks:

1. Several files cannot be purged by the Version 4.1 update procedure, but you can recover approximately 1700 disk blocks by using the PURGE command to remove old versions of the following files after you reboot your system (files that are parts of MicroVMS options are so designated).

- SYS$LIBRARY:CONVSHR.EXE

- SYS$LIBRARY:FDLSHR.EXE

- SYS$LIBRARY:SECURESHR.EXE

- SYS$UPDATE:VMSINSTAL.COM

- SYS$SYSTEM:DCL.EXE

- SYS$SYSTEM:DUDRIVER.EXE

- SYS$SYSTEM:F11BXQP.EXE

- SYS$SYSTEM:LTDRIVER.EXE (Common Utilities Option)

- SYS$SYSTEM:JOBCTL.EXE

- SYS$LIBRARY:LBRSHR.EXE

- SYS$SYSTEM:MAIL.EXE (Common Utilities Option)

- SYS$SYSTEM:MTAAACP.EXE (System Programming Option)

- SYS$SYSTEM:NETACP.EXE (DECnet Option)

- SYS$SYSTEM:NODRIVER.EXE (DECnet Option and Common Utilities Option)

- SYS$SYSTEM:OPCOM.EXE

- SYS$SYSTEM:PUDRIVER.EXE

- SYS$SYSTEM:RMS.EXE

- SYS$SYSTEM:SYS.EXE

- SYS$SYSTEM:TTDRIVER.EXE

- SYS$SYSTEM:XGDRIVER.EXE
- SYS$MANAGE:SYLOGIN.COM
- SYS$MANAGE:SYSTARTUP.COM
- SYS$MANAGE:WELCOME.TXT

2. Regenerate Stand-Alone Backup after installing Version 4.1. This version contains several corrections to images that are part of the Stand-Alone Backup procedure. To include these corrections in your system, you must regenerate all copies of Stand-Alone Backup after you install Version 4.1.

   To build a Stand-Alone Backup kit use the command procedure STABACKIT.COM as described in the following procedure (**STABACKIT.COM requires the Common Utilities Option**).

   1. Label two blank RX50 floppy diskettes SYSTEM_1 and BACKUP.

   2. Log in as SYSTEM and invoke STABACKIT.COM using the following command:

      ```
      $ @SYS$UPDATE:STABACKIT $FLOPPY1:
      ```

      After you have identified the target device, STABACKIT tells you that two blank floppy diskettes are needed to build the kit: the first one for the stand-alone VMS system files and the second one for the BACKUP application image.

      STABACKIT also notes that both floppy diskettes will be checked for bad blocks. If any bad blocks are found, this procedure will ask for another floppy and check the diskette for bad blocks.

   3. When STABACKIT prompts, insert the first system floppy diskette (labeled SYSTEM_1) in the drive and enter YES when you are ready to continue. The screen will appear as follows:

      ```
      Sysgen parameters for stand-alone VMS have been
      placed in the file:

              SYS$SYSROOT:[SYSUPD]VAXVMSSY.PAR

      Please place the first system floppy diskette in
      drive  _SALONE$DUA1:

      This volume will receive the volume labeled
      System_1.

      Enter "YES" when ready: YES

      Analyzing floppy diskette in _SALONE$DUA1:
      for bad blocks
      ```

      When the last file is copied, STABACKIT prompts you to insert the next floppy.

4. Insert the second floppy diskette marked BACKUP; then enter YES when you are ready to continue. The screen will appear as follows:

```
Please place the second floppy diskette in
drive _SALONE$DUA2:.

This drive will receive the volume labeled BACKUP.

Enter "YES" when ready: YES

Analyzing floppy diskette in _SALONE$DUA2:
for bad blocks
```

The next message appears indicating that the kit is built.

NOTE: At this point in the installation DIGITAL recommends use of the ANALYZE /DISK/REPAIR Utility on the system disk as described in the *VAX/VMS Verify Utility Reference Manual*.

1.6 Printing Patches Applied by the Update Kit

Requires the Secure User Environment Option.

As each image is patched, two output files are produced: the patched image and a journal file with the same name as the image and with a file type of JNL. Journal files contain a record of each patch made to native mode images during the update process, but do not contain information about modules that have been replaced.

If you want a listing of the patches produced by the update process, you must print the journal files using the following steps:

1. Complete the update procedure that installs Version 4.1, including rebooting the system, as described in Chapter 1 of the *MicroVMS User's Manual*

2. Log in to any account that has system privileges and issue the following command:

```
$ PRINT SYS$SYSTEM:*.JNL,SYS$LIBRARY:*.JNL
```

The journal files produced by the Version 4.1 update procedure occupy approximately 430 blocks; you may want to remove these files from your system disk after the update is complete.

.

Chapter 2

Problems Resolved by Version 4.1

This chapter summarizes the problems resolved by Version 4.1. Because Version 4.1 is an update and not a major release of the system, there are few functional changes to the software.

2.1 Mount Verification Canceling

An error in the mount verification canceling routine (entered via IPL 12 (hex C) from the console) causing the console terminal to hang has been corrected.

2.2 Initializing Structure Level 1 Disks

Requires the System Programming Option.

An error restricting the protection of the storage bit map and master file directory (MFD) has been corrected. Read access is now always granted to these files.

2.3 System Dump Analyzer

Requires the System Programming Option.

The following changes have been made to the System Dump Analyzer (SDA) for Version 4.1.

- When the System Dump Analyzer could not open either a dump file, the file SYS.STB, or an indirect command file for a symbol table file, the second line of the error message was displayed twice. This problem has been corrected.

- Typing [CTRL/C] while executing an indirect command file caused SDA to exit abnormally. This problem has been corrected.

- When you typed CTRL/C while reading a file with the READ command, SDA exited without allowing you to read other files. SDA now allows you to exit from the file and read another.

- System Dump Analyzer would abnormally exit from an indirect command file after you executed a carriage return to view another page. This problem has been resolved.

2.4   Installing with the /ACCOUNTING Qualifier

In Version 4.0, installing an image with the Install Utility's /ACCOUNTING qualifier had no effect. In Version 4.1, the /ACCOUNTING qualifier selectively enables image level accounting for the image.

2.5   Record Management Services

The following changes have been made to Record Management Services (RMS) for Version 4.1.

- In Version 4.0, any indirect command file that was accessed from a remote node using DECnet would fail when the first image was invoked. The first image would execute, but the remainder of the command file would be interpreted as user data (and not checked for DCL commands). RMS has been fixed so that this problem does not occur in Version 4.1.

- In Version 4.0 the SET VERIFY command had no effect on command files that were invoked from unit record devices or from remotely accessed command files. RMS has been fixed so that SET VERIFY will operate properly when command files are invoked in this way.

- In Version 4.0 image input verification would not operate properly if user mode ASTs were disabled. This was most noticeable in DEBUG command line input verification. In Version 4.1 RMS has been fixed so that user mode ASTs will not affect image input echoing.

2.6 $CHANGE_ACL

In Version 4.0, the $CHANGE_ACL system service ignored the BYPASS privilege when attempting to add an Access Control List (ACL) to an unowned device. A process also required privileges to read an empty ACL of a device. In Version 4.1 privileges are required only for adding an ACL to an unowned device.

2.7 $MGBLSC Mapping with BYPASS Privilege

In Version 4.0, the $MGBLSC system service ignored the BYPASS privilege when mapping a section. This occurred only if the normal UIC-based protection failed and the process did not have either a system UIC or the SYSPRV privilege. In Version 4.1 this problem has been corrected.

2.8 RMS $RENAME Service

In Version 4.1, if the RMS $RENAME service is used (as opposed to executing the RENAME.EXE image via a DCL command) and the file referred to by the old file name does not exist, RMS will return a status of RMS$_ACC, rather than RMS$_FNF.

2.9 Primaries Listed in File Definition Language Primaries Files

Requires the Program Development Tools Option.

For Version 4.0, Section FDL.2 of the *MicroVMS Programming Support Manual* gives a list of the FDL primaries, and states that this order is mandatory. Because of an ambiguity in the FDL syntax involving a primary which is new for Version 4.0, beginning with Version 4.1 the required order in which the primaries should be written is as follows:

1. TITLE
2. IDENT
3. SYSTEM
4. FILE
5. DATE
6. RECORD
7. ACCESS
8. SHARING
9. AREA
10. KEY

11. CONNECT

12. ANALYSIS_OF_AREA

13. ANALYSIS_OF_KEY

Please note that existing (valid) FDL files will continue to be accepted by the parser. Further, after the Version 4.1 patch is applied to FDLSHR, utilities and user programs which create FDL files with the FDL$GENERATE command will create them with the primaries in the correct order. However, those who create FDL files by hand using a text editor must adhere to the new syntax.

Chapter 3

Current Problems and Restrictions

This chapter lists known problems and restrictions in Version 4.1.

3.1 BAD_LRL Error with SORT

SORT generates a BAD_LRL error when it sorts certain sequential stream files. The error occurs for each record that ends with a $<CR> <LF>$ combination where the $<CR>$ appears in one disk block and the $<LF>$ appears in another. This error can be safely ignored. The user can work around the error by specifying a record size two bytes larger than the largest record specified in the /FORMAT=(RECORD_SIZE=n) qualifier.

3.2 Auto-login on Disconnectible Terminals

The autologin facility (described in Chapter 2 of the *MicroVMS Users Manual*) does not work on disconnectible terminals. Terminals marked disconnectible with the SET TERMINAL /DISCONNECT command cannot be controlled with the autologin feature. Therefore, to use the autologin feature on a particular terminal, it must be made nondisconnectible with the SET TERMINAL /NODISCONNECT command.

Chapter 4

Notes To Published Documentation

This chapter contains notes that correct or supplement published MicroVMS documentation (the *VAX/VMS Release Notes, Version 4.1* contain notes to information in the VAX/VMS documentation set).

4.1 Creating Group Logical Names at System Startup

Changes in the way that logical names are stored have affected certain site-specific command procedures that used the SET UIC command to create group logical names at system startup time. This note describes these changes and shows how command procedures can be altered to achieve the same effect under MicroVMS Version 4.0.

In VAX/VMS Version 3.0, there was a single table into which all group logical names were placed. This table always existed. A translation operation looked at the group number in each logical name block to see if the name belonged to the same group as the process which requested the translation.

In MicroVMS Version 4.0, there are many group tables, one for each group that exists during the life of the system. A group table is created as part of the process creation of the first process in the group.

In order to accomplish what was a simple operation in VAX/VMS Version 3.0, you need to create the group table before you can place names into that table. We have examined several solutions and have settled on the following solution for several reasons:

- The group table is set up as part of process creation and thus is identical in its properties with group tables created later in the life of the system.

- The solution offers more flexible control of logical names and who has privilege to change the initial settings.

- The solution does not require the use of the SET UIC command.

Assume that your VAX/VMS Version 3.0 solution was creating group logical names with a sequence of commands in SYSTARTUP.COM (or a similar file) that looked something like this:

```
$ SET UIC [AAA,xxx]
$ DEFINE /GROUP AAA_logical_name_1 AAA_equivalence_name_1
$ DEFINE /GROUP AAA_logical_name_2 AAA_equivalence_name_2
$    .
$    .
$    .
$ DEFINE /GROUP AAA_logical_name_n AAA_equivalence_name_n
$
$ SET UIC [BBB,xxx]
$ DEFINE /GROUP BBB_logical_name_1 BBB_equivalence_name_1
$ DEFINE /GROUP BBB_logical_name_2 BBB_equivalence_name_2
$    .
$    .
$    .
$ DEFINE /GROUP BBB_logical_name_n BBB_equivalence_name_n
```

First of all, you need to place into separate files all of the SYSTARTUP.COM commands (or whatever file contained the VAX/VMS Version 3.0 commands) that create logical names for other groups. For example, the commands shown above could be put into files as follows:

```
          Contents of SYS$MANAGER:AAA_GROUP_NAMES.COM
          -------------------------------------------
$ DEFINE /GROUP AAA_logical_name_1 AAA_equivalence_name_1
$ DEFINE /GROUP AAA_logical_name_2 AAA_equivalence_name_2
$    .
$    .
$    .
$ DEFINE /GROUP AAA_logical_name_n AAA_equivalence_name_n
          Contents of SYS$MANAGER:BBB_GROUP_NAMES.COM
          -------------------------------------------
$ DEFINE /GROUP BBB_logical_name_1 BBB_equivalence_name_1
$ DEFINE /GROUP BBB_logical_name_2 BBB_equivalence_name_2
$    .
$    .
$    .
$ DEFINE /GROUP BBB_logical_name_n BBB_equivalence_name_n
$    .
$    .
$    .
```

The system manager can grant (or deny as appropriate) members of each group **write** access to the files so that group members can modify the names that they start out with. (**Write** access to these files should be based on approximately the same criteria currently used to grant these same users GRPNAM privilege.)

Finally, SYSTARTUP.COM (or whatever file contained the VAX/VMS Version 3.0 commands) would be modified to contain the following commands.

```
$ RUN /UIC=[AAA,xxx] -
        /INPUT=SYS$MANAGER:AAA_GROUP_NAMES.COM -
        SYS$SYSTEM:LOGINOUT

$ RUN /UIC=[BBB,xxx] -
        /INPUT=SYS$MANAGER:BBB_GROUP_NAMES.COM -
        SYS$SYSTEM:LOGINOUT
```

This causes the UIC to be changed for each detached process in a clean way.

In fact, you can even let these names get created in a batch job, as in the following example. (These commands would follow the startup of the various batch queues.)

```
$ SUBMIT /USER=SMITH_AAA SYS$MANAGER:AAA_GROUP_NAMES
$ SUBMIT /USER=SMITH_BBB SYS$MANAGER:BBB_GROUP_NAMES
```

4.2   ALLOCATE/GENERIC Qualifier Not Documented Correctly

The description of the /GENERIC qualifier of the ALLOCATE command in Appendix DCL of the *MicroVMS User's Manual* and online HELP is incorrect. The *device-type* value should not be specified with the /GENERIC qualifier (as documented), but should be specified as a parameter to the ALLOCATE command. The example in the DCL appendix should read:

```
$ ALLOCATE/GENERIC RX50 ACCOUNTS
```

This command allocates the first free diskette drive and equates its name to the process logical name ACCOUNTS.

4.3   Change in Mail Documentation

In Section 4.10.1.5 of the *MicroVMS User's Manual*, the following sentence does not apply and should be deleted.

"You can specify only one distribution list in a response to the TO: prompt".

MicroVMS allows you to send mail to more than one distribution list and the lists need not be nested.

Appendix A

MicroVMS Version 4.1 Mandatory Update Listing

The following information applies only if you have ordered the VMS V4.0 microfiche kit.

Requires the System Programming Option.

This appendix is a list of the descriptions of changes made to Version 4.1, but not reflected in the microfiche kit. In order to see the binary changes please use the command:

```
$ ANALYZE/IMAGE file-spec
```

The following patches were made after the microfiche kit was manufactured.

```
 1) ANALYZOBJ (patch image)
    ! ANALYZOBJ.EXE
    !
    !   ECO02   MGN0006         29-OCT-1984
    !           MODULE: EXEINPUT
    !           This patch resolves SPR #CS00321, so that ANALYZE/IMAGE
    !           can analyze fixup sections larger than 127.
    !
    !   ECO01   MGN0005         29-OCT-1984
    !           MODULE: EXESTUFF
    !           ANALYZE/IMAGE now returns all images with V4 image header
    !           format so we do not have to distinguish between versions
    !           for analysis.
 2) ANALYZRMS (patch image)
    ! ANALYZRMS.EXE
    !
    !   ECO02   DGB0088         22-OCT-1984
    !           MODULE: RMSINPUT
    !           Use the extended UIC format in ANALYZE/RMS/FDL.
    !
    !   ECO01   DGB0085         27-SEP-1984
    !           MODULE: RMS2IDX
    !           Unwind the stack (that is, quit trying to analyze the file)
    !           if the area descriptor of an indexed file contains an
    !           illegal bucket size.
```

```
3) AUTHORIZE (new image)

   ! AUTHORIZE.EXE
   !
   !   ECO01   JRLOO38            18-Oct-1984
   !           MODULE: UAFPARSE
   !           Because of a lost update packet in the source manager,
   !           some security auditing features were left out of V4.0.
   !           They are restored here.

4) BACKUP (patch image)

   ! BACKUP.EXE
   !
   !   ECO04   LYOO4           4-OCT-1984 15:43
   !           MODULE  : LIST, MAIN
   !           ROUTINE : FIN_LIST, MAIN
   !           Reset file and block counter after listing each save set
   !           Add "End of BACKUP operation" message.
   !           Print "End of..." message after a save or copy operation
   !           not just after listing an already existing journal or
   !           save set.
   !           ************** CAUTION *************************
   !           The unused byte named COM_PADDING in the common area
   !           is being used in this patch for a flag.  It is at address
   !           000003C8.
   !           ************** CAUTION *************************
   !
   !   ECO03   LYOO3           2-OCT-1984 12:57
   !           MODULE  : LIST, COMMAND
   !           ROUTINE : INIT_LIST, CHECK
   !           If listing a copy operation then say
   !           "Listing of BACKUP operation" instead of
   !           "Listing of save set"
   !
   !   ECO02   LYOO2           1-OCT-1984 10:36
   !           MODULE:  STAACP
   !           ROUTINE: READY_DISK
   !           Always issue an IO$_AVAILABLE before requesting a new
   !           disk be mounted.  This avoids the endless loop on RX50s
   !           if an error is detected on a disk.
   !
   !   ECO01   LYOO01          1-OCT-1984 09:05
   !           MODULE: SAVE
   !           Update version number to V4.1

5) BADBLOCK (new image)

   ! BADBLOCK.EXE
   !
   !   ECO01   HHOO59          01-OCT-1984
   !           MODULE: SCANFILE
   !           Enhance BADBLOCK to handle a larger class of I/O errors.
   !           Also, rework completion routines.
```

```
6) COBRTL (patch image)

   ! COBRTL.EXE
   !
   !    EC001    TS             02-Nov-1984
   !             MODULE:  COB$DISPLAY
   !             In COMMON_SCREEN, create a
   !             larger output buffer to prevent
   !             ACTVIO when buffer overflows.

7) CONVSHR (patch image)

   ! CONVSHR.EXE
   !
   !    EC001    JWT0198           7-Nov-1984
   !             MODULE: CONV$FSTIO
   !             Add to area total allocation ONLY on extends.
   !             The previous logic would add an initial allocation
   !             twice.
   !
   !    EC002    JWT0199           7-Nov-1984
   !             MODULE: CONV$MAIN
   !             Fix bogus FORTRAN carriage control character for
   !             conversion.
   !
   !    EC003    JWT0200           26-Nov-1984
   !             MODULE: CONV$FSTIO
   !             Fix bad VBN-size formula.  CONVERT made 4-byte VBN
   !             pointers when VBN was greater than 00100000.

8) COPY (patch image)

   ! COPY.EXE
   !
   !    EC002    TSK0016           8-Nov-1984
   !             Make sure directory files are checked against the command line
   !             criteria.  Since directory files are created in COPY$COPY and
   !             all other output files are created in COPY$OPN_OUTFIL,
   !             COPY$CHECK_FILE_FOR_MATCH must be called from COPY$COPY.
   !
   !    EC001    MGN0003          11-Sep-1984
   !             MODULE: COPYMAIN
   !             This fixes an access violation when copying files from the
   !             CI port.  The routine COPY$INOPN_ERR was taking the NAM
   !             block from the RAB instead of using the FAB.

9) CREATE (patch image)

   ! CREATE.EXE
   !
   !    EC0001   SHZ             21-Sep-1984
   !             MODULE:  CREATE_DIR
   !             Issue the $PARSE with SYNCHK bit set in $NAM for CREATE/DIR
   !             so directory gets created in correct spot when using
   !             searchlists.
```

10) DCL (patch image)
```
    ! DCL.EXE
    !
    !   ECO0O1  HWS            18-Sep-1984
    !           MODULE: STATUS
    !           Remove special handling of forced exit and exceeded CPU
    !           errors.
    !
    ! Patch STATUS
    !
```

11) DCLTABLES (miscellaneous fix)
```
    ! DCLTABLES
    !
    !   ECO01   HWSO111          O8-NOV-9184
    !           Update the command definition for DELETE.
    !
    !   ECO02   HWSO112          O8-NOV-1984
    !           Update the command definition for SHOW.
    !
    !   ECO03   HWSO113          O8-Nov-1984
    !           Update the command definition for DCLINT.
```

12) DEBUG (patch image)
```
    ! DEBUG.EXE
    !
    ! module DBGADDEXP
    ! Fix a problem in which an EXAMINE of PASCAL constant arrays can fail.
    !
    ! module DBGCALL
    ! Fix a problem in which the CALL command clobbers register R1 (that is,
    ! once you get into the called routine R1 has a different value than
    ! it had right before the call).
    !
    ! module DBGDEFINE
    ! Fix a problem where the string was always being upcased in
    ! DEFINE/COMMAND.
    !
    ! module DBGEVALOP
    ! Fix a problem in which WHEN clauses did not work for languages COBOL and
    ! RPG.
    !
    ! module DBGEVENT
    ! Fix a problem in which exception breaks could be missed.
    !
    ! module DBGEVENT
    ! Fix a problem with watchpoints that span more than 1 page.
    !
    ! module DBGIFTHEN
    ! Fix a problem in which FOR loops didn't work for COBOL, PLI, and RPG.
    !
    ! module DBGLEVEL1
    ! Bump version number to 4.1
    !
    ! module DBGNHELP
    ! Fix a bug where the HELP command can sometimes ACCVIO.
    !
```

```
              ! module DBGSTO
              ! Enable the EDIT command.
              !
              ! module DBGVALUES
              ! Fix a bug in which an EXAMINE of packed decimal numbers can give
              ! the wrong answer.
              !
              ! module RSTCNTRL
              ! Fix a bug in ANALYZE/PROCESS (which uses the debugger).
              ! This patch uses a patch area extending from 0C94 to 0Cxx.
              !
              ! module DBGSOURCE
              ! Fix a bug in the EDIT command.

13) DEBUGHLP (miscellaneous fix)

      !
      !   DEBUG.EXE
      !
      ! Insert a help module into SYS$HELP:DEBUGHLP

14) DIRECTORY (patch image)

      ! DIRECTORY.EXE
      !
      !   ECO01   LMP0311          5-Nov-1984
      !           MODULE: DISPLAY
      !           This ECO is one part of a two part fix to correct a bug in
      !           displaying information about magtape files (during a full
      !           display).  This part clears the high word of the file
      !           characteristics, because the magtape ACP returns it using
      !           the ODS-1 (16-bit) format without clearing the high word.

15) DUDRIVER (patch image)

      ! DUDRIVER.EXE
      !
      !   ECO01   ROW0418          6-NOV-1984
      !           MODULE: DUDRIVER
      !           Correct MSCP$K_MD_SPNDW (spin down modifier) value so that
      !           DISMOUNT /UNLOAD does.

16) DZDRIVER (patch image)

      !DZDRIVER.EXE
      !   ECO01   MIR0001          30-NOV-1984
      !           MODULE: DZVDRIVER
      !           Due to a timing problem with devices on the
      !           QBUS raise the device IPL of the DZV from
      !           21 to the IPL that everything comes in on,
      !           IPL 23.

17) EDTSHR (patch image)

      ! EDTSHR.EXE
      !
      !   ECO01   KEB0001          19-DEC-1984
      !           MODULE: LSHOW
      !           Patch version number so that it is consistent with other
      !           operating systems that ship EDT V3.
```

18) ERF (new image)

    ! ERF.EXE
    !
    !   ECO01   SARO293   2-Nov-1984
    !   MODULE: ERF
    !   Made /EXCLUDE=UNKNOWN work.

19) ERFPROC1 (new image)

    ! ERFPROC1.EXE
    !
    !   ECO01   SARO292   2-Nov-1984
    !   MODULE: ERFPROC1
    !   Added TA78 device dependent support.

20) F11BXQP (new image)

    ! F11BXQP.EXE
    !
    !   ECO01   ACGO468   18-Sep-1984
    !   MODULES: QUOTAUTIL, CLENUP
    !   Count a write access to the quota file when quotas are
    !   enabled, to prevent the quota file from being deleted
    !   when quotas are enabled anywhere in a cluster.
    !
    !   ECO02   ACGO470   27-Sep-1984
    !   MODULE: RWATTR
    !   Correct the file access checks related to ACL modification.
    !
    !   ECO03   CDSOOO1   10-Nov-1984
    !   MODULES: DELBAD, TRUNC
    !   Fix bugs affecting bad block handling. Make multiheader
    !   BADBLK.SYS files work correctly. Also update revision
    !   date on BADBLK.SYS when bad blocks are deallocated to it.
    !
    !   CDSOOO2   10-Nov-1984
    !   MODULE: DEACCS
    !   Fix bug related to deferred truncation in a cluster.
    !   One symptom was the failure of a log file to be
    !   truncated to end of file if it was being read on
    !   another node when the log file was closed.
    !
    !   CDSOOO3   10-Nov-1984
    !   MODULE: RDBLOK
    !   Add ambiguity resolution logic. This fixes race
    !   conditions where an extension header is simultaneously
    !   being requested under different serialization locks.
    !
    !   CDSOOO4   14-Nov-1984
    !   MODULES: RDBLOK, RWVB, ACCESS, MAKACC
    !   Correctly invalidate file system cache in noncluster
    !   and directory file situations.
    !
    !   CDSOOO5   15-Nov-1984
    !   MODULE: CLENUP
    !   Fix problem where delete of multiheader file returned
    !   spurious "no such file" error even though it succeeded.
    !

```
!          CDS0006          15-Nov-1984
!          MODULE: EXTHDR
!          Correctly set reference count when extension fcb is
!          created during file extension.  This caused spurious
!          "no such file" error trying to access those headers.
!
!          CDS0007          15-Nov-1984
!          MODULES: ACCESS, CHARGEQ, CREHDR, QUOTAUTIL, SMALOC
!          Expand test for clusterness so that extent and fid
!          caching are correctly enabled in certain noncluster
!          environments (MicroVAX, for example).
!          Also fix sense of test which was not doing a cluster
!          wide flush of file id caches when the index file
!          is extended.
!
!          CDS0008          2-Jan-1985
!          MODULES: RDBLOK
!          Perform buffer search again after writing buffer due
!          to minimal cache conditions.
!
!          CDS0009          2-Jan-1985
!          MODULE: CREATE
!          Fix bug which caused diskquota leakage when creating
!          file with a different UIC as owner.
```

21) FAL (patch image)

```
! FAL.EXE
!
!   ECO 01  JEJ0057          18-Oct-1984
!          MODULE: FALACTION
!          Return the resultant file name whether or not an error is
!          encountered during the $SEARCH operation.
!
```

22) FDLSHR (patch image)

```
! FDLSHR.EXE
!
!   ECO01  JWT0197          7-Nov-1984
!          MODULE: FDLGEN
!          Change the order of the FDL primaries to avoid
!          an ambiguity in the language.
```

23) FILMNTMSG (patch image)

```
! FILMNTMSG.EXE
!
!   ECO01  DGB0089          22-OCT-1984
!          MODULE: Patch for ANALYZRMS.
!          Fix ANALYZE/RMS to use the new FAO directive which
!          allows the extended UICs which are new in V4.
```

24) FORRTL (patch image)

```
! FORRTL.EXE
!
!   ECO01    SBL          27-Sep-1984
!            Module: FOR$BACKSPACE (1-011)
!            When doing BACKSPACE of a SEGMENTED file,
!            store the RFAs of the last physical record
!            in each logical record rather than the first.
!            This fixes the bug where BACKSPACE on segmented
!            files of very long (>2044 bytes) records would
!            misposition the file.
!
!   ECO02    MDL          15-Nov-1984
!            Module: FOR$IO_ELEM (2-048)
!            Zero out the record size field in the RAB when a zero-length
!            element is transmitted and the lower levels of abstraction
!            are skipped.
!
!   ECO03    MDL          15-Nov-1984
!            Module: FOR$$OPEN_DEFLT (1-099)
!            Correct the check for invalid key position, size or datatype
!            when opening existing indexed files.
```

25) JOBCTL (patch image)

```
! JOBCTL.EXE
!
!   ECO01    JAK0001          20-SEP-1984
!            MODULE: EXECUTOR
!            Allow a process 5 seconds to be run down before reissuing the
!            forced exit call in the next higher access mode.
```

26) LBRSHR (patch image)

```
! LBRSHR.EXE
!
!   ECO 3    GJA0103          09-Nov-1984
!            MODULES: LBR_GETHELP & LBR_OUTPUTHELP
!            - Change the prompt string from two CR/LF pairs to one
!            to allow line editing to work.
!            - Add some additional blank line output calls to try to
!            maintain a similar appearance to the two pair case.
!
!   ECO 2    GJA0102          09-Nov-1984
!            MODULE: LBR_GETHELP
!            - Move the code segment in LBR$GET_HELP which checks for
!            maximum key length to after the check for the key's existence.
!
!   ECO 1    GJA0101          09-Nov-1984
!            MODULE: LBR_INDEX
!            - Initialize the class and dtype fields in the (stack local)
!            descriptor passed back to the user routine in LBR$GET_INDEX.
```

```
27) LIBRTL (patch image)

    ! LIBRTL.EXE
    !
    !   EC002    LMP0313          5-Nov-1984
    !            MODULE: LIB$CREATE_DIR
    !            Ensure that on a particular error patch, the error status is
    !            returned, not the address of the local storage cell.
    !
    !   EC001    MDL              30-Oct-1984
    !            Module: OTS$CVTLT
    !            In OTS$CVT_L_TI, Prevent a plus sign from appearing
    !            in an otherwise blank field.

28) LINK (patch image)

    ! LINK.EXE
    !
    !   EC001    ADE0001          05-Nov-1984
    !            MODULE: LNKVMALLO
    !            Call LNK$GETFIXSIZE to determine the size of the fixup ISD
    !            for based images.  This calculation was being done inline
    !            and  was incomplete, leading to occasional underestimates
    !            of the fixup ISD size.

29) LTDRIVER (new image)

    ! LTDRIVER.EXE
    !
    !   EC001    JAY0001          12-NOV-1984
    !            MODULE: LTDRIVER
    !            Fix PASSALL handling.

30) MACRO32 (patch image)

    ! MACRO32.EXE
    !
    !   EC005    RRB0005          08-Oct-1984
    !            This patch resolves QAR #047 (FFT).
    !            Correctly parse global listing and object qualifiers
    !            only if this is the first input file of file list.
    !            Avoids naming output files after second, third... input
    !            file.  Also, correctly set the OFP bit in MAC$LIST_FAB
    !            rather than MAC$OBJECT_FAB in procedure GET_LISTING.
    !
    !   EC004    MGN0004          20-Sep-1984
    !            MODULE: ACTSTA
    !            This patch resolves SPR #11-62845.
    !            Fix problems with the use of the ASCII unary operator
    !            in an expression of an operand or .ADDRESS directive.
    !
    !   EC003    RRB0003          11-Sep-1984
    !            MODULE: ERROR
    !            This patch resolves Baselevel QAR #0111.
    !            Correctly signal an error finding a library
    !            specification.  Signal the error with the proper
    !            level of indirection.
    !
```

```
!   ECO02   MGN0002          11-Sep-1984
!           MODULE: ACTSTA
!           This patch resolves SPR #11-67895.
!           Do not allow local labels as symbols in the .ENTRY
!           directive.
!
!   ECO01   MGN0001          01-Sep-1984
!           MODULE: MACDEF
!           This patch resolves SPR #11-63941.
!           If the arguments in a macro definition form a line
!           more than 140 characters long, the MACRO assembler
!           returns an internal logic error.
!           The fix consists of making sure there is enough space
!           in the buffer where the arguments are stored to include
!           the link flag and the link address for the next buffer.
```

31) MAIL (patch image)

```
! MAIL.EXE
!
!   ECO 3   ROP0041          29-OCT-1984
!           Close temporary file at end of inbound server
!           loop.
!
!   ECO 2   ROP0037          04-OCT-1984
!           Upon the request for information on the current message
!           return a null descriptor if the TO field of the current
!           message is not genuine.
!
!   ECO 1   ROP0036          04-OCT-1984
!           Make sure new wastebasket names do not exceed MFR_S_FILEKEY
!           characters in length.
!           Modify only info record if SET WASTEBASKET is to the current
!           wastebasket name.
```

32) MTAAACP (patch image)

```
!MTAAACP.EXE
!
!   ECO01   MMD0336          8-November-1984
!           MODULE:MAIL
!           Fix for a race condition between DISMOUNT and REPLY/ABORT
!           where if the DISMOUNT is issued before the REPLY/ABORT
!           is finished the system could crash. The problem is that
!           the REPLY/ABORT does not set the VCB cancel I/O in progress
!           bit. Thus DISMOUNT will clear the databases out from under
!           the cancel.
!
!           MMD0334          8-NOVEMBER-1984
!           MODULE:IODONE
!           Change copy of local FIB to user's buffer to copy
!           only the magtape length of the FIB.
```

33) NCP (patch image)

```
! NCP.EXE
!
!   ECO01   TRC0001          09-Nov-1984
!           MODULE: NCPNETIO
!           Fix access violation when CLEAR KNOWN LINES ALL done on RSX
!           systems.
```

```
34) NETACP (patch image)

    ! NETACP.EXE
    !
    !   ECO03   PRB0003          2-NOV-1984 15:19
    !           MODULE: NETDLLTRN.MAR
    !           Fix state table to cycle line if the start sequence gets
    !           out of sequence (get a verify message ahead of a start
    !           message for some reason).
    !
    !   ECO02   PRB0002          2-NOV-1984 14:22
    !           MODULE: NETACPTRN.MAR
    !           Test for SS$_DEVNOTMNT as well as SS$_NOSUCHDEV on failures
    !           to connect to PSI. This accommodates the case where PSI was
    !           previously running but has since been shut down.
    !
    !   ECO01   PRB0001          18-SEP-1984 21:35
    !           MODULE: NETCONECT.MAR
    !           Use EPID, not IPID, in type 1 NCBs. This way, EPID will
    !           appear in SYS$NET.

35) NETCONFIG (edit text file)

    ! NETCONFIG.COM
    !
    !   ECO01   PRB0001          2-NOV-1984 14:42
    !           MODULE: SYS$MANAGER:NETCONFIG.COM
    !           Fix bug which prevented configuration of nodes with
    !           addresses 1.1, 2.2, 3.3, 4.4, etc.

36) NODRIVER (patch image)

    !NODRIVER.EXE
    !
    !   ECO02   MMD0332          27-September-1984
    !           Fix to the NODRIVER's SENSEMODEFDT routine to fix a
    !           CLRW off a location with no register. Fix to GETNXT
    !           to fix an instruction which was using the wrong
    !           register. Delete the instruction in SHUTDOWN_1INE
    !           which clears the driver's nounload flag.
    !
    !           MMD0333          5-November-1984
    !           Module:DDCMP
    !           Fix to reset the DDCMP counters in DDCMP$STARTINT before
    !           any transmit requests are queued.

37) OPCOM (patch image)

    ! OPCOM.EXE
    !
    !   ECO01   JRL0040          1-Nov-1984
    !           Module: SECURITY
    !           Re-establish temporarily tied-off AUTHORIZE field auditing.
```

38) PASRTL (patch image)

```
! PASRTL.EXE
!
!    ECO01    SBL             19-Dec-1984
!             Module: PAS$OPEN2 (1-017)
!             Check for "file-oriented-device" when a USER_ACTION
!             routine is specified, as well as when it is not.  This
!             fixes the problem where RESET does not reposition the file
!             to the beginning if USER_ACTION was used.
```

39) PUDRIVER (patch image)

```
! PUDRIVER.EXE
!
!    ECO01    EMBO117            7-NOV-1984
!             MODULE: PUDRIVER
!             Define two new devices.
!
!    Define basic driver offset
```

40) REMOVE (edit text file)

```
! REMOVE.COM
!
!    ECO01    JESOOO1          2-JAN-1984
!
!             Add in prompt for lost OPTION name for possible error
!             that occurs when installing a 4.0 option on a system
!             running a higher version.
!             Make UTIL a valid option.
!             Ask entire option question for each option being deleted.
```

41) RMS (patch image)

```
! RMS.EXE
!
!    ECO01    JEJO056           01-Oct-1984
!             MODULE: NTODISCON, NTODAPIO
!             Correct error recovery code to properly construct and
!             send the CONTINUE error recovery messages.
!
!    ECO02    JEJO059           08-Oct-1984
!             MODULE: RMOCACHE
!             Correct a small timing window which can allow a stream
!             hang if both multistreaming and deferred writeback are
!             in use.
!
!    ECO03    JEJO057           18-Oct-1984
!             MODULE: NTOSEARCH
!             Correct the handling of errors during a wildcard search
!             so that nonfatal errors, such as RMS$_PRV, do not result
!             in a hung transfer.
!
!    ECO04    JEJO060           18-Oct-1984
!             MODULE: RM1GETINT
!             Correct the EOD check logic to use the remote device
!             characteristics for a DAP-accessed file to determine if
!             the input stream should be checked for an end of program
!             input state.  This is to allow remote command procedures
!             to work correctly.
!
```

```
!   ECO05   JEJ0062          18-Oct-1984
!           MODULE: RMODUMMY
!           Set the RMS version number to V4.1.
!
!   ECO06   JEJ0061          22-Oct-1984
!           MODULE: RMOACCESS
!           Force GET access to terminals and network files if DCL
!           requests EXE access.  This is to allow SET VERIFY with
!           @TT: to work together.
!
!   ECO07   DGB0090          30-Oct-1984
!           MODULE: RMOXPFN
!           Only skip translating SYS$DISK if the file name has both
!           a directory and a device name.  Otherwise, if SYS$DISK is
!           a search list which includes a directory specification,
!           we will lose it.
!
!   ECO08   JEJ0063          09-Nov-1984
!           MODULE: RMOEXTRMS
!           When echoing SYS$INPUT to SYS$OUTPUT (for SET VERIFY=IMAGE),
!           declare the echo AST in the normal process permanent file
!           mode, which is supervisor, instead of in user mode.  This is
!           to correct the problem with echoing DEBUG commands.
!
!   ECO09   JWT0201          07-Dec-1984
!           MODULE: RMSORENAM
!           Change error returned when attempting to $RENAME a
!           nonexistent file.  V4.0 $RENAME returned ACC, this
!           patch corrects the code to FNF.
!
!   ECO10   JEJ0064          27-Dec-1984
!           MODULE: RM2CONN
!           Correct the error in RM$CONNECT2 which caused an RMS bugcheck
!           if a relative file were opened for block or record I/O and
!           connected for both.  This is a restriction and will now
!           correctly return an error when this is attempted.
42) RTPAD (patch image)

!   RTPAD.EXE
!
!   ECO01   JLV0001          12-Nov-1984
!           MODULE: CTSENSERT
!           Fix offset problem in setting up parse of READ CHAR
!           message.  Remove increment of error count intended
!           for CTDRIVER only.
43) SDA (patch image)

!   SDA.EXE
!
!   ECO03   MSH0088          6-NOV-1984
!           MODULE: PROCESS
!           Restore current process context at a critical point in
!           processing the SHOW SUMMARY/IMAGE command. This prevents
!           incorrect stack displays in case the user issues a SHOW
!           STACK command before SHOW SUMMARY/IMAGE is completed.
!
```

```
!   ECO02   EMB0115          6-NOV-1984
!           MODULE: CLUSTER
!           Fix a misdirected branch which causes the SDA image to
!           exit if an invalid address is specified on the SHOW
!           RSPID/CONNECTION=address command.
!
!   ECO01   JJ00001          18-SEP-1984
!           MODULE: HANDLER
!           Avoid extraneous redisplay of error messages by setting
!           the "inhibit message display" bit for all conditions in
!           the signal array passed to HANDLER.
!
!           MODULE: COMMANDS
!           Use SDA error messages rather than generic (SHR) ones
!           on file open failures. Clean up if ^C is typed while
!           reading a (.STB) file.
!
!           MODULE: MAIN
!           Avoid RMS-F-CCR error when execution of an indirect
!           (command) file requires a page break (by not prompting
!           in this case). Cleanly abort execution of an indirect
!           file when ^C is typed by signaling EOF.
```

44) SDLNPARSE (new image)

```
! SDLNPARSE.EXE
!
!   ECO01   KDM0001          10-OCT-1984
!           Provide image left off MicroVMS PROG kit.
!
```

45) SECURESHR (patch image)

```
! SECURESHR.EXE
!
!   ECO01   LMP0310          5-Nov-1984
!           MODULE: SYSACLSRV
!           Change $CHANGE_ACL so that privs are not required to read a
!           null ACL of an unowned device.  Also, allow BYPASS to change
!           the ACL of an unowned device (in addition to SYSPRV).
```

46) SET (patch image)

```
!SET.EXE
!
!   ECO02   MMD0337          1-Jan-1984
!           MODULE: SWITCH_TERMINAL
!           Delete code which changes the DDT address in the DDB
!
!   ECO01   AEW0006          07-Nov-1984
!           MODULE: SETFILE
!           This change enables the /NOEXPIRATION_DATE qualifier for
!           SET FILE.  It enables any expiration date on a file to be
!           negated, leaving no expiration date specified.
```

47) SETPO (patch image)

```
!SETPO.EXE
!
!    ECO01    AEWOOO2           O9-Nov-1984
!             MODULES: SETPOMESS
!                      SETPODISP
!             This patch eliminates the double error messages received when
!             an error is encountered in SET PASSWORD.  This is the result of
!             a change to SET MESSAGE enabling the reporting of error messages
!             so the code patched here is in the SETPO DISPATCH module and
!             the SET MESSAGE module.
```

48) SHOW (patch image)

```
! SHOW.EXE
!
!    ECO01    PLL   O7-Nov-1984
!             Module:  SHOWTERM
!             Fix to eliminate an access violation when
!             SHOW TERMINAL is dealing with a terminal
!             defined in TERMTABLE (that is, a non-DEC terminal).
!
! addr of module SHOWTERM
```

49) SMBSRVSHR (patch image)

```
! SMBSRVSHR.EXE
! Print Symbiont is a shareable image.
!
!    ECOOOO2 MODULE:FORMAT
!             RRBOOO2          30 Oct 1984
!             Correctly position output to the first printable line of
!             the paper for FTN and PRN carriage control type files.
!
!    ECOOOO1 MODULE:FORMAT (PSM$MAIN_FORMAT)
!             MODULE:DISPATCH (CARRIAGE_CONTROL)
!             RRBOOO1          30 Oct 1984
!             Correctly translate NEWLINE into carriage return--line feed
!             pair.  Fixes output of PRN and FTN type files to correctly
!             print on terminal executor queues.
```

50) SMGSHR (new image)

```
! SMGSHR.EXE
!
!    2-019    PLL           5-Nov-1984
!             Module: SMG$INPUT
!             In SMG$READ_STRING, don't use ESCAPE as the
!             default modifier if the terminal is set to
!             PASTHRU and NOESCAPE.
!             Also, SMG$CREATE_VIRTUAL_KEYBOARD should not try
!             to set the keypad mode for a VT173, since it
!             had no keypad.
!
!    1-018    PLL           21-Oct-1984
!             Module: SMG$MIN
!             Fix an internal routine, SMG$$OUTPUT_MINIMAL_UPDATE,
!             to perform QIOs at the end of a line, rather than
!             fragmenting the line into many small QIOs.
!
```

```
!   1-016   STAN            21-Oct-1984
!           Module: SMG$$MINIMUM_UPDATE
!           Stop an internal routine, SMG$$PUT_SCREEN,
!           from outputting the terminal sequence to return to
!           normal video rendition after each character.  Also,
!           it now outputs the normal sequence only when some other
!           rendition was previously in effect.
!
!   1-013   PLL             10-Oct-1984
!           Module: SMG$$PUT_TEXT_TO_BUFFER
!           A fix to this internal routine prevents an access
!           violation which was occurring when a bell character
!           was written to a virtual display.
!
!   1-073   PLL             14-Oct-1984
!           Module: SMG$DISPLAY_OUTPUT
!           SMG$CONTROL_MODE now will correctly validate the bits
!           passed to it.  Previously it did not allow some valid
!           combinations.
!
!   1-097   PLL             10-Oct-1984
!           Module: SMG$DISPLAY_LINKS
!           SMG$RESTORE_PHYSICAL_SCREEN now leaves the scrolling
!           region as the entire physical screen, rather than in
!           some random place.  This causes subsequent programs to
!           appear correctly on the screen.
51) STABACKUP (patch image)

!  STABACKUP.EXE
!
!   EC004   LY004           4-OCT-1984 15:43
!           MODULE  : LIST, MAIN
!           ROUTINE : FIN_LIST, MAIN
!           Reset file and block counter after listing each save set
!           Add "End of BACKUP operation" message.
!           Print "End of..." message after a save or copy operation
!           not just after listing an already existing journal or
!           save set.
!           ************** CAUTION **************************
!           The unused byte named COM_PADDING in the common area
!           is being used in this patch for a flag.  It is at address
!           0000185C.
!           ************** CAUTION **************************
!
!   EC003   LY003           2-OCT-1984 12:57
!           MODULE  : LIST, COMMAND
!           ROUTINE : INIT_LIST, CHECK
!           If listing a copy operation then say
!           "Listing of BACKUP operation" instead of
!           "Listing of save set"
!
```

```
!   ECOO2   LYOO2            1-OCT-1984 10:36
!           MODULE:  STAACP
!           ROUTINE: READY_DISK
!           Always issue an IO$_AVAILABLE before requesting a new
!           disk be mounted.  This avoids the endless loop on RX50s
!           if an error is detected on a disk.
!
!   ECOO1   LYOO1            1-OCT-1984 09:07
!           MODULE: SAVE
!           Update version number to V4.1
```

52) SYLOGIN (miscellaneous fix)

```
! SYLOGIN.COM
!
!   ECOO1   DGBO1O9            22-DEC-1984
!           Correct spelling of INITIALIZE symbol.
```

53) SYS (patch image)

```
! SYS.EXE
!
!   ECOO3   ACGO469            21-Sep-1984
!           MODULE: SYSDASSGN
!           When waiting for pending I/O to finish on a channel
!           being deassigned, reverify the channel on each check.
!           Failing to do so could result in the channel being
!           deassigned twice, resulting in a negative reference
!           count on the device.
!
!   ECOO4   TCMOOO1            27-Sep-1984
!           MODULE: MEMORYALC
!           When pool expansion fails, if we're still in INIT, don't
!           broadcast a message to the console terminal (this causes an
!           access violation since the OPAO data structures have not been
!           initialized yet).
!
!   ECOO5   EMB65497            29-Oct-1984
!           MODULE: SYSASCEFC
!           This patch resolves SPR 11-65497.
!           Save the contents of R4 before the call to allocate
!           a common event block.  This call will destroy the contents
!           of R4 if a wait occurred.  Restore R4 to its previous
!           value after the call.
!
!   ECOO6   Bundled updates
!
!           LMPO314            6-Nov-1984
!           MODULE: IPCONTROL
!           Fix a bug which caused console interrupts to be disabled when
!           issuing the second (or succeeding) prompt.  Also, fix a bug
!           in which the stack was cleaned up before restoring the saved
!           console status (from the stack).
!
```

```
!               LMPO312          5-Nov-1984
!               MODULE: SYSCRMPSC
!               This patch insures that the BYPASS privilege can be used when
!               doing the protection check to validate access to the section.
!
!               SSAO032          7-Nov-1984
!               MODULE: ACCOUNT
!               MODULE: SYSRUNDWN
!               These patches fix a bug that inhibited the writing of image
!               accounting records for images INSTALLed with /ACCOUNTING.
!
!               JLVOOO1          12-Nov-1984
!               MODULE: SYSBRKTHR
!               Fix missing check for OPER privilege. Fix exit
!               path to handle this case.
!
!   ECOO7   HWSOO1           27-NOV-1984
!               MODULE: INIT, SYSCOMMON
!               Set version to X4.1
!
!   ECO  9   LJKxxxx
!               Bundle three unrelated patches, all of which fix problems
!               that will cause the system to crash.
!
!   9 a     LJK4OO1          3-Dec-1984
!               Module:          SYSDEVALC
!               Traverse I/O database in a way that is protected from
!               dynamic UCB deletion.
!
!   9 b     LJK4OO2          4-Dec-1984
!               Modules:         SYSDELPRC, SYSRUNDWN
!               Only call the privileged rundown vectors once when a process
!               is being deleted. This prevents a call in kernel mode to code
!               that no longer exists.
!
!   9 c     LJK4OO3          5-Dec-1984
!               Module:          PDAT
!               Prevent automatic working set adjustment on the null
!               process by setting the DISAWS bit in PCB$L_STS in the
!               PCB of the null process.
!
!   ECO10   HWSOO2           O5-JAN-1985
!               MODULE: INIT, SYSCOMMON
!               Set version to V4.1
54) SYSGEN (patch image)
    ! SYSGEN.EXE
    !
    !   ECOO1   WHMOOO1          30-Oct-1984
    !               MODULES: LOADER and SYSBOOCMD
    !               Fix extraneous "Vector in use" message when booting from QVSS.
    !               Fix SHOW/HEX ASCII-parameter ACCVIO.
```

```
55) SYSLOAWS1 (new image)

    ! SYSLOAWS1.EXE
    !
    !   EC01    WHM0003           01-NOV-1984
    !           MODULE: OPDRVWS1 and SYSLOAVEC
    !           Add a second entry point to the input character interrupt
    !           routine to support being called from VCDRIVER.

56) SYSTARTUP (miscellaneous fix)

    ! SYSTARTUP.COM
    !
    !   EC002   DGB0108           22-DEC-1984
    !           Change /LENGTH qualifier to /PAGE qualifier for SET TERM.
    !
    !   EC001   DGB0101           16-NOV-1984
    !           Update the MicroVMS version number in SYS$MANAGER:SYSTARTUP.COM.
    !           This patch is included with VMS systems but applies only to
    !           MicroVMS.

57) TTDRIVER (new image)

    ! TTDRIVER.EXE
    !
    !   EC001   MIR0450
    !           Remove code added in V04-001 that is no longer necessary.
    !           Fix problem with disconnect where disconnection could
    !           happen when a read completion was waiting on the queue.
    !           This required a fork queue length check.
    !           Make terminal driver return F6 and F10 when line editing
    !           is disabled.

58) UVSTARTUP (edit text file)

    ! UVSTARTUP.COM
    !
    !   EC001   DGB0095           12-DEC-1984
    !
    !           Generalize the way we set up logical names for
    !           additional MicroVAX I bootstrap support.

59) VAXEMUL (new image)

    ! VAXEMUL.EXE
    !
    !   EC001   LJK0045           3-Jan-1985
    !           MODULE:           VAX$DECIMAL_ARITHMETIC
    !
    !           The result string in ADDP4 and SUBP4 must be probed for
    !           write access to ensure that the instruction does not modify
    !           memory before trying to restart.

60) VMBUVAX1 (new file)

    ! VMBUVAX1.COM
    !
    !   EC001   DGB0091           31-OCT-1984
    !           Provide a new command file for MicroVAX I customers so
    !           they can build a special VMB which allows them to boot
    !           from any MSCP disk on their system.
```

61) VMBUVAX1P (new file)

```
! VMBUVAX1P.EXE
!
!   ECOO1   DGBOO92          31-OCT-1984
!                Provide a new VMB for MicroVAX I customers which they can use
!                in conjunction with [SYSUPD]VMBUVAX1.COM to build a special
!                floppy diskette which they can use to boot their system from
!                any MSCP disk.  Note that this is a "VUF" file rather than
!                a "VUI" file because VMB is not a patchable image.
```

62) VMSINSTAL (edit text file)

```
! VMSINSTAL.COM
!
!   ECOO1   HWSOOO1          26-NOV-1984
!                Fix GENERATE_SDL_DEFINITIONS to correctly check for the
!                existence of SDLNPARSE.EXE.
```

63) VMSKITBLD (new file)

```
! VMSKITBLD.DAT
!
!   ECOO1   KDMOOO1          10-OCT-1984
!                Add SDLNPARSE.EXE to PROG_I option for MicroVMS.
!
!   ECOO2   DGBOO93          01-NOV-1984
!                Add VMBUVAX1.COM and VMBUVAX1P.EXE.
```

64) WELCOME (miscellaneous fix)

```
! WELCOME.TXT
!
!   ECOO1   DGBO1O2          16-NOV-1984
!                Update the MicroVMS version number in SYS$MANAGER:WELCOME.TXT.
!                This patch is included with VMS systems but applies only to
!                MicroVMS.
```

65) XDDRIVER (patch image)

```
!XDDRIVER.EXE
!
!   ECOO1   RNGOOO1          27-December-1984
!                Module:XDDRIVER
!                Fix problem with computing modulo value for calculating
!                modulo arithmetic for use with trib vector.
```

66) XQDRIVER (new image)

```
!  XQDRIVER.EXE
!
!   ECOO1   RNGOOO1          01-Oct-1984
!                MODULE: XQDRIVER
!                Fix additional buffer quota test.
!                Add support for new CPU.
!                Add some performance enhancements.
!                Fix loopback mode.
!                Allow for second DEQNA, by obtaining vector address
!                from IDB.
```

B OTS$MOVE3 and OTS$MOVE5 Routine Descriptions

During the final stages of producing the *VAX/VMS Run-Time Library Routines Reference Manual*, the descriptions of two new language-independent support routines were inadvertently omitted. These routines, OTS$MOVE3 and OTS$MOVE5, are documented in this appendix.

Information on these routines is included in the VAX/VMS system help files. To see it, type the following command:

```
$ HELP RTL OTS$ OTS$MOVE*
```

OTS$MOVE3 - Move Data Without Fill

OTS$MOVE3 moves up to 2^{31-1} (2,147,483,647) bytes from a specified source address to a specified destination address.

FORMAT OTS$MOVE3 *length ,source ,dest*

---

corresponding jsb entry point OTS$MOVE3_R5

ARGUMENTS *length*

type: **longword integer (signed)**
access: **read only**
mechanism: **by value**

Number of bytes of data to move. The **length** argument is a signed longword interger containing the number of bytes to be moved. Value may range from 0 through 2,147,483,647 bytes.

source

type: **byte (unsigned)**
access: **read only**
mechanism: **by reference, array reference**

Data to be moved by OTS$MOVE3. The **source** argument contains the address of an unsigned byte array that contains this data.

dest

type: **byte (unsigned)**
access: **write only**
mechanism: **by reference, array reference**

Item into which **source** will be moved. The **dest** argument is the address of an unsigned byte array into which OTS$MOVE3 writes the source data.

DESCRIPTION OTS$MOVE3 performs the same function as the VAX MOVC3 instruction except that **length** is a longword integer rather than a word integer. When called from the JSB entry point, the register outputs of OTS$MOVE3_R5 follow the same pattern as those of the MOVC3 instruction.

R0	0
R1	Address of one byte beyond the source string
R2	0
R3	Address of one byte beyond the destination string
R4	0
R5	0

For more information, see the description of the MOVC3 instruction in the *VAX–11 Architecture Reference Manual.* See also the routine LIB$MOVC3, which is a callable version of the MOVC3 instruction.

CONDITION VALUES RETURNED   None.

OTS$MOVE5Move Data With Fill

OTS$MOVE5 moves up to 2^{31-1} (2,147,483,647) bytes from a specified source address to a specified destination address, with separate source and destination lengths, and with fill. Overlap of the source and destination arrays does not affect the result.

FORMAT

OTS$MOVE5 *srclen ,source ,fill ,dstlen ,dest*

corresponding jsb entry point

OTS$MOVE5_R5

ARGUMENTS

srclen

type: **longword integer (signed)**
access: **read only**
mechanism: **by value**

Number of bytes of data to move. The **srclen** argument contains a signed longword integer that is this number. Value may range from 0 through 2,147,483,647.

source

type: **byte (unsigned)**
access: **read only**
mechanism: **by reference, array reference**

Data to be moved by OTS$MOVE5. The **source** argument contains the address of an unsigned byte array that contains this data.

fill

type: **byte (unsigned)**
access: **read only**
mechanism: **by value**

Character used to pad the source data if **scrclen** is less than **dstlen**. The **fill** argument contains the address of an unsigned byte that is this character.

dstlen

type: **longword integer (signed)**
access: **read only**
mechanism: **by value**

Size of the destination area in bytes. The **dstlen** argument is a signed longword integer containing this size. Value may range from 0 through 2,147,483,647 bytes.

dest

type: **byte (unsigned)**
access: **write only**
mechanism: **by reference, array reference**

Item into which **source** will be moved. The **dest** argument is the address of an unsigned byte array into which OTS$MOVE5 will write the source data.

DESCRIPTION

OTS$MOVE5 performs the same function as the VAX MOVC5 instruction except that the **srclen** and **dstlen** arguments are longword integers rather than word integers. When called from the JSB entry point, the register outputs of OTS$MOVE5_R5 follow the same pattern as those of the MOVC5 instruction.

R0	Number of unmoved bytes remaining in source string
R1	Address of one byte beyond the source string
R2	0
R3	Address of one byte beyond the destination string
R4	0
R5	0

For more information, see the description of the MOVC5 instruction in the *VAX-11 Architecture Reference Manual*. See also the routine LIB$MOVC5, which is a callable version of the MOVC5 instruction.

CONDITION VALUES RETURNED

None

Appendix C

Additional MicroVAX I Bootstrap Support

Requires the Common Utilities Option

The MicroVAX I bootstrap ROM can only boot a system disk attached to the first Mass Storage Control Protocol (MSCP) controller. If you want to boot a system disk on a successive controller, you must first create a special floppy diskette to use as a console floppy.

You must already have installed MicroVMS Version 4.1 on the fixed disk (RD51 or RD52) located inside the processor box as described in Section 1.1.2 of the *MicroVMS User's Manual*. After you have installed the update, halt the system and restart it following the procedure in Section 1.2.1 of the *MicroVMS User's Manual*.

C.1 Creating the Console Floppy Diskette

To create the console floppy, take the following steps:

1. Log into the SYSTEM account.

2. Invoke the command procedure to create the floppy disk, by entering the following command:
   ```
   $ @SYS$UPDATE:VMBUVAX1
   ```

This command procedure prompts you for the physical name of the disk from which you want to be able to boot (that is, your new system disk). For example, the physical device mnemonic for the second MSCP controller on the system would be DUBx (the controller letter in the device name is "B"). The last element in the physical device name is the unit for the disk drive.

If you have one fixed and one removable disk, DIGITAL recommends that you use the removable disk for your system disk. In this way, when you perform backups, you can remove the system disk and back up the fixed disk (containing user files) to a spare disk using the stand-alone backup procedure described in Section 1.6 of the *MicroVMS User's Manual*. If there are also user files on the system disk, you can back them up to floppy diskettes.

The following example sets a console floppy diskette that will boot a disk on a second MSCP controller as the system disk:

```
$ @SYS$UPDATE:VMBUVAX1
What system disk do you want to be able to boot from?  DUB0
```

Ignore the messages that are displayed on your terminal screen. The system will then prompt you for the name of the floppy drive on which the intermediate media (console floppy) will be mounted. The following example mounts the media in floppy drive 1 in response to the prompt from the command procedure.

```
Where will the intermediate RX50 media be mounted?  DUA1
```

You will then be prompted to insert a diskette into drive 1 and asked if you are ready. If there is a diskette in the first drive from a previous step, remove it. Take out a scratch diskette, label it "CONSOLE FLOPPY," and insert it into drive 1. After you have inserted the console floppy into drive 1 enter YES in response to the following prompt:

```
Are you ready?  YES
```

The system will inform you when it has completed building the console floppy with a message such as the following:

```
The media on DUA1 may now be used to boot device DUB0
```

You may want to write-protect the console floppy in order to prevent anyone from accidentally overwriting the contents. To write-protect a floppy diskette, cover the square notch in the upper right corner of the diskette. You may take the console floppy out of drive 1; however, you will be unable to boot the system either manually or automatically when the console floppy is not in the drive.

C.2 Placing MicroVMS on the New System Disk

Take the following steps to place MicroVMS on the disk on the second MSCP controller.

1. Load the target system disk—Load the target system disk and spin it up.

2. Mount the console floppy—You may wish to add the following command line to the end of SYS$MANAGER:SYSTARTUP.COM to mount the console floppy so that other users do not remove the floppy thinking that the drive is not in use.

   ```
   $ MOUNT/SYSTEM/FOREIGN $FLOPPY1 INTERMEDIATE
   ```

3. Notify users to log out—Make sure that nobody but you is logged on to the system (notifying any users that they must log off) and that no batch jobs are running (see Section 4.8 of the *MicroVMS User's Manual*).

4. Copy MicroVMS to the new system disk—Copy the contents of your current system disk using the following DCL commands (where *old-disk* is the name of the device which currently serves as your system disk and *new-disk* is the name of the device you want to use as your new system disk).

   CAUTION: The following set of commands will destroy the previous contents of the new disk.

```
$ INITIALIZE new-disk MICROVMS
$ MOUNT/FOREIGN new-disk
$ BACKUP/IMAGE/VERIFY/IGNORE=INTERLOCK old-disk:  new-disk:
```

   For example, if your current system disk is called DUA0 (the fixed RD51 or RD52 disk) and you want your new system disk to be DUB0, enter the BACKUP command from above as follows:

```
$ BACKUP/IMAGE/VERIFY/IGNORE=INTERLOCK DUA0:  DUB0:
```

   Ignore any error messages which report specific files not copied by BACKUP.

5. Dismount the new system disk—To complete the procedure, enter the following command to dismount the new system disk.

```
$ DISMOUNT/NOUNLOAD new-disk
```

C.3   Booting from the New System Disk

Take the following steps in order to start the system using the new system disk.

1. Check the console floppy—Make sure that the console floppy is still loaded in the floppy drive.

2. Halt the system—Press the HALT button on the computer panel to enter console mode; the console prompt ($>>>$) should appear on the console terminal (you must press the HALT button twice so that it pops back out again).

3. Start the system from the new system disk—Boot the system by entering the "B" command followed by the name of the drive containing the console floppy diskette. For example, if the console floppy is loaded in floppy drive 1, enter the following boot command:

```
>>>B DUA1
```

4. Continue with normal system startup—Continue with step 2 in Section 1.2.1 of the *MicroVMS User's Manual*.

C–4   Additional MicroVAX I Bootstrap Support
Booting from the New System Disk

When you are satisfied that your new system disk is working properly, delete all the MicroVMS files from the old system disk (for instructions on deleting directories, see Section 5.3.2 of the *MicroVMS User's Manual*). If you do not delete the MicroVMS files from your old system disk, when the system restarts, it might accidentally use the old disk as the system disk if the console floppy is not loaded.

READER'S COMMENTS

Note: This form is for document comments only. DIGITAL will use comments submitted on this form at the company's discretion. If you require a written reply and are eligible to receive one under Software Performance Report (SPR) service, submit your comments on an SPR form.

Did you find this manual understandable, usable, and well organized? Please make suggestions for improvement.

_____

_____

_____

_____

_____

_____

Did you find errors in this manual? If so, specify the error and the page number.

_____

_____

_____

_____

_____

_____

Please indicate the type of user/reader that you most nearly represent:

☐ Assembly language programmer
☐ Higher-level language programmer
☐ Occasional programmer (experienced)
☐ User with little programming experience
☐ Student programmer
☐ Other (please specify) _____

Name _____ Date _____

Organization _____
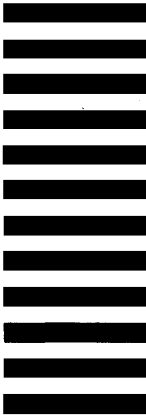
Street _____

City _____ State _____ Zip Code _____
or Country