VAX/VMS Release Notes Version 4.2

Order Number: AA-Z106A-TE

July 1985

The VAX/VMS Release Notes for Version 4.2 contain additions and restrictions to Version 4.2 of VAX/VMS. The information in these notes is not included elsewhere in the VAX/VMS Version 4.2 documentation kit.

Operating System and Version: VAX/VMS Version 4.2

Software Version: VAX/VMS Version 4.2

digital equipment corporation
maynard, massachusetts

First Printing, July 1985

The postpaid READER'S COMMENTS form on the last page of this document requests the user's critical evaluation to assist in preparing future documentation.

ZK2917

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. TEX is a registered trademark of the American Mathematical Society.

Contents

Preface

Before installing your VAX/VMS kit, read this document. The *VAX/VMS Release Notes Version 4.2* contain the following sections:

- New and Changed Features — This summary lists new features and major changes in Version 4.2 of VAX/VMS.
- Installation and Mandatory Update Notes — This chapter presents important installation and mandatory update information.
- Upgrading to Version 4.2 — This chapter presents instructions on how to upgrade your system to Version 4.2.
- Notes for the General User — This chapter presents additions and restrictions to Version 4.2 of VAX/VMS that are of interest to general users.
- Notes for the System Manager — This chapter presents additions and restrictions to Version 4.2 of VAX/VMS that are of interest to system managers (including security and network managers).
- Notes for the Application Programmer — This chapter presents additions and restrictions to Version 4.2 of VAX/VMS that are of interest to application programmers and others who use the Run-Time Library, utility, and system service routines.
- Notes for the System Programmer — This chapter presents additions and restrictions to Version 4.2 of VAX/VMS that are of interest to system programmers and device driver writers.
- Index — This index is specific to information documented in the VAX/VMS Release Notes for Version 4.2.

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.

1 New and Changed Features

The following is a list of the new features and major changes for the VAX/VMS Version 4.2 upgrade.

- **Networking** — The new features for networking include dynamic switching of asynchronous DDCMP lines and the ability to copy information about remote nodes from the node database of any accessible node. New to the Network Control Program (NCP) are the COPY KNOWN NODES command, parameters for dynamic asynchronous DDCMP connections, and a tracepoint for the X.25 trace module.
- **Run-Time Library** — A description of the new features for the Run-Time Library routines is located in the New and Changed Features Section of the *VAX/VMS Run-Time Library Routines Reference Manual* for Version 4.2.
- **Security** — New features are Access Control Lists (ACLs) on global sections and system logical name tables, new alarms enabled with the SET AUDIT command, and a new audit reduction facility (SECAUDIT.COM). Also, there are two new DCL commands to access and maintain the breakin database: SHOW INTRUSION and DELETE/INTRUSION_RECORD.
- **Symbolic Debugger** — New features include support of VAX Ada, screen-mode enhancements (support of large terminal screens, new instruction display, enhanced scrolling and source line display), exception lexical functions, aggregate watchpoints, and interface to the VAX Language Sensitive Editor. New debugger commands include EDIT, DISABLE AST, ENABLE AST, SHOW AST, SET MODE [NO]DYNAMIC, SET MODE [NO]LINE, SET MODE [NO]SCROLL, SET PROMPT, and SHOW EXIT_HANDLERS. Enhancements have also been made to the SHOW MODULE command.
- **System Dump Analyzer** — New commands are SHOW CLUSTER, SHOW CONNECTIONS, SHOW PORTS, and SHOW RSPID. Also, there is a new qualifier available with the VALIDATE QUEUE command: /SELF_RELATIVE.
- **System Services** — There are two new system services: Get Queue Information ($GETQUI) and Get Queue Information and Wait ($GETQUIW).
- **Text Processing** — The VAX Text Processing Utility (VAXTPU) is a new high-performance, programmable, text processing tool. VAXTPU is designed to aid application and system programmers in the development of text processing interfaces. VAXTPU supports two DIGITAL-supplied editing interfaces, the EDT emulator and EVE. DIGITAL Standard Runoff has added a new qualifier (/REVERSE_EMPHASIS) and a new option (LN03) for the /DEVICE qualifier.

2 Installation and Mandatory Update Notes

This chapter presents important installation and mandatory update information for VAX/VMS Version 4.2.

2.1 Problems with Standalone BACKUP

Standalone backup kits that are not on the actual VMS distribution media (such as the magnetic tape kit) may display the system date as one year and one day off. This is simply a cosmetic problem that will not affect the installation. Once the installation is completed, however, you should rebuild a standalone BACKUP kit from your newly installed system. Instructions on how to build a new kit may be found in the *Guide to VAX/VMS Software Installation.*

A more important concern with Standalone BACKUP involves a set of special circumstances where standalone BACKUP may request that the user insert the next application floppy or TU58 volume before Standalone BACKUP has finished using the current application floppy. This only occurs when you are booting console media on a VAX with a CI adapter and disks or tapes accessible through the CI (either HSC disks or tapes, or disks served from other VMS nodes through the MSCP server).

The solution to this problem is not to insert the requested volume immediately, but to wait until all activity on the device has stopped. For example, the red light on the TU58 will go out or the floppy will stop clicking. Wait at least ten seconds after all activity has stopped before changing volumes. Standalone BACKUP will print a message describing each remote disk which has been configured into the system and is accessible. Failure to wait a sufficient amount of time might result in some remote disks or tapes not being available for the backup.

2.2 Installing the Mandatory Update

After you have installed or upgraded to VAX/VMS Version 4.2, but before you have installed any VAX/VMS options, you must immediately install the additional mandatory update.

The mandatory update is labeled as follows:

- VAX/VMS V4.2 BINRX01 Mandatory Update
- VAX/VMS V4.2 16MT9 Mandatory Update
- VAX/VMS V4.2 TU58 Mandatory Update

Use the following procedure to install the mandatory update:

1. Log in to the System Manager's account (SYSTEM).
2. Apply the mandatory update to the drive.

3 Enter the following command, where *ddcu* is the name of the device on which you have mounted the update):

```
$ @SYS$UPDATE:VMSINSTAL VMSMUP042 device-name
```

The procedure prompts you for certain information (such as whether you have inserted the mandatory update and are ready to proceed). Upon completion, the procedure shuts down the system, after which you must reboot it.

Note: Included in this update is a patch to SYS.EXE. Do not delete the existing version of SYS.EXE; use the PURGE command to remove old versions of the files after you reboot the system.

The following patches are applied to the system during the mandatory update.

```
1) AUTHORIZE (patch image)
   ! AUTHORIZE.EXE
   !
   !   EC001   JRL0077         26-Jun-1985
   !           MODULE: UAFMAIN
   !           Finish final stage of conversion from explicit command
   !           prompting/parsing to implicit using CLI$DCL_PARSE
   !
   !   EC002   JRL0090         28-Jun-1985
   !           MODULE: UAFPARSE
   !           Eliminate problem with parsing of
   !           /<x>_RESTRICT
   !
   !   EC003   JRL0093         08-Jul-1985
   !           MODULE: UAFMAIN
   !           Restore file modification messages when AUTHORIZE is
   !           quit with the CTRL/Z key
   !

2) BASRTL (miscellaneous fix)
   !
   !   BASRTL.EXE
   !
   !           KC1079                    15-Jul-1985
   !           MODULE:  BAS$$UDF_WL
   !           Correct initialization of ISB$V_MAT_PRINT
   !

3) BASRTL (patch image)
   ! BASRTL.EXE
   !
   !   EC001   KC1079          15-Jul-1985
   !           MODULE: BAS$$UDF_WL
   !           Clear ISB$V_MAT_PRINT so MAT PRINT outputs the correct
   !           number of blank lines

4) CTDRIVER (patch image)
   ! CTDRIVER.EXE
   !
   !   EC001   PL              03-JUL-1985
   !           Module: CTDRIVER.MAR
   !           Fix setting of speed, crfill, and parity
   !           for non-VMS to VMS case
   !
   ! addr of CTDRIVER

5) DUDRIVER (patch image)
   ! DUDRIVER.EXE
   !
   !   EC002   ROW0473         07-JUL-1985
   !           MODULE: DUTUSUBS
   !           Enhance new unit check for shadowing
```

```
6) F11BXQP (patch image)
   ! F11BXQP.EXE
   !
   !  ECO01    LMP0331         03-Jul-1985
   !           MODULE: DEACCS
   !           Undo the change made in LMP0331 to enable the
   !           protection check on the write attributes call
   !

7) LIBDECOMP (edit text file)
   ! LIBDECOMP.COM
   !
   !   ECO01   BJT0012         03-Jul-1985
   !           Fix file type of SYS$LIBRARY:ERFLIB to be .TLB

8) NCP (patch image)
   !NCP.EXE
   !
   !
   !   ECO 01                  03-Jul-1895
   !           Remove the parsing that inserted an unnecessary
   !           space in front of all tracepoint names
   !

9) NETDRIVER (patch image)
   ! NETDRIVER.EXE
   !
   !   ECO 01  ADE0044         22-Mar-1985
   !           MODULE: NETDRVNSP.MAR
   !           Change value of NSP$V_SEQ_NAR to reflect NSP
   !           definition -- was 14, is now 12
   !
   !   ECO 02  PRB0371         03-JUL-1985
   !           MODULE: NETDRVNSP.MAR
   !           Fix bug resulting in incorrect PROBE of receive
   !           buffer in IOPOST which led to ACCVIO being reported
   !           in the IOSB.  IRP$W_BCNT is used for the probe, but
   !           was too large because some data segments had be
   !           pre-copied to the user buffer by NETDRIVER but not
   !           accounted for in IRP$W_BCNT.  Fix involves accumulating
   !           the number of bytes copied to user buffer by NETDRIVER
   !           in IRP$W_ABCNT and subtracting that count from
   !           IRP$W_BCNT prior to going to IOPOST. This correctly sets
   !           IRP$W_BCNT for the PROBE.
   !

10) NODRIVER (new image)
   ! NODRIVER.EXE
   !
   !           MMD0356         10-July-1985
   !           Fix to scheduling the special receive no buffer, so that
   !           it is not queued after the circuit has been shut off;
   !           also fix a branch in start_transmit which could cause the
   !           system to crash.
   !

11) SETPO (patch image)
   ! SETPO.EXE
   !
   !   ECO01   JRL0092         02-Jul-1985
   !           MODULE: SETPWD
   !           Make sure the text returned by SET_PASSWORD_GENERATE
   !           is upcased
   !
```

```
14) SMGRTL (miscellaneous fix)
    !
    !  SMGRTL.EXE
    !
    !  TS020                    15-Jul-1985
    !  Module:   SMGDEF.SDL
    !  Remove curly braces from comments for VAX PASCAL
    !

15) SPKITBLD (edit text file)
    ! SPKITBLD.COM
    !
    !   ECOnn   JJ00011         11-Jul-1985
    !           If using a TK50, don't use the /DENSITY qualifier on
    !           the BACKUP command

16) STARTUP (edit text file)
    ! STARTUP.COM
    !
    !   ECO01   BJT0014         14-Jul-1985
    !           Fix error in deleting symbol for tailored systems

17) SYS (patch image)
    ! SYS.EXE
    !
    !  ECO28    BJT0011         21-Jun-1985
    !           Set version number to "V4.2"
    !  ECO29    WMC0001         08-Jul-1985
    !           Fix image activation from sequential device

18) VAXCRTLG (new image)
    !
    ! VAXCRTLG.VUI              28-Jun-1985
    !
    ! VAXCRTLG.EXE was built with a missing universal symbol, namely
    ! CC$_GFLOAT.  This new version contains the same code and data as
    ! the original shareable image, but with the additional universal
    ! symbol.
    !
```

3 Upgrading to Version 4.2

This chapter tells you how to upgrade your VAX/VMS operating system from Version 4.0 or Version 4.1 to Version 4.2. If your operating system is not currently at one of these levels, you will have to update to Version 4.0 before you can upgrade to Version 4.2. Alternatively, you could perform a new installation of VAX/VMS Version 4.2 on a blank disk.

Note: You must install a mandatory update immediately after upgrading to VAX/VMS Version 4.2. See Chapter 2 of this manual for instructions.

A major part of the upgrade is automated and does not require that an operator be present throughout. For this reason, you should do the upgrade from a hard-copy device to have a record of events that occur during the upgrade.

Sections 3.1 and 3.2 list the steps you should take before you start the upgrade. The actual upgrade is described in Section 3.3. Sections 3.4 through 3.6 deal with post-upgrade information.

Here is a list of materials you need to do the upgrade:

- The Version 4.2 software distribution kit
- A blank disk for building the upgraded system
- A blank console volume (floppy diskette if you have a VAX–11/780, VAX–11/782, or VAX–11/785; RL02 if you have a VAX 8600; TU58 if you have any other VAX processor)

Note: Upgrades are not supported for VAX–11/730 users with dual-RL02 system configurations. See the installation booklet *Installing VAX/VMS on a Dual-RL02 System* for instructions.

You should read this entire installation procedure before beginning the upgrade to be sure that you are aware of all upgrade requirements.

Note: If you have changed the names of system directories in your current operating system, or deleted VAX/VMS files from them, the upgrade procedure may not work correctly. You will have to restore your operating system to a standard system before you can continue with the upgrade.

The upgrade procedure has been engineered and tested so that layered products should not have to be reinstalled. Although great effort has gone into this process, there is no guarantee that all layered products, either DIGITAL supported or third party provided, will not have to be reinstalled due to differences in product-specific installation procedures. For example, any product that creates directories that are synonyms for system directories (such as VAX–11 RSX) would have to be reinstalled. Also, any product that generates files from VMS libraries (such as STARLET) would have to be reinstalled.

You should also note the following before beginning the upgrade:

- As part of the upgrade, the paging, swapping, dump, and authorization files are purged back to one version.
- Everything in the [SYSERR] directory is deleted.
- All operator and accounting logs are deleted. If you want to keep any of the files, move them to a user directory before starting the upgrade.
- The upgrade cannot be applied to a tailored system. A tailored system must be installed on a separate, clean disk.

DIGITAL recommends that you back up your system disk before upgrading because Version 4.2 makes some files obsolete and these are subsequently deleted.

You must back up your console RL02 if you are upgrading a VAX 8600.

If the upgrade is interrupted for any reason, it can resume from the point at which the system was most recently booted. There are three times during the upgrade procedure when the system reboots.

If you want the upgrade to resume automatically, you must boot the system exactly as requested by the upgrade procedure after the planned system shutdown in Phase 1.

To allow the upgrade procedure to resume at any point, the procedure places the new Version 4.2 system files in an alternate disk directory root (SYSF) from your current system's files, which are in SYS0. This ensures that you have at least one set of bootable system files at all times.

3.1 Applying Upgrades to VAXcluster Systems

Section 3.1 pertains only to VAXcluster environments. If you are not installing Version 4.2 in a VAXcluster environment, proceed directly to Section 3.2.

3.1.1 General Considerations

First, all members of a VAXcluster must run the same version (major and minor) of VAX/VMS. Therefore, VAXcluster sites must be prepared to upgrade all VAX systems in a cluster at the same time. The reason for requiring the same version of VAX/VMS on all VAX systems is that the high degree of sharing achieved between systems in a VAXcluster is the result of coordination at many levels of VAX/VMS. This level of coordination cannot, in general, be achieved across major or minor releases of VAX/VMS.

However, it is also true that, when possible, DIGITAL will endeavor to allow adjacent releases of VAX/VMS to coexist in a VAXcluster for the purpose of incrementally updating the various systems in the VAXcluster. DIGITAL recommends that this "mixed version" operation of a VAXcluster be used only to allow an incremental upgrade of new versions of VAX/VMS.

Note: The term "common system root" refers to a directory structure residing on a common system disk containing the system files which are shared by several processors in a cluster environment. The term "private system root" refers to a directory structure residing in either a private, local system disk or a shared system disk where the system files are not shared.

The term "system root" is a generic term referring to either common or private system roots.

When upgrading a common system root, you need to perform only one complete upgrade from one of the nodes sharing the common system root. However, all other nodes may require modifications to the console boot command files as well as manually invoking AUTOGEN to update the system configuration parameters. Alternatively, you may use the MAKEROOT command procedure to create new alternate roots for these nodes (see Section 3.6).

3.1.2 Specific VAX/VMS Version 4.2 Considerations

VAX/VMS Versions 4.0, 4.1 and 4.2 may be intermixed in VAXcluster configurations for the purposes of incrementally applying the upgrade from Version 4.0 or 4.1 to Version 4.2. Please note the following:

- All systems booted from a common system root run the same version of VAX/VMS.
- DIGITAL recommends that generic batch queues direct jobs to Version 4.0, Version 4.1, or Version 4.2 systems, or only one queue. The purpose of this recommendation is to avoid inconsistent and possibly confusing behavior during the upgrade and checkout of VAX/VMS Version 4.0.
- When a VAX/VMS Version 4.2 system boots in the presence of a VAX/VMS Version 4.1 system, the following informational message will be printed on the system console.

  ```
  %CSP-I-DIFSWVER, Different versions of VAX/VMS exist in cluster
  ```

- DIGITAL recommends that the upgrade from Version 4.0 or Version 4.1 to Version 4.2 be completed on all members of the cluster as quickly as practical.
- Note that during the upgrade you need to boot a system stand alone that is normally in a cluster. In some cases, depending on your site's start up procedures, this system may hang while waiting for other members of the cluster.

  To resolve this situation, set the SYSGEN parameter STARTUP_P1 to be "MIN" and boot your system. This will provide a minimal startup and not invoke site-specific startup procedures.

  Once the system is running, invoke SYSGEN and specify the following so all available devices are configured.

  ```
  SYSGEN>AUTOCONFIGURE ALL
  ```

3.1.3 Upgrading a VAXcluster Environment: Rolling Upgrade

This section describes the technique used to maintain partial system availability during an upgrade in which old and new versions of VAX/VMS can simultaneously exist in the same VAXcluster. Additional pertinent information regarding clusters is available in Chapter 5 of the *Guide to VAXclusters*.

Note: VAX/VMS Version 4.2 supports the running of mixed versions of VMS during an upgrade procedure only.

This procedure is applicable to VAXclusters that have multiple system roots. It is not applicable when all systems boot from a single common system root.

This upgrade is applied to all system roots, one system root at a time, without shutting down the entire cluster. With a common system root, only one system, the system running the upgrade, can be running while the upgrade is applied.

To upgrade your system, do the following in the specified order:

1. Check the votes and make adjustments to maintain the proper quorum that will allow your cluster to continue operating throughout this process (the procedure is described in full detail in Chapter 5 of the *Guide to VAXclusters*).
2. Complete all the steps in Section 3.2 of these release notes.
3. Shut down all systems on the common system root except the one to be upgraded. Do this by following the standard shutdown procedure and adjusting the votes and quorum to allow the one necessary running node.
4. Upgrade the single running node according to Section 3.3 of these release notes.
5. If the system root is a common system root, reboot the other systems on the system root. This will allow all systems on the common system root to run the upgraded version. Before rebooting, the cluster is running with mixed versions of VAX/VMS.
6. Repeat all tasks in this section for each system root until all roots are running the upgraded version.

3.1.4 Upgrading a VAXcluster Environment: Concurrent Upgrade

This section describes the upgrading technique to be used when the VAXcluster cannot run more than one version simultaneously.

A concurrent upgrade is performed by bringing down the entire cluster and applying the upgraded version to each private and common system root, one at a time. Then the cluster is brought back up to run with the upgraded version.

Note that applying the upgrade to a single node on a common system root will in turn upgrade all connecting nodes. All private disks need to be upgraded individually.

To upgrade your system, do the following:

1. Make note of the votes and quorum, so upon completion of the upgrade they can be reset to their original values (the procedure is described in full detail in Chapter 5 of the *Guide to VAXclusters*).

2. Set all votes and quorum to 1 and check the SYSGEN parameters SCSNODE, ALLOCLASS, STARTUP_P1, and VAXCLUSTER. VAXCLUSTER and ALLOCLASS must be 0, SCSNODE must be blank, and STARTUP_P1 must be "MIN". You can check and (optionally) set these values (see Section 3.2.6).
3. Shut down the entire cluster using the standard shutdown procedure.
4. Reboot a single system and install the upgraded version on the system according to Sections 3.2 and 3.3. This will apply the upgrade to the system root from which the system is booted.
5. Restore the votes and quorum to their original settings and reset any SYSGEN parameters (such as SCSNODE, ALLOCLASS or VAXCLUSTER) that may have been modified.
6. Shut down the system according to standard shutdown procedures.
7. Repeat steps 4 through 6 until the upgrade has been applied to each system root.
8. Bring up the entire cluster according to your normal operating procedures. The entire cluster will now be running the upgraded version of VAX/VMS.

3.2 Preparing to Upgrade the System

Use the following procedure to prepare for the upgrade. Note that the system disk and the distribution volume cannot be moved from one device to another during the upgrade.

1. Log in to the system manager's account, SYSTEM, on the console terminal.
2. Use stand alone BACKUP to back up your current system disk to a scratch disk. (A step-by-step procedure for backing up your system disk is given in Chapter 4 of the *Guide to VAX/VMS Software Installation.*)
3. If your current system disk can be removed, store it in a safe place so that it is readily available if needed. Use the backup copy to upgrade your system.

Note: If you have a VAX 8600, use the command procedure SYS$UPDATE:CONSCOPY to create an RL02 backup copy of your console media.

4. Boot the backup copy of your system disk (or your current system disk, if it cannot be removed).
5. When the backup copy of your system disk boots, log in under the system manager's account, SYSTEM. (You may want to consider having a second terminal logged in for doing support tasks during the upgrade.)
6. Before using the upgrade procedure, the SYSGEN parameters SCSNODE, ALLOCLASS, STARTUP_P1, and VAXCLUSTER must be properly set. VAXCLUSTER and ALLOCLASS must be 0, SCSNODE must be blank, and STARTUP_P1 must be "MIN". You can check and (optionally) set these values as follows:

```
$ RUN SYS$SYSTEM:SYSGEN
SYSGEN>  USE CURRENT
SYSGEN>  SHOW VAXCLUSTER
Parameter Name              Current   Default   Minimum   Maximum Unit  Dynamic
--------------              -------   -------   -------   ------- ----  -------
VAXCLUSTER                        1         1         0         2 Coded-value
SYSGEN>  SET VAXCLUSTER 0
SYSGEN>  SHOW SCSNODE
Parameter Name              Current   Default   Minimum   Maximum Unit  Dynamic
--------------              -------   -------   -------   ------- ----  -------
SCSNODE                  "NODE    "    "    "    "    "    "ZZZZ" Ascii
SYSGEN>  SET SCSNODE ""
SYSGEN>  SHOW STARTUP_P1
Parameter Name              Current   Default   Minimum   Maximum Unit  Dynamic
--------------              -------   -------   -------   ------- ----  -------
STARTUP_P1               "         "    "    "    "    "    "ZZZZ" Ascii
SYSGEN>  SET STARTUP_P1 "MIN"
SYSGEN>  SHOW ALLOCLASS
Parameter Name              Current   Default   Minimum   Maximum Unit  Dynamic
--------------              -------   -------   -------   ------- ----  -------
ALLOCLASS                         1         0         0       255 Pure-number
SYSGEN>  SET ALLOCLASS 0
SYSGEN>  WRITE CURRENT
SYSGEN>  EXIT
```

If it is necessary to modify any of the above parameters, you **must** reboot your system before applying the upgrade to your system.

Note: This procedure is part of an effort to isolate the disk being upgraded. The system being upgraded must not mount disks in use by another system, nor may any other system mount the disks in use by this system. Should either situation occur, you may irretrievably corrupt the data on the disks.

7 To upgrade your system, you must have a minimum of 16,000 free blocks on the system disk you intend to upgrade. If you do not have this space, delete as many user files as necessary to get it. You can confirm the free block count with the following command:

```
$ SHOW DEVICES/FULL ddcu:
```

Here *ddcu* represents the physical device name of the drive that contains the backup copy of your system disk.

8 Make sure the restart switch on your processor control panel is set to automatic restart. The upgrade will continue automatically after shutdown(s) as long as this switch is positioned as shown in the following table.

VAX System	Restart Switch	Required Setting
VAX–11/725	AUTO/RESTART BOOT	ON
VAX–11/730	AUTO/RESTART BOOT	ON
VAX–11/750	POWER ON ACTION	(RESTART)
VAX–11/780	AUTO/RESTART	ON
VAX–11/785	AUTO/RESTART	ON
VAX–8600	RESTART/BOOT	RESTART/BOOT

9 Prevent users from logging into the system by typing the following command:

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

10 If you are running DECnet–VAX, shut down the network. First, enter the following command:

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

When the NCP prompt is displayed, type the following:

```
NCP>SET EXECUTOR STATE OFF
```

11 Press CTRL/Z to return to DCL command level.

12 You should stop all queues to avoid locked-file errors during the upgrade. Note that queues are initialized during the upgrade, and any jobs pending will be lost. Check to see if there are any jobs in the queues with the following command:

```
$ SHOW QUEUE/DEVICE/BATCH/FULL/ALL
```

If no queues are set up, an appropriate message will be displayed and you may bypass this step.

Use a DCL command of the following form to stop each queue:

```
$ STOP/QUEUE/NEXT queue_name
```

3.3 Performing the Upgrade

At this point, you begin the actual upgrade procedure.

1 Place the VAX/VMS Version 4.2 distribution volume on the appropriate drive. If you are using a tape drive, put the drive on line; if you are using a disk drive, spin the drive up.

2 Invoke the VMSINSTAL command procedure as follows.

```
$ SET DEFAULT SYS$UPDATE:
$ @VMSINSTAL
```

VMSINSTAL responds by announcing itself and asking if you have a satisfactory backup copy of your current system:

```
        VMS Software Product Installation Procedure
It is date at time.
Enter a question mark (?) at any time for help.
*Are you satisfied with the backup of your system disk [YES]?
```

3 If you have already backed up your old system disk, press the RETURN key to continue with the upgrade.

If you have a VAX 8600 and you have not backed up your console RL02, answer *NO* and perform the backup as described in Section 3.2.

If you have not backed up your old system disk, you may do so now by entering *NO*. VMSINSTAL will return you to DCL level so that you can do the backup. When the backup is completed, invoke VMSINSTAL to restart the upgrade.

4 Next, VMSINSTAL requests the name of the drive holding the distribution volume. Enter the device name in the *ddcu:* format.

For example, if the distribution volume is a magnetic tape mounted in a TS11 that is the first device on the UNIBUS, enter *MSA0:*. See Chapter 4 of the *Guide to VAX/VMS Software Installation* if you need more information on device names.

The upgrade procedure may abort if the device name format appears incorrect. If you specify a nonexistent device or a device that is not connected, an "invalid device" error message will be displayed. If this happens, use the DCL command SHOW DEVICES to verify device status.

5 When VMSINSTAL prompts you for the name of the product you wish to install, respond by entering VMS.

6 When your distribution volume is ready for mounting, enter *Y*.

VMSINSTAL responds with the following messages:

```
%MOUNT-I-MOUNTED, VMS042       mounted on _ddcu:
The following products will be installed:
  VMS    V4.2
       Beginning installation of VMS 4.2 at <date hh:mm>
%VMSINSTAL-I-RESTORE, Restoring product save set A...
                        VAX/VMS V4.2 Upgrade Procedure
                     .
                     .
                     .
                     .
```

If you are using a tape distribution volume, there will be a pause of several minutes between the first and second lines.

7 Next, the procedure describes the upgrade and discusses the automatic restart capability.

8 This is followed by a set of messages that describe cautions and requirements related to doing the upgrade:

- The procedure states that the kit *must* be stored in directory [000000]. The kit is stored in [000000] by default.
- The procedure cautions you about preserving files that may be deleted by the upgrade.
- Next, the procedure notes the importance of RESTAR.CMD matching your memory interleaving.
- The procedure describes how AUTOGEN may attempt to re-size your pagefile and swapfile if you do not manually override it.
- If you have DECnet–VAX installed, you must have the Version 4.0 DECnet–VAX license which is packaged in a separate distribution kit. No Version 3.0 DECnet–VAX license will work with a Version 4.0 or Version 4.2 system.
- If you are upgrading a VAX–11/785, you are told that the default bootstrap command procedure must be set to boot the system device *before* you start the upgrade.
- If you are upgrading a VAX–8600, you are told that you should have made a backup copy of your console RL02.

If you indicate that you want to interrupt the upgrade to comply with any of these conditions, the procedure returns you to DCL command level and the upgrade terminates. If you do this, you must reinvoke VMSINSTAL when ready to resume the upgrade.

9 When the upgrade resumes, the next step is determined by your system configuration.

Note: If your system includes either a CI780 or CI750, the upgrade gives you the option of creating a common disk for a VAXcluster.

3.3.1 Upgrade Phase 1

The upgrade is presented in five phases. Two descriptions are provided for Phase 1: one for users upgrading systems other than VAX–11/750 and one for users upgrading VAX–11/750 systems. The final four phases are common to all VAX systems.

You should now proceed to Phase 1 of the upgrade. If you are upgrading a VAX–11/750, skip the next section and go to Section 3.3.1.2.

3.3.1.1 Upgrade Phase 1 for VAX–11/725, VAX–11/730, VAX–11/780, VAX–11/785 and VAX–8600 Systems

This section describes the first phase of the upgrade for users who are upgrading VAX–11/725, VAX–11/730, VAX–11/780, VAX–11/785 and VAX–8600 systems.

1 To ensure system security, the upgrade procedure requires you to change the passwords for the SYSTEM, SYSTEST, and FIELD accounts before continuing. The first time you set these passwords, the system requires a password of six or more characters.

2 The upgrade procedure now turns off the disk quotas on the system disk and removes directory entries that point to nonexistent files.

3 Your upgraded system will be built in system root SYSF.

At this point the upgrade procedure will prompt you through the building of a new console volume that will let you boot the new system from SYSF. Specifically, the procedure will change the default bootstrap command procedure (DEFBOO.CMD) on the new volume to boot from SYSF.

Note: During this section of the installation, all command procedures used for upgrading a VAX 8600 will use file extension .COM in place of .CMD.

4 The procedure prompts you to insert your current console volume in the console drive. Your current console volume (the procedure calls it your "old" console volume) is used as a base to build the new console volume but it is *not* altered by the procedure.

5 When you indicate that you are ready to continue, the procedure copies your old console volume to a disk directory.

6 Now the procedure prompts you to set DEFBOO.CMD to boot the backup copy of your system disk (assuming you have not done so previously). Using the form dduBOO.CMD, enter the name of the boot file you want to copy to DEFBOO.CMD. For example, if the system disk is in DBA1, you enter DB1BOO.CMD. If the system disk is controlled by either an HSC or a UDA, the revised version of CIBOO.CMD or DUABOO.CMD that you built when you first installed VAX/VMS on the HSC- or the UDA-controlled system disk must be used. The selected command procedure must be able to boot the system disk from the drive where you have placed it without any operator intervention (for example, registers R0 through R5 are correctly initialized by the command procedure).

Do not specify a conversational boot command file. The upgrade kit is set with parameters that will boot any system. The procedure does build a conversational boot file (UPGGEN.CMD) that boots from SYSF, but you should only use this file for situations prescribed by the procedure.

7 When this is done, put your old console volume away for safe keeping and insert a blank volume in the console drive. Do not remove this volume for the rest of the upgrade.

Note: If you have a VAX 8600, your old console RL02 is used during this upgrade. A new one is not created; therefore, you should have a backup copy of your console RL02.

Note: If the new console volume is a TU58, be sure the RECORD switch is set to permit writing; that is, the switch should be slid in the direction of the arrow inscribed on it.

8 Now the procedure gives you the opportunity to have the Bad Block Locator Utility (BAD) check the new console volume before it is initialized. BAD checks the new console volume for any defective blocks, and DIGITAL recommends that you run it. If you choose to run BAD, allow an additional 30 minutes if you are using a TU58 volume. If you are using a floppy volume, an additional 15 minutes is required.

Note: The Bad Block Locator Utility is not used for upgrades on the VAX 8600.

Details on running BAD can be found under the ANALYZE/MEDIA command in the *VAX/VMS DCL Dictionary* or under the description of BAD in the *VAX/VMS Utilities Reference Volume.*

9 Finally, the procedure initializes the new console volume but it will not copy the files to it from the disk until just before the first reboot.

10 The upgrade procedure now does a directory cleanup and removes installed images.

11 Next, the procedure builds the directory tree in SYSF and and deletes all operating system files not needed to complete the upgrade.

12 Now the upgrade procedure restores the Version 4.2 *required* save set to the backup copy of your system disk and verifies that the save set files have been restored correctly.

13 When the verification pass is complete, the upgrade procedure purges the paging, swapping, dump, and authorization files and puts the most current version in the new directory tree.

14 After using AUTOGEN to save your old SYSGEN parameters and preparing the startup command file for Phase 2, the upgrade procedure copies the console files to the new volume and builds its boot block in preparation for the first reboot.

15 Now the procedure tells you it will shut down to reboot the partially installed Version 4.2 system. It also tells you that the upgrade continues automatically after the reboot but that if if you must restart the upgrade manually, simply enter the letter B.

16 Now the procedure cautions you not to move the distribution volume or the system volume and not to change the SYSGEN parameters for the remainder of the upgrade.

17 Next, the procedure tells you how to handle a shutdown failure or a reboot failure.

- On a VAX–11/730 system, the microcode may be reloaded. If the system fails to boot correctly, power the system off and then on again so that the console will reload the microcode from the newly created TU58.
- If the system fails to boot in a CI780 environment because of insufficient nonpaged dynamic memory, use the conversational boot command procedure UPGGEN.CMD to increase the NPAGEDYN system parameter.

Note: If you need to increase this parameter, submit a Software Performance Report (SPR) that describes your complete hardware configuration and what you did to get the system to boot.

To increase this parameter, use the following sequence of console commands:

```
>>> @UPGGEN.CMD
```

This will put you in SYSBOOT where you should enter the following command:

```
SYSBOOT> SET NPAGEDYN 120000
```

Then use this command to return to the upgrade procedure:

```
SYSBOOT> CONTINUE
```

When the system reboots, it identifies itself and announces the beginning of Phase 2.

```
VAX/VMS Version 4.2 <date hh:mm>
```

During the shutdown, the following error message will appear:

```
%SHUTDOWN-I-ISTOPQUEMAN, the queue manager will now be stopped.
%SYSTEM-F-DEVOFFLINE, device is not in configuration or not available.
```

This error is caused by known problems and may be ignored during the upgrade procedure.

Skip to Section 3.3.2, which describes Phase 2.

3.3.1.2 Upgrade Phase 1 for VAX–11/750

This section describes the first phase of the upgrade for users who are upgrading VAX–11/750 systems.

1 To ensure system security, the upgrade procedure requires you to change the passwords for the SYSTEM, SYSTEST, and FIELD accounts before continuing. The first time you set these passwords, the system requires a password of six or more characters.

2 The upgrade procedure now turns off the disk quotas on the system disk and removes directory entries that point to nonexistent files.

You are now asked if you want to boot using the TU58 rather than directly from the disk. If you have a CI750, you *must* answer YES to this question. DIGITAL recommends that you answer YES in any case to ensure you have a console TU58 with the latest versions of the VMB and BOOT58 programs.

If you choose to boot directly from the system disk, skip the next step and go to step 4.

3 Your upgraded system will be built in system root SYSF so that your current system in SYS0 is available if needed. At this point, the procedure will guide you through the building of a new console TU58 if you previously indicated that you would be booting from the TU58. Specifically, the procedure will change the default bootstrap command procedure (DEFBOO.CMD) on the new volume to boot from SYSF.

a Position the BOOT DEVICE switch on your processor control panel to position A and leave it in position A for the rest of the upgrade.

b The procedure prompts you to insert your current console TU58 in the console drive. Your current console volume (the procedure calls it your "old" console volume) is used as a base to build the new console volume, but it is *not* altered by the procedure.

c When you indicate you are ready to continue, the procedure copies your old console volume to a disk directory in Files–11 format.

d Now the procedure prompts you to set DEFBOO.CMD to boot the backup copy of your system disk (assuming you have not done so previously). Using the form dduBOO.CMD, enter the name of the boot file you want to copy to DEFBOO.CMD. For example, if the system disk is in DBA1, you enter DB1BOO.CMD. If the system disk is controlled by either an HSC or a UDA, the revised version of CIBOO.CMD or DUABOO.CMD that you built when you first installed VAX/VMS on the HSC- or the UDA-controlled system disk must be used. The selected command procedure must be able to boot the system disk from the drive where you have placed it without any operator intervention (for example, registers R0 through R5 are correctly initialized by the command procedure).

Do not specify a conversational boot command file. The upgrade kit is set with parameters that will boot any system. The procedure does build a conversational boot file (UPGGEN.CMD) that boots from SYSF, but you should only use this file for situations prescribed by the procedure.

e When this is done, put your old console volume away for safe keeping and insert a blank volume in the console drive. Do not remove this volume for the rest of the upgrade.

Note: Be sure the RECORD switch on the new console TU58 is set to permit writing; that is, the switch should be slid in the direction of the arrow inscribed on it.

f Now the procedure gives you the opportunity to have the Bad Block Locator Utility (BAD) check the new console volume before it is initialized. BAD checks the new console volume for any defective blocks, and DIGITAL recommends that you run it. If you choose to run BAD, allow an additional 30 minutes.

Details on running BAD can be found under the ANALYZE/MEDIA command in the *VAX/VMS DCL Dictionary* or under the BAD utility in the *VAX/VMS Utilities Reference Volume*.

g Finally, the procedure initializes the new console volume but it will not copy the files to it from the disk until just before the first reboot.

Note: Unless you are using RL02 distribution media, you should not have to intervene in the upgrade from this point on.

h The upgrade procedure now cleans up its directories and removes installed images.

i Next, the procedure builds the directory tree in SYSF and deletes all operating system files not needed to complete the upgrade.

j Now the upgrade procedure restores the Version 4.2 *required* save set to the backup copy of your system disk and verifies that the save set files have been restored correctly.

k When the verification pass is complete, the upgrade procedure purges the paging, swapping, dump, and authorization files and puts the most current version in the new directory tree.

l After using AUTOGEN to save your old SYSGEN parameters and preparing the startup command file for Phase 2, the upgrade procedure copies the console files to the new volume and builds a boot block on it in preparation for the first reboot.

m Now the procedure tells you it will shut down to reboot the partially installed Version 4.2 system. It also tells you that the upgrade continues automatically after the reboot and how to boot the new system if you must restart the upgrade manually,

- If you are booting from the console TU58, you will reboot the system from the BOOT58 command level. To invoke BOOT58, press CTRL/P to put the system in console mode; then enter the following command:

  ```
  >>> B/800 DDA0
  ```

  When you get the BOOT58 prompt, make the following entry:

  ```
  BOOT58> B
  ```

- If you are booting from the disk, you will have to use CTRL/P to get into console mode and then reboot with this command:

  ```
  >>> B/F0000000 ddcu
  ```

n Next the procedure tells you how to handle a reboot failure.

If the system fails to boot in a CI750 environment because of insufficient nonpaged dynamic memory, use the conversational boot command procedure UPGGEN.CMD to increase the NPAGEDYN system parameter.

Note: If you need to increase this parameter, please submit a Software Performance Report (SPR) that describes your complete hardware configuration and what you did to get the system to boot.

o To invoke UPGGEN.CMD when booting from a disk, use CTRL/P to get into console mode, then enter the following command:

```
>>> B/F0000001 ddcu
```

If you are booting from a TU58, invoke BOOT58 from console mode with this command:

```
>>> B/800 DDA0
```

When you get the BOOT58 prompt, enter this command:

```
BOOT58> @UPGGEN.CMD
```

p At the SYSBOOT prompt, enter the following command:

```
SYSBOOT> SET NPAGEDYN 120000
```

q Then use this command to return to the upgrade procedure:

```
SYSBOOT> CONTINUE
```

r During the shutdown, the following error message will appear:

```
% SHUTDOWN_I_STOPQUEMAN, the queue manager will now be stopped.
% SYSTEM_F_DEVOFFLINE, device is not in configuration or not available.
```

This error is caused by known problems and may be ignored during the upgrade procedure.

Skip to Section 3.3.2, which describes Phase 2.

When the system reboots, it identifies itself and announces the beginning of Phase 2.

```
VAX/VMS Version 4.2 <date hh:mm>
```

This completes Phase 1 of the upgrade. The rest of the upgrade does not require that an operator be present; however, if you are booting from the TU58, you will have to manually reboot each time the system shuts down (once in Phase 4 and once in Phase 5). If you are booting from the disk, reboots are automatic and do not require your intervention.

3.3.2 Upgrade Phase 2

The rest of the VAX/VMS Version 4.2 files in the LIBRARY and OPTIONAL save sets are restored from the distribution kit. This step varies, depending on your configuration.

- In a standard VAX/VMS configuration with either RK07, RA60, or magnetic tape distribution media, the LIBRARY save set is restored, and then the OPTIONAL save set is restored.
- In a standard VAX/VMS configuration with RL02 distribution media, the OPTIONAL save set is restored first. Then the procedure prompts you to remove the first volume of the distribution kit and to insert the second volume. When you comply, the LIBRARY save set is restored.

3.3.3 Upgrade Phase 3

Phase 3 of the upgrade merges the VAX/VMS-distributed files that are commonly edited by system managers with new VAX/VMS files. Ignore any "file not found" error messages that appear at this time.

The files HELPLIB.HLB, DCLTABLES.EXE, STARLET.OLB, and IMAGELIB.OLB are modified in an attempt to preserve layered product installations. Modules from the original files are added to the new copies.

Next, the upgrade procedure merges all the miscellaneous user files that exist in the old system directories into a new set of system directories, temporarily called [SYSF.SYSEXE], [SYSF.SYSMGR], [SYSF.SYSLIB], and so on. The amount of time this takes depends on the number of user files.

3.3.4 Upgrade Phase 4

During Phase 4 of the upgrade, the new site-specific console volume is modified to allow reboot of the complete VAX/VMS Version 4.2 system. Be sure the new site-specific console volume created in Phase 1 is in the console drive (CSA1: for VAX–11/780, VAX–11/785, VAX 8600 (original RL02), and VAX–11/750 systems; CSA2: for VAX–11/730 and VAX–11/725 systems).

When the modification is completed, you receive the following message:

```
System shutting down to boot the complete Version 4.2 system.
```

Leave the newly created site-specific console volume in the console drive. The system disk must also remain where it is in order to proceed to the next phase of the upgrade.

The system will attempt an automatic reboot after the shutdown.

Note: If you are upgrading a VAX–11/750 and are booting from the console TU58, you will have to manually reboot by entering the letter B when you get the BOOT58> prompt.

If the system fails to boot at this point because of insufficient nonpaged dynamic memory, you must use a standard conversational bootstrap to increase the NPAGEDYN (nonpaged dynamic memory) system parameter.

Note: If you need to increase this parameter, please submit a Sofware Performance Report (SPR) that describes your complete hardware configuration and what you did to get the system to boot.

If you need to invoke a conversational bootstrap at this point in the upgrade, invoke the conversational boot command procedure for your system disk. For example, if the system disk is on the first MASSBUS device, invoke DB0GEN, if the system is on the first UDA device, DU0GEN, and so forth.

When the boot is completed, the following message appears:

```
VAX/VMS Version 4.2 <date hh:mm>
```

3.3.5 Upgrade Phase 5

Phase 5 of the upgrade procedure creates a new site-specific SYSGEN parameter file, AUTOGEN.PAR, that combines the default values for Version 4.2 and the site-specific values you were using for your Version 4.0 or 4.1 system.

The command procedure cleans up several directories, announces the upgrade to Version 4.2 is complete, and makes several suggestions:

```
After the upgrade finishes, there are several things that you may wish
to do:
- DECOMPRESS THE SYSTEM LIBRARIES - For space considerations, many of the system
libraries are shipped in a data compressed format.  If you have enough disk
space, you may decompress them for faster access. Use SYS$UPDATE:LIBDECOMP.COM
to data expand the libraries.  If you choose not to decompress these libraries
there will be a negative impact on the performance of the HELP and LINK
commands.
- EDIT SYSTEM SPECIFIC FILES - The system specific files in SYS$MANAGER,
SYSTARTUP.COM, SYSHUTDWN.COM,  and SYCONFIG.COM have all been
superceded by blank files.  Your copies of these files still exist in
SYS$MANAGER as the next lower version number.
```

If the upgrade produced a common disk, the following message will be printed.

```
- SYSTEM SPECIFIC DIRECTORIES - The upgrade has produced a new common system
disk.  If there exist system roots other than the one that has just been
upgraded, it will be necessary to update the information contained in the
other system specific roots.  For example, you may want to run AUTOGEN or to
reconfigure your network database files. If you wish to recreate the other
system roots from scratch, use the SYS$MANAGER:MAKEROOT.COM procedure to
remove and recreate the specified systems roots.  Then run AUTOGEN and
reconfigure your network.

Note that new, blank copies of SYSTARTUP.COM, SYSHUTDWN.COM, and
SYCONFIG.COM are present in both SYS$SPECIFIC:[SYSMGR] and SYS$MANAGER:.

- Purge unnecessary files - There are a number of files that may be purged
after the upgrade is complete to free up space on your system disk.  First,
you may purge the lower-numbered versions of the following files from the
previous version of the operating system:

        [SYSEXE]SHUTDOWN.COM
        [SYSEXE]STARTUP.COM
        [SYSHLP]HELPLIB.HLB
        [SYSLIB]DCLTABLES.EXE
        [SYSLIB]ENCRYPSHR.EXE
        [SYSLIB]IMAGELIB.OLB
        [SYSLIB]LBRSHR.EXE
        [SYSLIB]*RTL*.EXE
        [SYSLIB]STARLET.OLB
        [SYSMGR]RTTLOAD.COM
        [SYSMGR]STARTNET.COM

Second, you may purge the lower-numbered versions of the following
files from the new version of the operating system.  These files may be
purged since they are the same as those shipped in the last version of
the operating system.

- Delete SYS$SYSTEM:STARTUP.UP5 and UPGRADE.KIT - These files are left
by the upgrade should this phase fail for some reason.  You may delete
them when the upgrade has completed.

If this system is to be part of a VAXcluster, you will need to invoke
AUTOGEN with this command:

        @SYS$UPDATE:AUTOGEN SAVPARAMS REBOOT

- APPLY THE MANDATORY UPDATE - You must apply the mandatory update as soon as
possible.   You will not be able to install any layered products or options
until after the mandatory update has been applied.
```

Then the system shuts down and automatically reboots.

Note: If you are upgrading a VAX-11/750 and are booting from the console TU58, you will have to manually reboot by entering the letter B when you get the BOOT58> prompt.

If your system is part of a VAXcluster, you will have to execute AUTOGEN with this command after the upgrade finishes.

```
$ @SYS$UPDATE:AUTOGEN SAVPARAMS REBOOT
```

Note: If your system includes a CI780 or CI750, VAX/VMS waits for several minutes to form or join a cluster. If the upgrade procedure does not continue within 5 minutes, please submit a Software Performance Report.

The decompressed libraries require approximately 4000 additional blocks of disk space and the decompression process may take up to 2 hours depending on the type of processor you are using. A general guideline for decompressing individual library files is that HELP libraries increase by 50% and OBJECT libraries by 25% when decompressed.

These steps must be performed after the upgrade has finished:

1. Reset any SYSGEN parameters (such as SCSNODE, ALLOCLASS or VAXCLUSTER) that may have been modified to allow the upgrade to proceed.
2. Apply the mandatory update to your system according to Chapter 2 of this manual. Until the mandatory update is applied, layered products cannot be installed.
3. Reinstall DECnet if you are running DECnet. This product must be reinstalled.

3.4 Special File Handling During Upgrade

After you have completed the upgrade, you may wish to remove files you no longer need. To do so, you need to know that certain files are handled specially during the upgrade. These include files that may have been edited by users, and shareable images that may have been linked into user images.

The distribution-kit copies of the following files become the latest version, and any pre-Version 4.2 copies become older versions of the files. Note that some of these files are new with Version 4.2 and will not be a part of your previous system.

```
[SYSLIB]*RTL*.EXE
[SYSLIB]CDDSHR.EXE
[SYSLIB]DCLTABLES.EXE
[SYSLIB]LBRSHR.EXE
[SYSLIB]SCRSHR.EXE
[SYSLIB]SMGSHR.EXE
[SYSLIB]SUMSHR.EXE
[SYSMGR]RTTLOAD.COM
[SYSMGR]SYSTARTUP.COM
[SYSMGR]SYCONFIG.COM
[SYSMGR]SYSHUTDWN.COM
[SYSMGR]STARTNET.COM
[SYSEXE]STARTUP.COM
[SYSEXE]SHUTDOWN.COM
[SYSHLP]HELPLIB.HLB
[SYSLIB]ENCRYPSHR.EXE
[SYSLIB]IMAGELIB.OLB
[SYSLIB]STARLET.OLB
```

You may wish to reconcile the new versions with any local modifications to these files before deleting the older versions.

The upgrade procedure does not copy older versions of the following files to the system disk:

```
[SYSEXE]NOTICE.TXT
[SYSEXE]RIGHTSLIST.DAT
[SYSEXE]SYSUAF.DAT
```

The upgrade does not delete the following files, but you may delete them manually after the upgrade has completed:

```
[SYSEXE]STARTUP.UP5
[SYSEXE]UPGRADE.KIT
```

The upgrade procedure may have changed the size of the following files to better fit the system; so you may want to check these files to be sure that the sizes are appropriate:

```
[SYSEXE]SYSDUMP.DMP
[SYSEXE]PAGEFILE.SYS
[SYSEXE]SWAPFILE.SYS
```

3.5 Installed Images

As part of the upgrade, the file SYS$MANAGER:VMSIMAGES.DAT is created for your site. This is a combined list of files that are installed for enhanced system performance. Some files in this list may already be in your site's SYS$MANAGER:SYSTARTUP.COM file and should be removed from SYSTARTUP.COM.

3.6 Creating Alternate Roots on a Common System Disk

In a VAXcluster that features a common system disk, the MAKEROOT.COM command procedure is used to create system roots for nodes other than the first node of the cluster. To invoke MAKEROOT, enter this command:

```
$ @SYS$MANAGER:MAKEROOT
```

Note that MAKEROOT will abort if your system disk is not a cluster common system disk, or if the SYSGEN parameter VAXCLUSTER is 0.

When MAKEROOT executes, it requests the name of the new system root. Enter the root name using the form SYSx, where x is a hexadecimal digit in the range 1 through 9 and A through D (for example, SYS1 or SYSA). Note that system roots SYSE and SYSF are reserved for other system functions.

You may not specify the root of the currently running system (usually SYS0) and if you specify any other existing system root, you are asked if you wish to continue, that is, if you want to modify the existing system root. If you do want to modify the existing system root, MAKEROOT deletes the following files from it:

- [SYSEXE]MODPARAMS.DAT
- [SYSEXE]VAXVMSSYS.PAR
- [SYSMGR]VMSIMAGES.DAT

If a SYSCOMMON directory exists in the directory tree, it too will be deleted.

Note that MAKEROOT does not check the format of the existing system root. DIGITAL recommends you delete all the files in the directory tree, and the directory tree itself, or choose a different root.

Next, MAKEROOT asks for the new node name and its SCSSYSTEMID. The node name can be no more than six characters.

After you provide the new node name and the node's SCSSYSTEMID value, MAKEROOT prompts for the size of the new root's page file and swap file. The values you provide are subsequently used by AUTOGEN.

Finally, MAKEROOT creates the new directory tree, the page file, and the swap file. It also generates a VAXVMSSYS.PAR file for the new root using the SYSGEN parameters of the currently running node as the basis for the new node.

When MAKEROOT completes this, it displays the following message:

```
Now you must build console media for the new system.  This
can be done in the following manner:

Copy all the files from the console to the disk, using
the following commands:

$ @SYS$UPDATE:CONSCOPY  ""  SAVE  ""  CSA1:
$ DISMOUNT CSA1:

Next, remove your console media, insert a scratch console media,
and enter the following commands:

$ @SYS$UPDATE:CONSCOPY  ""  RESTORE  ""  CSA1:
$ EXCHANGE COPY CSA1:DEFBOO.CMD *.*

Edit DEFBOO.CMD, and adjust the value deposited in R5 to contain
the root name in the high 4 bits.  For example, if you are building
the console for root SYS5, set the value deposited in R5 to 50000000
(50004000 on a VAX-11/782).

After you edit DEFBOO.CMD, copy it to another disk file,GENBOO.CMD,
for instance, edit GENBOO.CMD to include the low bit set in R5.
For example, if R5 was set to 50000000 in DEFBOO.CMD,
set it to 50000001 in GENBOO.CMD.  This will boot VAX/VMS
out of root 5, stopping in SYSBOOT.

$ EXCHANGE COPY DEFBOO.CMD,GENBOO.CMD CSA1:
$ DISMOUNT CSA1:

Remove the new console, and replace the original console
media in the drive.

Insert the newly-created console media in the console drive
on the new node.

You must now boot the target node into the newly-created root.
Use the conversational bootstrap GENBOO.CMD, which you just
created, as you must change the startup command procedure.

>>> @GENBOO.CMD

When SYSBOOT prompts, enter the following commands:

SYSBOOT> SET /STARTUP SYS$SYSTEM:STARTUP.COM
SYSBOOT> SET STARTUP_P1 "MIN"
SYSBOOT> CONTINUE
```

Following the system boot, login and invoke AUTOGEN with this command:

```
$ @SYS$UPDATE:AUTOGEN SAVPARAMS REBOOT
```

When the system comes up, be sure to install the mandatory update described at the start of this chapter.

4 Notes for the General User

This chapter presents additions and restrictions in Version 4.2 of VAX/VMS that are of interest to general users. This information is not documented elsewhere in the VAX/VMS Version 4.2 documentation kit.

4.1 DELETE and PURGE Commands: Access Requirements for the /ERASE and /LOG Qualifiers

The DELETE and PURGE commands require that you have the following types of access if you specify the /ERASE or /LOG qualifier:

- /ERASE requires write, read, and delete access to any file to be deleted.
- /LOG requires read and delete access to any file to be deleted.

If you do not specify the /ERASE or /LOG qualifier, delete access is sufficient.

We expect to remove the requirement of write access for the /ERASE qualifier in a future release.

4.2 LOGIN Error Messages

The following is a list of error messages not documented in previous versions of VAX/VMS.

- LOGIN - breakin evasion in effect

  Facility: LOGIN, Login Processor

  Explanation: Access to an account was denied because breakin evasion was in effect for that account. (This error is not displayed to the user attempting to log in, but appears in the accounting record and security audit record. The user sees the message "User authorization error".)

  User Action: None.

- LOGIN - error accessing network authorization file

  Facility: LOGIN, Login Processor

  Explanation: LOGIN could not read SYS$SYSTEM:NETUAF.DAT.

  User Action: Check to see that the file exists and has system read access.

- LOGIN - error activating command interpreter tables 'table name'

  Facility: LOGIN, Login Processor

  Explanation: LOGIN cannot access the specified command interpreter tables.

  User Action: Check to see that the CLI table file in question exists and is accessible.

- LOGIN - error connecting to 'terminal name'

  Facility: LOGIN, Login Processor

  Explanation: An error occurred attempting to reconnect to the specified disconnected terminal.

  User Action: Retry. Contact the system manager if the problem persists.

- LOGIN - error protecting command interpreter tables pages

  Facility: LOGIN, Login Processor

  Explanation: An error occurred write-protecting the command interpreter table pages.

  User Action: Retry. If the problem persists, consult your system manager.

- LOGIN - invalid password

  Facility: LOGIN, Login Processor

  Explanation: The specified password is incorrect. (This error is not displayed to the user attempting to log in, but appears in the accounting record and security audit record. The user sees the message "User authorization error".)

  User Action: None.

- LOGIN - no such user

  Facility: LOGIN, Login Processor

  Explanation: The specified username does not exist. (This error is not displayed to the user attempting to log in, but appears in the accounting record and security audit record. The user sees the message "User authorization error".)

  User Action: None.

- LOGIN - system password timeout

  Facility: LOGIN, Login Processor

  Explanation: A user failed to correctly enter the system password within the allowed time (LGI_SYSPWDTMO). (This message only appears in the system accounting and audit logs. No message is displayed to the user when this event occurs.)

  User Action: None.

- LOGIN - you are not authorized to do reconnections

  Facility: LOGIN, Login Processor

  Explanation: Your authorization profile does not permit you to connect to disconnected jobs.

  User Action: Retry without specifying /CONNECT.

- LOGIN - you are not authorized to log in from this source

  Facility: LOGIN, Login Processor

  Explanation: Your authorization profile does not permit you to login from the mode of access you are attempting to use (for example, dialup, SET HOST).

User Action: Log in using a different connection, or contact your system manager to have your authorization changed.

- LOGIN - you are not authorized to log in at this time

  Facility: LOGIN, Login Processor

  Explanation: Your authorization profile does not permit you to use the system at this time.

  User Action: Wait until a time at which you are authorized to use the system, or contact your system manager to have your authorization changed.

- LOGIN - you are not authorized to log in today

  Facility: LOGIN, Login Processor

  Explanation: Your authorization profile does not permit you to use the system today.

  User Action: Wait until a day on which you are authorized to use the system, or contact your system manager to have your authorization changed.

- LOGIN - you are not authorized to specify CLI parameters

  Facility: LOGIN, Login Processor

  Explanation: Your authorization profile does not permit you to specify an alternate CLI or CLI tables.

  User Action: Retry without specifying qualifiers.

- LOGIN - your account has expired - contact your system manager

  Facility: LOGIN, Login Processor

  Explanation: The expiration date of your account has passed.

  User Action: Contact your system manager to have your account renewed.

- LOGIN - your password has expired - contact your system manager

  Facility: LOGIN, Login Processor

  Explanation: The expiration date of your password has passed, and you did not change it at your last chance.

  User Action: Contact your system manager to have your password changed.

- LOGIN - account is auto-login only

  Facility: LOGIN, Login Processor

  Explanation: An attempt was made to log into an account whose AUTOLOGIN flag was set in the UAF record, using a normal interactive login or an explicit DECnet access control string. (This error is not displayed to the user attempting to log in, but appears in the accounting record and security audit record. The user sees the message "User authorization error".)

  User Action: None.

- LOGIN - account is disabled

  Facility: LOGIN, Login Processor

  Explanation: An attempt was made to log into an account whose DISUSER flag was set in the UAF record. (This error is not displayed to the user attempting to log in, but appears in the accounting record and security audit record. The user sees the message "User authorization error".)

  User Action: None.

- LOGIN - invalid SYS$INPUT for interactive login

  Facility: LOGIN, Login Processor

  Explanation: An attempt was made to initiate an interactive process with a file specified as SYS$INPUT.

  User Action: Use a non-file-structured device for SYS$INPUT on interactive processes. Storage devices are not permitted. The express intent of this error is to prevent LOGIN from reading username and password from a file.

- LOGIN - licensed number of system users exceeded

  Facility: LOGIN, Login Processor

  Explanation: The maximum number of interactive users for which your system is licensed are already logged in.

  User Action: Try to log in again later when other users have logged off; contact your system manager to have the system's license upgraded.

4.3 Mail Utility

This section describes changes to the Mail Utility.

4.3.1 Using the Mail Utility in a Cluster Running Mixed Versions of VAX/VMS

If a user creates a MAIL.MAI file under Version 4.2 and someone attempts to send MAIL to that user on a cluster node that is running a previous version of VAX/VMS, the sender will receive the following error message:

```
%MAIL-E-SENDERR, error sending to user 'username'
-RMS-F-DUP, duplicate key detected (DUP not set)
```

The sender should resubmit the mail message to a node on the cluster that is running Version 4.2.

4.3.2 /SELF Qualifier Documentation Incomplete for MAIL Command

The description of the /SELF qualifier in the *VAX/VMS Mail Utility Reference Manual* should include the following information:

- The /SELF qualifier is negatable.
- If you send a message from the DCL level (that is, you do not receive the MAIL> prompt from within the Mail Utility), specifying /SELF or /NOSELF overrides any setting you have established by the SET COPY_SELF command within the Mail Utility.
- Specifying /SELF or /NOSELF on the DCL command line has no effect if you enter the Mail Utility and receive the MAIL> prompt.

Thus, for example, you could specify the following command to send MYFILE.DAT to user CHAMPAGNE, and avoid receiving a copy of the file yourself even if you have previously entered the SET COPY_SELF command within the Mail Utility.

```
$ MAIL/NOSELF MYFILE.DAT CHAMPAGNE
```

4.4 Sort/Merge Utility: Corrections to the Specification Files

The following section describes corrections made to the Sort/Merge Utility specification files for Version 4.2.

- The specification files containing the /FIELD and /KEY specifications were incorrectly interpreted. One problem caused the entire record to be interpreted as a key. Another problem caused an incorrect interpretation of the length of an integer key. Both of these problems have been corrected.
- Packed decimal constants in /INCLUDE, /OMIT, and /TEST specifications were incorrectly interpreted. This problem has been resolved.
- Some Version 3.0 specification files were incorrectly translated by the specification file translator. This correction affects pre-version 4.2 specification files that activated key stripping.

4.5 SPAWN Command

The following section describes corrections to the SPAWN Command.

4.5.1 Output to a Spooled Device

In VAX/VMS Version 4.1, SPAWN would not accept a spooled device as an output specification. This restriction has been removed for Version 4.2.

4.5.2 Specifying a Search List

In VAX/VMS Version 4.1, specifying a searchlist in either the explicit input or output file specifications resulted in permanent allocation of Process I/O space, thus limiting the number of files that DCL could have open at a given time. This problem has been fixed in Version 4.2.

4.5.3 Using a Search List

In VAX/VMS Version 4.1, specifying an explicit input file specification using a search list could result in the file not being found if it did not exist in the first location in the searchlist. This problem has been resolved in Version 4.2.

4.6 VAX Text Processing Utility (VAXTPU)

This section presents additional information about the VAX Text Processing Utility (VAXTPU) that is not included in the *VAX Text Processing Utility Reference Manual.*

4.6.1 Moving the Cursor with Line Editing Keys

If you use DCL line editing keys to move the cursor position on the command line of a VAXTPU interface and a broadcast message is sent to your terminal, the cursor will be placed at the end of the command line after the broadcast is received (rather than being placed at its previous location). To correct the cursor's position, move the cursor to the desired location.

4.6.2 Changing the Editing Process from EVE to EDT

To change the default editing interface from EVE to the EDT Keypad Emulator, copy the section file SYS$LIBRARY:EDTSECINI.GBL to SYS$LIBRARY:TPUSECINI.GBL

4.6.3 Running VAXTPU with the /NODISPLAY Qualifier

The VAXTPU built-in procedures SPAWN and ATTACH will not work when you run VAXTPU with the /NODISPLAY qualifier.

4.6.4 Converting an Indexed File

If you read an indexed file into a VAXTPU buffer and then write it out, the file is converted from indexed to sequential organization. VAXTPU does not put out a message identifying the change.

4.6.5 The SAVE Built-In Procedure

The SAVE ('filespec') built-in procedure does not close the file that is saved until you exit from VAXTPU.

4.6.6 Setting a SHIFT_KEY for VAXTPU

There is no SET command to specify the comment associated with the editor's SHIFT_KEY. If a key that you are setting to the shift key was previously defined with a program and/or an associated comment, the LOOKUP_KEY built-in procedure will return the program or comment previously associated with the key (even though the key does not perform its previous function). If you then set a different key to be the SHIFT_KEY, the key that was previously set to the SHIFT_KEY is restored to its original definition. For example:

```
DEFINE_KEY ('MOVE_VERTICAL (1)', PF4, 'move');
SET (SHIFT_KEY, PF4);
LOOKUP_KEY (PF4, COMMENT);
```

The previous statements will cause the comment "move" to be returned for PF4, even though it is functioning as the editor's SHIFT_KEY. The following statement will cause PF4 to return to its MOVE_VERTICAL function:

```
SET (SHIFT_KEY, PF3);
```

If this behavior is unacceptable to you, use UNDEFINE_KEY (new-shift-key) before using SET (SHIFT_KEY, new-shift-key).

5 Notes for the System Manager

This chapter presents additions and restrictions in Version 4.2 of VAX/VMS that are of interest to system managers (including security and network managers). This information is not documented elsewhere in the Version 4.2 documentation kit.

5.1 Adding an Operating System to an Existing System Disk

There are cirumstances in which you will find it convenient to add another operating system to a system disk:

- You want to test software on a new operating system version. You could, for example, install the new version on an alternate directory root and create a bootstrap command procedure to select that version for testing sessions.
- You need to conserve disk drives.

Note that you must add the new system on an unused directory root. Use the **ADD** function of VMSKITBLD.COM in SYS$UPDATE, as illustrated in the following prodedure. (Do not confuse this function with the MAKEROOT.COM command procedure, which creates system directory roots for VAXCluster member nodes.)

1 Log in under the system manager's account.

2 Establish SYS$UPDATE as the default directory:

```
$ SET DEFAULT SYS$UPDATE
```

3 Place the target system disk (assuming you are using a removable disk) in an appropriate drive and put it on line.

4 Enter the following command to invoke VMSKITBLD.COM:

```
$ @VMSKITBLD
```

5 In response to VMSKITBLD prompts, supply the requested information.

6 After you supply the information, VMSKITBLD asks:

```
Operation [BUILD,ADD,COPY,COMMON]?
```

Enter the word ADD.

You will receive messages that either prompt you for information needed to complete the operation, or inform you of the procedure's status.

When you are prompted for the mounted source disk name, enter SYS$SYSROOT: if the source is a common root; otherwise, enter SYS$SYSDEVICE:.

When you are prompted for the source disk top level system directory, enter the directory from which you wish to copy system files.

When you are prompted for the target disk top level system directory, be sure to enter a directory root not already in use. (Note that in addition to roots used by existing systems on the disk, roots SYSE and SYSF are reserved for other system functions.)

A typical message sequence might look like this:

```
Enter mounted source disk name (DDCU:): SYS$SYSROOT:
Enter SOURCE disk top level system directory [default = SYS0]: SYS4 [RET]
Enter target disk name (DDCU:): SHEMP$DUA1: [RET]
Enter the target disk's label [default = VAXVMSRL4]: [RET]
Enter TARGET disk top level system directory [default = SYS0]: SYSA [RET]
  Allocate and mount target disk(s).
  _SHEMP$DUA1: allocated
%MOUNT-I-MOUNTED, VAXVMSRL4   mounted on _SHEMP$DUA1:
Create directory entries on the target disk.
Copy the system executive files.
Copy the system library files.
Copy the system message files.
Copy the system manager files.
Copy the system update command files.
Copy the system EXE files.
Copy the system help files.
Write a bootblock.
Copy BLISS require files and STARLET DCL library.
Copy coding examples.
Copy the UETP files.
```

VMSKITBLD.COM informs you when the operation completes by sending the following message to your terminal:

```
Kit is complete.
```

At this point, the target system directory contains all the required VAX/VMS files for a complete system. (Note that if any layered products from the source system are needed on the new system, you must reinstall them.)

7 Create a bootstrap command procedure, as follows, to boot the new operating system. (Note that for VAX–11/730 and VAX–11/725, the console device is CSA2:.)

a Copy DEFBOO.CMD (DEFBOO.COM for VAX 8600) from the console medium to your default disk directory, renaming the file SYxBOO.CMD (or SYxBOO.COM), where **x** represents the selected root. For example, if you wish to boot from SYSA, name the file SYABOO.CMD:

```
$ RUN SYS$SYSTEM:SYSGEN
CONNECT CONSOLE
$ MOUNT /FOREIGN CSA1:
$ EXCHANGE COPY /LOG CSA1:DEFBOO.CMD SYABOO.CMD
```

b Invoke an editor to modify SYABOO.CMD. The table below shows how to determine the line that you must change for your processor; it shows the line before and after you change it. Note that you change only the first digit.

Processor	Line to Change	Line After Change
VAX 8600	DEPOSIT R5 XXXXXXXX	DEPOSIT R5 AXXXXXXX
VAX–11/780	DEPOSIT R5 XXXXXXXX	DEPOSIT R5 AXXXXXXX
VAX–11/782	DEPOSIT R5 XXXXXXXX	DEPOSIT R5 AXXXXXXX
VAX–11/750	D/G 5 XXXXXXXX	D/G 5 AXXXXXXX
VAX–11/730	D/G/L 5 XXXXXXXX	D/G/L 5 AXXXXXXX
VAX–11/725	D/G/L 5 XXXXXXXX	D/G/L 5 AXXXXXXX

c Copy the edited file back to the console medium:

```
$ EXCHANGE COPY /LOG SYABOO.CMD CSA1:
```

8 After shutting down the system and pressing CTRL/P to halt the processor, you can bootstrap with the procedure you created.

- For processors other than the VAX–11/750, enter the following command at the console prompt:

  ```
  >>> B SYA
  ```

- On the VAX–11/750, you can boot in two ways:

  a From unit 0 (DB0:,DM0:,DR0:,DU0:):

  ```
  >>> B/AXXXXXXX
  ```

  b From the console TU58:

  ```
  >>> B DDA0
  BOOT58> B SYA
  ```

When the SYSBOOT> prompt appears, enter the following commands:

```
SYSBOOT> USE DEFAULT
SYSBOOT> CONTINUE
```

If you are running in a VAXcluster, it is important to verify and, if necessary, reset values (using SHOW and SET commands) for the following SYSGEN parameters before rebooting:

- ALLOCLASS
- DISK_QUORUM
- QUORUM
- SCSSYSTEMID
- SCSSYSTEMIDH
- SCSNODE
- VAXCLUSTER
- VOTES

For more information on these parameters, refer to the *VAX/VMS System Generation Utility Reference Manual* or Chapter 5 in the *Guide to VAXclusters.*

9 After the system bootstraps, log in as system manager and run the AUTOGEN procedure described in Chapter 11 of the *Guide to VAX/VMS System Management and Daily Operations.*

5.2 Authorize Utility

This section describes changes to the Authorize Utility.

5.2.1 /ACCESS Qualifier

The syntax string for the /ACCESS qualifier to the MODIFY command has been enhanced to allow more readable, flexible usage. The following commands produce identical results.

```
UAF> MODIFY SAM /ACCESS=(primary, 2-3, 5, secondary, 8-12)
UAF> MODIFY SAM /ACCESS="Primary: 2-3, 5; Secondary: 8-12"
UAF> MODIFY SAM /ACCESS=(p,2,s,8,p,3,s,9,p,5,s,10-12)
UAF> MODIFY SAM /ACCESS="2-3 SEC 8-12 PRIM 5"
```

5.2.2 /PWDEXPIRED and /PWDLIFETIME Qualifiers

On page AUTH-13 of the *VAX/VMS Authorize Utility Reference Manual*, the /PWDEXPIRED and /PWDLIFETIME qualifiers should appear as /[NO]PWDEXPIRED and /[NO]PWDLIFETIME, respectively.

5.2.3 /DEFPRIVILEGES and /PRIVILEGES Qualifiers

You can specify the keyword [NO]ALL for the /DEFPRIVILEGES and /PRIVILEGES qualifiers to disable/enable all user privileges.

5.2.4 Secondary Passwords

Beginning with Version 4.2, users cannot initially give themselves secondary passwords. The initial setting of the secondary password must be done by the system manager using the Authorize Utility. The reason for this change is to protect careless users who leave their terminal sessions unattended.

In earlier versions of VAX/VMS, anyone could essentially render an account useless by simply adding a secondary password that the account's owner did not know. If a user tries to initiate a secondary password on VAX/VMS Version 4.2, the system will now respond as follows:

```
$ SET PASSWORD/SECONDARY
%SET-F-PWD2NOTSET, system manager must initially set secondary passwords
```

5.2.5 AUTOLOGIN Flag

A flag named AUTOLOGIN has been added to the flags field in the user authorization file (SYSUAF). The flag is set by specifying the qualifier /FLAGS=AUTOLOGIN to one of the following Authorize Utility commands: ADD, MODIFY, or COPY. When set, it makes the account available only by using the autologin mechanism. The following forms of access are disabled:

- Login by any terminal, LAT connection, or SET HOST involving presentation of username and password
- Access by DECnet task using explicit access control

The following forms of access remain permitted:

- Interactive login by the autologin mechanism
- Batch jobs
- Proxy access by DECnet task

5.3 AUTOGEN Command Procedure

This section describes modifications to AUTOGEN.COM.

5.3.1 Changes to SYS$SYSTEM:PARAMS.DAT

The SMALLDISK parameter has been replaced by the DISKSIZE parameter. The DISKSIZE parameter indicates the capacity of the system disk in blocks.

Three new configuration parameters have been added to PARAMS.DAT:

1. NUM_HOST—Number of VAX/VMS systems that are members of a VAXcluster
2. NUM_SERVER—Number of disk or tape servers that use the MSCP protocol (HSC50, UDA50, VAX systems running the MSCP server software)
3. WSTYPE—Type of VAXstation present on the system, if any

If you are running in a VAXcluster, you should add NUM_HOST and NUM_SERVER to MODPARAMS.DAT on each node, specifying the number of hosts and number of servers visible from that node.

5.3.2 Specification of AUTOGEN Shutdown Time

To specify AUTOGEN shutdown time, you must define the logical name AGEN$SHUTDOWN_TIME using the DCL command DEFINE. The logical name value is the number of minutes until shutdown. Note that the VAX/VMS Version 4.0 AUTOGEN parameter SHUTDOWN_TIME is no longer valid.

5.3.3 SYSGEN Parameter Values Calculated by AUTOGEN

As of VAX/VMS Version 4.2, AUTOGEN calculates appropriate values for the SYSGEN parameter SCSCONNCNT.

AUTOGEN no longer propagates site-specific values for the SYSGEN parameters LRPSIZE and SRPSIZE. To override default values for these parameters, you must edit the file SYS$SYSTEM:MODPARAMS.DAT and specify your own site-specific values.

5.4 Backing Up the System Disk: Recommended Procedure

The following material replaces Sections 2.8.2 through 2.8.2.3 of the *Guide to VAX/VMS System Management and Daily Operations* and Sections 4.7 through 4.7.3 of the *Guide to VAX/VMS Software Installation.*

5.4.1 Using Stand-Alone Backup

To back up your system disk, DIGITAL recommends that you use stand-alone BACKUP, which employs a subset of Backup Utility qualifiers. (Refer to the *VAX/VMS Utilities Reference Volume* for a complete description of the Backup Utility and its qualifiers.) It is especially important that you understand the functions of the /IMAGE and /PHYSICAL qualifiers before using stand-alone BACKUP.

If your system was not distributed on magnetic tape, you must build a stand-alone BACKUP kit either on console media or on disk. You can then bootstrap stand-alone BACKUP from the console block storage device or from the alternate directory root SYSE on a Files–11 disk.

Installing and using stand-alone BACKUP in an alternate root on your system disk saves time when backing up your system disk, because you do not have to boot stand-alone BACKUP from your console volume, which is a relatively slow process.

The procedures described here apply to all VAX processors. However, you should note the following:

- You can install stand-alone BACKUP in any available disk directory root from SYS1 through SYSE; however, DIGITAL has established SYSE as the standard alternate system directory root for stand-alone BACKUP. The discussion in this section assumes you will install stand-alone BACKUP in SYSE
- You cannot use these procedures to restore your operating system to the system disk from which stand-alone BACKUP is booted.
- There are some limitations with regard to the VAX–11/750, which are noted within the procedures.
- The console device is CSA1 on VAX systems with a single console drive and CSA2 on systems with two console drives.

The following sections explain how to

- Install stand-alone BACKUP in alternate directory root SYSE
- Create a command procedure to bootstrap stand-alone BACKUP from SYSE.
- Bootstrap stand-alone BACKUP from SYSE

5.4.2 Installing Stand-Alone BACKUP in Alternate Directory Root SYSE

You install stand-alone BACKUP in SYSE using the STABACKIT command procedure. First, log in to the SYSTEM account. Then enter this command:

```
$ @SYS$UPDATE:STABACKIT SYS$SYSDEVICE:
```

Be sure to enter the command exactly as shown. You will not be prompted for the destination if you forget to enter it.

After you install stand-alone BACKUP in SYSE, the next step is to create a command procedure to boot it.

5.4.3 Creating a Command Procedure to Bootstrap Stand-Alone BACKUP from SYSE

The recommended procedure for creating a command procedure to boot stand-alone BACKUP from alternate root SYSE is to modify an existing boot command procedure. Ideally, this should be the default boot command procedure, DEFBOO.CMD (DEFBOO.COM for VAX 8600).

First, make a copy of the boot procedure in your default device directory. Then, edit the copy as instructed. Finally, give the edited copy an appropriate file name, and store it on the console volume.

You may choose any unique name in the form xxxBOO.CMD (or xxxBOO.COM for VAX 8600) for the command procedure you create. However, DIGITAL suggests you use a file name identical to the copied file, except that the first letter should be an **X**.

For example, if you use a copy of DEFBOO.CMD, the new file should be named XEFBOO.CMD.

The following procedure assumes you will use a copy of DEFBOO.CMD and rename it XEFBOO.CMD.

1 If necessary, use the following command sequence to configure the console device:

```
$ RUN SYS$SYSTEM:SYSGEN
SYSGEN> CONNECT CONSOLE
SYSGEN> EXIT
```

2 Mount the console device:

```
$ MOUNT/FOREIGN CSA$1
```

3 Invoke the Exchange utility to make a copy of DEFBOO.CMD in your default disk directory:

For VAX-11/780 and VAX-11/750:

```
$ EXCHANGE COPY CS1:DEFBOO.CMD *
```

For VAX-11/730 and VAX-11/725:

```
$ EXCHANGE COPY CS2:DEFBOO.CMD *
```

4 Rename the copy of DEFBOO.CMD in your default directory.

```
$ RENAME DEFBOO.CMD XEFBOO.CMD
```

5 Invoke an editor to modify XEFBOO.CMD. The table below shows how to determine the line that you must modify for your processor; it shows the line before and after you change it. Note that you change only the first digit.

Processor Type	Line to Change	Line After Change
VAX 8600	DEPOSIT R5 XXXXXXXX	DEPOSIT R5 EXXXXXXX
VAX–11/780	DEPOSIT R5 XXXXXXXX	DEPOSIT R5 EXXXXXXX
VAX–11/782	DEPOSIT R5 XXXXXXXX	DEPOSIT R5 EXXXXXXX
VAX–11/750	D/G 5 XXXXXXXX	D/G 5 EXXXXXXX
VAX–11/730	D/G/L 5 XXXXXXXX	D/G/L 5 EXXXXXXX
VAX–11/725	D/G/L 5 XXXXXXXX	D/G/L 5 EXXXXXXX

6 When you have edited the file, call EXCHANGE to store it on the console volume:

For VAX-11/780 and VAX-11/750:

```
$ EXCHANGE COPY XEFBOO.CMD CSA1:
```

For VAX-11/730 and VAX-11/725:

```
$ EXCHANGE COPY XEFBOO.CMD CSA2:
```

5.4.4 Bootstrapping Stand-Alone BACKUP from SYSE

To boot stand-alone BACKUP from SYSE, you must perform the following operations:

1 Shut the system down.

```
$ @SYS$SYSTEM:SHUTDOWN.COM
```

2 The procedure asks the following questions. Respond by pressing the RETURN key.

```
How many minutes until final shutdown? [0] RET
Reason for shutdown? [Standalone] RET
Do you want to spin down the disk volumes? [NO] RET
```

3 As the shutdown continues, the console terminal prints several shutdown messages. When the console terminal prints the following message, shutdown is completed:

```
SHUTDOWN COMPLETE - USE CONSOLE TO HALT SYSTEM
```

At this point, halt the processor using CTRL/P.

4 When the processor halts, enter the following command to boot stand-alone BACKUP from SYSE:

For VAX-11/750:

```
>>> B/EXXXXXXX
```

For all other processors:

```
>>> B XEF
```

5 If you are not using a modified version of the default bootstrap command procedure to boot stand-alone BACKUP, use the first three characters in the name of the bootstrap command procedure you created. For example, if you are using XU0BOO.CMD, type the following command:

```
>>> B XU0
```

5.5 Batch/Print Facility

This section contains corrections and enhancements to the Batch/Print facility, especially with respect to the print symbiont.

5.5.1 General Formatting Problems Resolved

The following is a list of problems that have been corrected for VAX/VMS Version 4.2:

- The Version 4.0 print symbiont incorrectly interprets null prefix and null postfix characters in FORTRAN and PRINT carriage control files. This problem, fixed in Version 4.1, caused output (for instance, PL/I listings) to wrap on output queues serving a printer controlled by the terminal driver. All modifications required to properly interpret the prefix and postfix characters have been included in the Version 4.2 update.
- In Version 4.0 the job reset sequence is not performed upon abortion of the previous job. This may result in the loss or corruption of output. The Version 4.2 print symbiont has been modified to properly perform the job reset sequence whenever a job completes regardless of the completion status of that job.
- The Version 4.0 print symbiont is not capable of positioning output to the first printable line of the paper. This positioning problem, fixed in Version 4.1, no longer exists.
- In Version 4.0 the print symbiont fails to properly terminate an escape sequence if that sequence spans across multiple records. Furthermore, data embedded within an escape sequence is not properly interpreted. These two related problems are resolved in Version 4.2.
- An extra blank line is generated on paper if the bottom margin is nonzero and FEED is enabled for the queue or specified on the PRINT command in Version 4.0. In Version 4.2 this problem no longer occurs.
- All form feeds generated by the standard Version 4.2 print symbiont are preceded by a carriage return. In the event that form feed generated by the standard Version 4.2 print symbiont is determined unnecessary (by the main format routine), then both the form feed and the carriage return are discarded.

Note: User modified symbionts need not be relinked after update of Version 4.2.

5.5.2 Print Symbiont Termination in Creation of File Name Banner

Previously in Version 4.0 of VAX/VMS, the print symbiont could abort with the following error message displayed on the console terminal:

```
%%%%%%%%%%%%  OPCOM     date-time  %%%%%%%%%%%%
Message from JOB_CONTROL
%JBC-W-SYMDEL, unexpected symbiont process termination
%%%%%%%%%%%%  OPCOM     date-time  %%%%%%%%%%%%
Message from JOB_CONTROL
-SYSTEM-F-ACCVIO, access violation, reason mask=6E,
        virtual address=03000000, PC=00000000, PSL=2FFC0000
```

This problem occurred when a batch job failed due to the inability of the system to create the batch log file. With this failure the job controller created a special print job to signal the error condition. The print symbiont subsequently aborted when attempting to create a file-separation page because no file identification is provided in the special message from the job controller.

The Version 4.2 print symbiont properly creates a file separation page and no longer causes access violations.

5.5.3 Print Symbiont Process Termination or Errant Communication Looping

The following operator console error message is associated with a VAX/VMS Version 4.0 communication problem.

```
%%%%%%%%%%%%  OPCOM     date-time  %%%%%%%%%%%%
Message from JOB_CONTROL
%JBC-W-SYMDEL, unexpected symbiont process termination
%%%%%%%%%%%%  OPCOM     date-time  %%%%%%%%%%%%
Message from JOB_CONTROL
-PSM-F-BADLOGIC, internal logic error detected at PC 0000B46E
```

This same problem may force the print symbiont into an internal compute loop. The messages sent to the job controller have been revised in order to insure the proper communication in the VAX/VMS Version 4.2 print symbiont.

5.5.4 Print Symbiont Looping Problems Resolved

The VAX/VMS Version 4.0 print symbiont process will enter a compute loop if an output queue, paused at end of file, is restarted. Furthermore, if an attempt is made to position the output queue to a page less than page #1, then the print symbiont process will enter a compute loop. Both of these problems are resolved in VAX/VMS Version 4.2.

5.5.5 Tab Expansion Determined at Start of Queue

When the output queue is started, the VAX/VMS Version 4.2 print symbiont determines if tab expansion is required by accessing the current device characteristics. The Version 4.2 print symbiont expands horizontal tabs only when the device is incapable of handling the tab character. On a device controlled by the LCDRIVER or LPDRIVER, the DCL command SET PRINTER/TAB will set the tab characteristic for that device. On a serial line controlled by the terminal driver, the DCL command SET TERMINAL/TAB will set the tab characteristic for that serial device.

The device characteristics for a particular output queue are determined at the START of that output queue. Therefore, we recommend setting the device characteristics before starting the output queue. If the characteristics of a device need to be reset after the output queue has been started then we recommend stopping the queue, resetting the device characteristics, and then, restarting the output queue. Please be sure the output queue has completely stopped before changing the device characteristics.

5.5.6 Generation of Blank Pages When Setup or Reset Sequence is Specified

In Version 4.0 of VAX/VMS, it is possible to create library setup/reset modules which will be output to the device during the processing of the current print job. Setup/reset modules may be output before a specific file, before all files, or after the current job is completed. The VAX/VMS Version 4.0 print symbiont incorrectly inserts form feeds after all setup or reset modules regardless of content. In VAX/VMS Version 4.2, only those modules that insert printable text will be followed by a form feed. No form feed will

be inserted after a recognized escape sequence, device control sequence, or operating system command.

We realize that certain limitations exist for output devices that require control sequences in the ASCII range of printable characters. Certain limitations may also exist for those devices that allow the user to reposition output to the top of the page after insertion of printable text. We believe this area of the symbiont may require additional flexibility beyond that provided in this functional update. We are currently investigating mechanisms by which additional flexibility may be provided.

5.5.7 Device Reset Sequence and Form Feed Interaction

Blank pages issued between jobs may be due to interactions between form feed and the device reset escape sequence. Certain programmable devices require the form feed to precede the reset sequence. Extra page problems may be resolved on such devices by inserting a form feed before the reset escape sequence in the device control library module.

5.5.8 Formatting Problems with a Nonzero Left Margin

The VAX/VMS Version 4.0 print symbiont displays errant behavior when a nonzero left margin is defined on the current form. This print symbiont positions text to column one, rather than using the left margin value when a vertical format effector is encountered in the output stream. Furthermore, the text required to overstrike and bold output is also positioned to column one (rather than the left margin value). Both of these problems have been corrected in VAX/VMS Version 4.2.

5.5.9 Correction to the Expansion of Horizontal Tabs

In VAX/VMS Version 4.0 the expansion of horizontal tabs was incorrectly performed because the tab stop was based on a zero left margin rather than the current left margin. The proper expansion of horizontal tabs (when the left margin is nonzero) requires that all tabs be expanded by the print symbiont regardless of the capability of the device. The VAX/VMS Version 4.2 print symbiont will always expand tabs if the left margin defined in the current form definition is nonzero.

5.5.10 Print Symbiont Use of CTRL/O

The VAX/VMS Version 4.2 print symbiont cancels CTRL/O on the first write associated with every task (file to process). In Version 4.0 if a CTRL/O was issued on the device, then all jobs submitted to the controlling queue were lost until a subsequent CTRL/O was issued on that device.

5.5.11 Correction to the Burst Separation Page Error

If the user requested a burst separation page in VAX/VMS Version 4.0, the corresponding flag separation page was not printed and no burst of characters appeared over the perforation. The corrected behavior, implemented in this update, generates the corresponding flag separation page when only a request for burst separation page is made. Therfore, in Version 4.2 a singular request for a burst separation page is an implied request for both the burst separation page and the corresponding flag separation page. Subsequently, the burst of characters will appear over the perforation between the burst separation page and the corresponding flag separation page.

5.5.12 Using the PRINT Command with the /PASSALL Qualifier

In Version 4.0 of VAX/VMS it was not possible to bypass all the interpretation and formatting performed by the print symbiont main format routine even though the file was submitted in PASSALL mode. In Version 4.2 the DCL command PRINT/PASSALL will properly bypass the main formatting routine of the print symbiont. In bypassing the main format routine, the print symbiont will no longer generate page headers or initiate page setup. This change in behavior is considered a correction to previous errant behavior.

5.5.13 New Qualifier: /[NO]PAGE_SETUP

In VAX/VMS Version 4.2 the qualifier /PAGE_SETUP has been added to DCL command DEFINE/FORM. This qualifier allows the operator or system manager to define a setup module which will be output to the device before every output page. Page setup is ignored if the job is submitted in the PASSALL mode. Consult Appendix DCL of the *VAX/VMS DCL Dictionary* for further information.

5.5.14 Overriding the Base Process Priority Value

The print symbiont process is created by the job controller with a default base process priority value of 4. This value may be overridden with the DCL command START/QUEUE/BASE_PRIORITY=n.

In Version 4.0 of VAX/VMS, if the base priority of the symbiont was greater than that of the users on the local processor and a search was initiated on an output queue controlled by this symbiont, then the users may have experienced degraded system performance on that local processor. In order to avoid this problem, the Version 4.2 print symbiont sets the priority of its process to the system generation parameter DEFPRI upon the initiation of a search. When the search operation is completed, the print symbiont resets its base process priority to the value of JPI$_PRIB. The Version 4.2 behavior is designed to avoid undue competition for what may be scarce resources.

5.5.15 New Qualifier: /[NO]RECORD_BLOCKING

In VAX/VMS Version 4.2 the qualifier /[NO]RECORD_BLOCKING has been added to the DCL commands SET QUEUE, START/QUEUE, and INITIALIZE /QUEUE. This qualifier can be used to control the buffering of data from the symbiont to the output device. The default is /RECORD_BLOCKING. The use of /NORECORD_BLOCKING will only affect the performance of the specified output execution queue. Performance of other output queues controlled by the same symbiont process will not be significantly impaired.

Note that if you do not create a new system job queue file when you update your system to VAX/VMS Version 4.2, record blocking will be disabled for all output execution queues. Thus, printer, terminal, and server queues created before the upgrade and which previously buffered data to the output device will be marked /NORECORD_BLOCKING (no attribute for record blocking present) and will not buffer output to the output device.

In order to take advantage of the more efficient buffering of output to the device without creating a new system job queue file, issue the following DCL command for each output execution queue:

```
$ SET QUEUE /RECORD_BLOCKING queue_name
```

5.5.16 Enhanced Qualifier: /WSDEFAULT=n

In VAX/VMS Version 4.2 the qualifier /WSDEFAULT for the the DCL commands START/QUEUE and INITIALIZE/QUEUE also can be specified for an output execution queue. This qualifier establishes working set default to be used when creating the symbiont process. Further information is available in the *VAX/VMS DCL Dictionary*.

5.5.17 Enhanced Qualifier: /WSEXTENT=n

In VAX/VMS Version 4.2 the qualifier /WSEXTENT for the DCL commands START/QUEUE and INITIALIZE/QUEUE also can be specified for an output execution queue. This allows the user to define the working set extent to be used when creating the symbiont process. Further information is available in the *VAX/VMS DCL Dictionary*.

5.5.18 Enhanced Qualifier: /WSQUOTA=n

In VAX/VMS Version 4.2 the qualifier /WSQUOTA for the DCL commands START/QUEUE and INITIALIZE/QUEUE also can be specified for an output execution queue. This allows the user to define the working set page size (working set quota) to be used when creating the symbiont process. Further information is available in the *VAX/VMS DCL Dictionary*.

5.6 DECnet–VAX

This section describes modifications to DECnet–VAX.

5.6.1 False DECnet Error

There is a false error reported when running the Version 4.2 VMS UETP DECnet test phase in a shared system disk configuration. This error report looks like:

```
%UETP-W-TEXT, The process -ddddnnnnnnnnn- returned a final status of:
        %DIFF-F-OPENIN, error opening !AS as input
```

Where dddd is the circuit designation for the circuit in your system and nnnnnnnnn is node number and process count dependent. This error is caused by an error in the VAX/VMS DIFF utility and will be fixed in a future update of VAX/VMS. A workaround for this problem is to edit the file UETDNET00.COM and reverse the order of the parameters in the only DIFF command in the file.

5.6.2 Network Jobs Partially Counted Against User Job Quotas

In previous versions of VAX/VMS, network jobs were not counted against a user's job quotas (the UAF fields MAXJOBS and MAXACCTJOBS). In VAX/VMS Version 4.2, they are partially counted. Network jobs in excess of four jobs are now counted against MAXJOBS. Network jobs are still not counted against MAXACCTJOBS.

The reason for the partial exemption is that a wildcard copy of many small files can result in several network server processes on the remote node. This occurs when the local process accessing the files requests the next file access before the currently active network server is ready to accept the request. Lacking the exemption, an account with a small MAXJOBS might well be unable to correctly serve wildcard network file access requests.

When this problem is corrected in a future VAX/VMS update, network jobs will be counted against user and account job quotas with no exemptions.

5.6.3 New Logical Names in Network Jobs

In VAX/VMS Version 4.2, when a job is initiated by using a DECnet network connection, two new logical names are defined. Jobs falling into this category include SET HOST logins, and DECnet task logins (for example, remote file access, task access, and so on).

The two new logical names are defined in the job logical name table. They are as follows:

SYS$REM_NODE	Name of the remote node from which the job was originated.
SYS$REM_ID	Identification of the process on the remote node who initiated the job. This identification is, on VAX/VMS systems, either the process' username or PID, depending on whether proxies are in effect or not. Other operating systems may send other data (for example, terminal number) as the remote process identification.

5.6.4 Changes to Networking Guide Documentation

The following changes apply to the *Guide to Networking on VAX/VMS*:

In the calculation for the value of the SYSGEN parameter NPAGEDYN on page 5-41, the number 14000 in the equation should be changed to 27500, and the explanation given for the value 14000 should be changed to the following:

27,500 is the total number of bytes required to load the XE (UNA) driver (13,000 bytes) and NETDRIVER (14,500 bytes). If you need other drivers (DMC, DMP, DMF, CI, NO), the total will differ: 4300 bytes for the XM (DMC); 9800 bytes for the XD (DMP); 15,000 bytes for the XG (DMF); 3000 bytes for the CN (CI); 13,000 bytes for the NO (asynchronous) driver.

5.6.5 Changes to Network Control Program (NCP) Documentation

The following information applies to the *VAX/VMS Network Control Program Reference Manual*:

In the description of the STATE parameter in the NCP command SET /DEFINE CIRCUIT (see page NCP-64 in the Utilities Reference Volume), the reference to the CLEARED circuit state is incorrect and should be removed from the documentation. This state is not supported for DECnet–VAX.

5.6.6 Network Control Program (NCP) Messages

The following is a list of new messages for the Network Control Program (NCP).

- PURABO, Purge aborted due to errors

  Explanation: The purge requested for the NCP COPY command was aborted. The local node database was not affected.

  User Action: None needed. This is an informational message.

- EXEABO, Executor characteristics not defined

  Explanation: The executor characteristics that are usually defined after a PURGE command as part of the NCP COPY command were not defined. Also, if there is no executor node defined, this error will be output.

  User Action: None needed. This is an informational message.

- REMABO, Remote node not defined

  Explanation: The remote node name and address are not defined by the NCP COPY command. If the remote node does not have a name and address defined in the local database, this error will be output. If the PURGE command is executed, the remote node name and address will not be replaced in the local database

  User Action: None needed. This is an informational message.

5.6.7 Changes to DTS/DTR Documentation

The following information applies to the *VAX/VMS DECnet Test Sender /DECnet Test Receiver Utility Reference Manual.*

The description of the /[NO]DISPLAY qualifier on page DTS-8 should be replaced as follows:

/[NO]DISPLAY=number
Instructs DTS to print the specified number of bytes (in hexadecimal) of data and interrupt messages to DTR. The default is /NODISPLAY.

5.7 Error Log Utility

Several changes have been made to the Error Log Utility documentation, which is a manual in the *VAX/VMS Utilities Reference Volume.*

Under privileges and restrictions it should be noted that only read access is required to access the file ERRORLOG.SYS. (It is not necessary to rename the file ERRORLOG.SYS to ERRORLOG.OLD before using the Error Log Utility.)

If only a summary report is desired, the command line must specify both the /NOFULL qualifier and the /SUMMARY qualifier.

The following keywords have been added to the lists of keywords for the /EXCLUDE and /INCLUDE command qualifiers.

/EXCLUDE

ENVIRONMENTAL_ENTRIES — Exclude environmental entries from the report.
SNAPSHOT_ENTRIES — Exclude shapshot entries from the report.

/INCLUDE

ENVIRONMENTAL_ENTRIES — Include environmental entries in the report.
SNAPSHOT_ENTRIES — Include shapshot entries in the report.

The format of the error log report identification section has been slightly modified; the content remains the same. The third line in this section (which had specified the type of error log entry being reported, the date and time the entry was made, the processor type and revision level, and the system serial number) now consists of two lines: the third and the fourth. The third line now specifies the type of error being reported, and the date and time. The fourth line specifies the remainder of the information. The following example shows this change.

```
V A X / V M S        SYSTEM ERROR REPORT      COMPILED 6-MAR-85 14:39
                                                              PAGE   1.

***************************** ENTRY       5. *****************************
ERROR SEQUENCE 42.                              LOGGED ON SID 01380101

DEVICE ERROR,  5-MAR-85 14:42:16.93
               KA780 REV# 7. SERIAL# 257.

MASSBUS SUB-SYSTEM, UNIT _DBB1:
      RH780 CSR       00000020
                                    ADAPTER IS MBA
      RH780 CR        00000004
                                    INTERRUPT ENABLE
   .
   .
   .
```

5.8 Forced Error Handling with Digital Storage Architecture (DSA) Disks

Most VAX/VMS utilities and DCL commands treat the presence of the forced error flag as a fatal error. For example, if you use the DCL command COPY to move a file that contains a block with the forced error flag, the resulting error will cause the operation to terminate.

However, BACKUP is designed to continue in the presence of almost all errors, including forced errors. BACKUP will continue to process the file, creating a new copy of the file in the output saveset. An error message indicating the forced error will be displayed, but the forced error will NOT be present in the new copy of the file that is being created. Subsequent use of the new file (for example, in a restore operation) will indicate no errors. Thus, data that was formerly marked as bad with the forced error flag may be accidentally propagated and now seem correct.

System managers (and other users of BACKUP) should assume that forced errors reported by BACKUP signal degradation of the data in question, and act accordingly. The safest procedure is to replace the file containing the forced error with a good copy of the file from a previous BACKUP operation.

For more information on Digital Storage Architecture (DSA) and forced errors, see the *VAX/VMS I/O User's Manual*.

5.9 *Guide to VAX/VMS System Management and Daily Operations*: Documentation Changes

The following corrections apply to the VAX/VMS Version 4.0 *Guide to VAX/VMS System Management and Daily Operations*:

- Page 2-11. In the example in Section 2.3.5, the symbol definition for the INSTALL command should be as follows:

  ```
  $ INSTALL = $SYS$SYSTEM:INSTALL/COMMAND_MODE
  ```

- Page 2-32. Replace instructions in Step 3 with the following:

  When the console terminal prints the console mode prompt (> > >), issue the following command:

  For VAX-11/780, VAX-11/730, VAX-11/725:

  ```
  >>> B CS1
  ```

For VAX-11/750:

```
>>> B DDA0
```

In Step 4, proceed on VAX–11/750 systems as described for VAX–11/730 and VAX–11/725 systems.

- Page 3-15. Sections 3.4.1 through 3.4.1.2 should contain the same information as that in Sections 7.2.4 through Sections 7.2.4.2 in the *Guide to VAX/VMS Software Installation.*
- Page 4-24. In the example shown, the value to be deposited in R1 is F0E000, not E.
- Page 4-26. In Section 4.3.2, the text following the second bullet should be replaced with the following:

  Leave the system disk in the original drive. Place a backup copy in another drive, swap unit plugs in the two drives, and try again.
- Page 5-12. In example 5-2, the line

  ```
  $ TT == F$GETDVI("TT","DEVNAM")-"-"
  ```

  should be replaced as follows:

  ```
  $ TT == F$GETDVI("TT","DEVNAM")-"_"
  ```
- Page 5-13. In Example 5-3, the line

  ```
  $ GOTO F$MODE()
  ```

  should be replaced as follows:

  ```
  $ GOTO 'F$MODE()
  ```
- Page 5-23. In example 5-5, the local symbol definition for DIR, after the LOOPEND label, is incorrect and should be replaced with the following:

  ```
  $ LOOPEND:
  $ IF F$SEARCH("*.*;*") .NES. "" THEN DELETE *.*;*
  $ DIR = (F$DIRECTORY()-"]"-">")-F$PARSE("[-]",,,-
          "DIRECTORY")-"]"-">")-"."-"["-"<"c
  ```
- Page 7-8. In Section 7.4, replace the final paragraph with the following text:

  Under these circumstances, reboot the system using a conversational bootstrap procedure, and enter the following command at the SYSBOOT> prompt:

  ```
  SYSBOOT> SET STARTUP_P1 "MIN"
  ```

  This command initiates a minimum startup procedure as described in Chapter 4.

5.10 Image Activation, Search Lists, and Known Images

One of the steps involved in image activation uses the VAX Record Management Service (RMS) to open the specified image file. When the image to be activated is specified as a logical name, the file specification that is the translation of that logical name is accessed. RMS then opens the image by first attempting to locate the image on one of the known file lists. If the image is not known (that is, the lookup operation fails) then RMS has no other choice but to incur the overhead of locating and opening the image file on disk.

If the image specification includes a semicolon (;) or a period (.) to delimit the version number (whether or not an explicit version number is actually specified) the known file lookup by RMS is skipped. In that case, RMS will always incur the overhead of opening the image file on disk.

The precedence of the known file lookup over the normal file system access during image activation is extended when an image is being activated by way of a search list. For each element on the search list that does not include a file version delimiter, RMS executes a known file lookup. This continues until a lookup is successful or until the search list is exhausted. If the search list is exhausted, RMS then evaluates the entire search list from its beginning a second time in an effort to locate and open the image file on disk. Further information about locating files using search lists can be found in the *Guide to VAX/VMS File Applications.*

Because of this behavior, it is suggested that care be taken when defining a search list which contains specifications for images that are installed. Regardless of the order of the elements of the search list, the first image in that search list that is found to be installed will be the image selected for activation. That will occur even if there are preceding images in the search list that are not installed.

5.11 Install Utility

This section describes changes to the Install Utility.

5.11.1 Enhanced LIST/GLOBAL/FULL Command

The LIST/GLOBAL/FULL command of the Install Utility now displays the following additional information on global sections:

- Owner and protection
- Access control entries (ACEs) if an access control list (ACL) exists

5.11.2 New /SUMMARY Qualifier

Used with the INSTALL/GLOBAL command, the /SUMMARY qualifier displays a summary of global section and global page usage on the system for both local and shared memory global sections.

5.12 Installation

This section describes documentation changes to the *Guide to VAX/VMS Software Installation.*

5.12.1 Installation of Stand-Alone Backup on an alternate system root

This section corrects an error in Section 4.7.1 of the *Guide to VAX/VMS Software Installation.* The example given on page 4-26 should either include a space between the device and directory specifications, or omit the directory specification completely. The reason for this is that the STABACKIT procedure handles the specifications separately as parameters. In addition, if you omit the directory specification, SYSE is the default. The example should read as either of the following:

```
$ @SYS$UPDATE:STABACKIT SYS$SYSDEVICE: [SYSE.SYSEXE]
$ @SYS$UPDATE:STABACKIT SYS$SYSDEVICE:
```

5.12.2 VMSINSTAL Error Messages

The *Guide to VAX/VMS Software Installation* describes VMSINSTAL error messages in Section 5.6. The ACCOUNT error message on page 5-13 should be replaced with the following:

ACCOUNT installation creates an account named name

Explanation: The product being installed will create (or update) an account with the specified name. If the operation is successful, the user authorization file (SYSUAF) will be updated accordingly.

User Action: None.

The VMSINSTAL procedure for Version 4.2 of VAX/VMS includes the following new error messages, which complement the existing messages described in Section 5.6 of the *Guide to VAX/VMS Software Installation*:

- BADACC Unable to CREATE/UPDATE account named\bold).

  Explanation: The product being installed has tried either to create an already existing account or to modify a nonexistent account.

- NOACCMAN Account manipulation on alternate roots not supported.

  Explanation: The product being installed on an alternate root cannot update the user authorization file (SYSUAF).

- NOSUCHNODE Remote node *nodename* not known or unavailable

  Explanation: The product being installed is located on a remote node that is either not known or not available.

5.13 Logical Names To Be Defined in Executive Mode

In performing logical name lookups, certain VAX/VMS images and utilities such as LOGINOUT and MAIL bypass the user- and supervisor-mode portions of the system logical name table (LNM$SYSTEM_TABLE). DIGITAL therefore recommends that logical names for important system components (public disks and directories, for example,) be defined in executive mode, using the DCL command DEFINE/SYSTEM/EXECUTIVE.

5.14 Monitor Utility Command Procedure and Example Revisions

For Version 4.2, the following command files, including corrections and updates, are provided in the SYS$EXAMPLES directory:

- MONITOR.COM
- SUBMON.COM
- MONSUM.COM

You should replace any command files that you copied from the Version 4.0 *VAX/VMS Monitor Utility Reference Manual* with the versions provided in the Version 4.2 SYS$EXAMPLES directory.

Example MON-6, along with introductory text, should be replaced as follows:

Example MON-6 shows a typical VAXcluster multifile summary generated by the following command:

```
MONITOR/INPUT=(MOE.DAT;*,CURLEY.DAT;*,LARRY.DAT;*) MODES, PAGE -
 /SUMMARY /BY_NODE /NODISPLAY /BEGINNING="18:17" /ENDING="20:17"
```

Example MON-6 VAXcluster Multifile Summary

```
          +-----+         VAX/VMS Monitor Utility
          | AVE |         TIME IN PROCESSOR MODES
          +-----+            MULTI-FILE SUMMARY

           Node:       MOE                CURLEY (2)         LARRY
           From: 15-APR-1984 18:17  15-APR-1984 18:17  15-APR-1984 18:17      Row      Row      Row      Row
           To:   15-APR-1984 20:17  15-APR-1984 20:17  15-APR-1984 20:17      Sum  Average  Minimum  Maximum
Interrupt stack              6.51               0.50               6.25     13.2      4.4     0.50     6.51
Kernel mode                 25.73               0.42              12.43     38.5     12.8     0.42    25.73
Executive mode               9.46               0.95               1.81     12.2      4.0     0.95     9.46
Supervisor mode              1.97               0.00               0.16      2.1      0.7     0.00     1.97
User mode                   32.72               5.33              79.32    117.3     39.1     5.33    79.32
Compatibility mode           0.00               0.07               0.00      0.0      0.0     0.00     0.07
Idle time                   23.61              92.63               0.02    116.2     38.7     0.02    92.63

          +-----+         VAX/VMS Monitor Utility
          | AVE |        PAGE MANAGEMENT STATISTICS
          +-----+            MULTI-FILE SUMMARY

           Node:       MOE               CURLEY(2)          LARRY
           From: 15-APR-1984 18:17  15-APR-1984 18:17  15-APR-1984 18:17      Row      Row      Row      Row
           To:   15-APR-1984 20:17  15-APR-1984 20:17  15-APR-1984 20:17      Sum  Average  Minimum  Maximum
Page fault rate             36.73               8.81               0.49     46.0     15.3     0.49    36.73
Page read rate              14.28               4.71               0.00     19.0      6.3     0.00    14.28
Page read I/O rate           1.20               0.70               0.00      1.9      0.6     0.00     1.20
Page write rate              0.00               0.00               0.00      0.0      0.0     0.00     0.00
Page write I/O rate          0.00               0.00               0.00      0.0      0.0     0.00     0.00
Free list fault rate         8.60               1.40               0.24     10.2      3.4     0.24     8.60
Modified list fault rate     5.83               2.29               0.00      8.1      2.7     0.00     5.83
Demand zero fault rate      12.96               1.68               0.24     14.8      4.9     0.24    12.96
Global valid fault rate      8.10               2.69               0.00     10.8      3.6     0.00     8.10
Wrt in progress fault rate   0.00               0.00               0.00      0.0      0.0     0.00     0.00
System fault rate            4.92               0.53               0.18      5.6      1.8     0.18     4.92
Free list size            7586.30            8630.14            9665.06  25881.5   8627.1  7586.30  9665.06
Modified list size          87.69             324.07              32.12    443.8    147.9    32.12   324.07
```

5.15 Mount Utility

The following section contains additions and corrections to the Mount Utility documentation.

5.15.1 Change in Job-Wide Support of the Mount Utility

The documentation for the job-wide MOUNT support was omitted from the VAX/VMS Version 4.0 documentation. It should read as follows:

In VAX/VMS Version 4.0, any subprocess in the process tree can mount or dismount a volume for the job. When a subprocess mounts a volume (for the job) as a private volume, the master process of the job becomes the owner of this device. This provision is necessary because the subprocess may be deleted and the volume should remain privately mounted for this job.

5.15.2 Mount Utility Messages

The following is a list of new messages for the Mount Utility:

- DUALLOC dual allocation on volume 'n'

  Facility: MOUNT, MOUNT command or the SET VOLUME/REBUILD command

  Explanation: In performing the volume rebuild, the volume rebuild utility found that a logical block on relative volume 'n' was allocated to more than one file.

  User Action: Use the Verify Utility with the ANALYZE/DISK_STRUCTURE/NOREPAIR command on the volume. The Verify utility should report the following error:

  MULTALLOC, file ('file-id') 'file-name' multiply-allocated blocks VBN 'n' to 'n' LBN 'n' to 'n', RVN 'n'

  Examine all the multiply-allocated block messages to determine the files involved in each instance of multiple allocation. Then, without allowing other file activity, perform the following steps:

  1. Copy all but one file involved in each instance to a new version.
  2. Delete the version (of each file that was copied) that contains the multiple allocation. This action marks blocks free in the storage bit map that are not, in fact, free blocks.
  3. Reenter the Verify Utility with the /REPAIR qualifier to correct the storage bit map.
  4. Examine each file for corruption and reconstruct from back-up media as necessary.

- REBLDREQ, rebuild not performed; some free spaces unavailable; disk quota usage stale

  Facility: MOUNT, MOUNT command

  Explanation: A volume has been improperly dismounted (such as during a system crash), and subsequently mounted with the /NOREBUILD qualifier.

An improperly dismounted volume must be rebuilt to recover any caching limits that were enabled on the volume at the time of the dismount. The rebuild operation will recover any preallocated free space in the EXTENT cache, preallocated file numbers in the FILE_ID cache, and rebuilds the disk quota information on the volume.

User Action: Issue the SET VOLUME/REBUILD command to rebuild the volume.

- REDCACHE volume mounted with reduced cache size

  Facility: MOUNT, MOUNT command

  Explanation: There is insufficient dynamic memory to allocate the file system buffer caches from paged pool. The amount of paged pool used is controlled by SYSGEN parameters ACP_HDRCACHE, ACP_MAPCACHE, ACP_DIRCACHE, and ACP_DINDXCACHE.

  User Action: Possible actions are:

  1. If the reduced file system cache does not compromise the performance the file system, then nothing needs to be done.
  2. If the volume was mounted with the MOUNT/PROCESSOR command, remove the /PROCESSOR qualifier so that shared file system block cache will be used rather than creating a dedicated file system cache. Also, set the value of the SYSGEN parameter ACP_MULTIPLE to zero.
  3. Reboot the system, stopping at SYSBOOT, and increase the value of parameter PAGEDYN. Alternatively, reduce the values of the parameters ACP_HDRCACHE, ACP_MAPCACHE, ACP_DIRCACHE, and ACP_DINDXCACHE.

5.16 Restriction on Dual Ported Non-DSA Disks in a VAXcluster

Do not use SYSGEN to AUTOCONFIGURE or CONFIGURE a dual ported, non-DSA disk which is already available on the system via an MSCP server. Establishing a local connection to the disk when the remote path is already known will create two uncoordinated paths to the same disk. Use of these two paths will potentially corrupt files and data of any volume mounted on the drive.

In a VAXcluster, dual ported non-DSA disks (MASSBUS or UNIBUS) may be connected between two nodes of the cluster. These disks may also be made available to the rest of the cluster using the MSCP server on either or both of the hosts to which a disk is connected.

During a normal bootstrap operation, the local path to the disk is discovered before the MSCP server path from the other host is found. If the documented restrictions regarding device naming conventions, allocation class, and the /DUAL_PORT characteristic have been observed, then the disk class driver correctly interprets the two paths as belonging to the same physical disk drive.

If the local path to the disk is not found during the bootstrap, then the MSCP server path from the other host will be the only available access to the drive. The local path will not be found during a boot if any of the following conditions exist:

1. The port select switch for the drive is not enabled for this host.

2 The disk, cable, or adapter hardware for the local path is broken.

3 There is sufficient activity on the other port to "mask" the existence of the port.

4 The system is booted in such a way that the SYSGEN AUTOCONFIGURE ALL in the STARTUP.COM procedure was not executed.

Use of the disk is still possible through the MSCP server path.

Once the configuration of the disk has reached this state, it is important NOT to add the local path back into the system I/O database. Since there is no automatic method for this to occur within VAX/VMS, the only possible way that this could occur would be to use the SYSGEN program to AUTOCONFIGURE or CONFIGURE the device. SYSGEN is currently not able to detect the presence of the disk's MSCP path, and will incorrectly build a second set of data structures to describe it. Subsequent events could lead to incompatible and uncoordinated file operations which might corrupt the volume.

In order to recover the local path to the disk, it is necessary to reboot the system connected to that local path.

Note that if the disk is NOT dual ported or is NEVER MSCP served on the remote host, this restriction does not apply.

5.17 SET HOST/HSC Facility

The SET HOST/HSC facility allows you to use most HSC50 commands and utilities from a remote terminal. For a description of HSC50 commands and utilities, see the *HSC50 User Guide*.

To activate the facility, you must first install and load FYDRIVER, the Diagnostic and Utilities Protocol (DUP) driver associated with the CI. The preferred method is to add the following command lines to your site-dependent startup procedure, SYS$MANAGER:SYSTARTUP.COM:

```
$ RUN SYS$SYSTEM:SYSGEN
CONNECT FYA0/NOADAPTER
```

After SYSTARTUP.COM executes, you can use the DCL command SET HOST/HSC to connect to an HSC50 by way of the CI bus. (For a complete description of this command, refer to the *VAX/VMS DCL Dictionary*.)

Once connection is made, you may perform operations as if you are attached to the local HSC terminal, with the exception of access to the Octal Debugging Tool (ODT) and off-line diagnostics.

Note: This version of VAX/VMS contains support for remote terminal access to an HSC50 using the Diagnostic and Utilities Protocol (DUP). A future version of the HSC50 software will support this protocol.

5.18 Configuring Devices Connected to an HSC50

During system startup, the site-independent startup command procedure, STARTUP.COM, invokes SYCONFIG.COM to check user-specified configuration requirements. When SYCONFIG.COM completes, control is returned to STARTUP.COM. A section of STARTUP.COM called CONFIGURE then runs a program that creates a detached process to detect any devices connected to an HSC50, loads their drivers, and makes the devices known to the system.

Note, however, that if you have set STARTUP$AUTOCONFIGURE to zero in SYCONFIG.COM, the CONFIGURE section of STARTUP.COM will not execute. To ensure proper configuration of devices connected to an HSC50 in this case, you must add the following command line to your site-independent command procedure, SYSTARTUP.COM:

```
$ @SYS$SYSTEM:STARTUP CONFIGURE
```

This command manually invokes the CONFIGURE section of STARTUP.COM.

DIGITAL is currently considering modifications to system startup procedures that will eliminate the problem of configuring devices connected to the HSC50.

5.19 System Generation Utility (SYSGEN)

This section describes additions and corrections to the System Generation Utility (SYSGEN).

5.19.1 CONNECT CONSOLE Command: /Remote Qualifier

In VAX/VMS Version 4.2, the /REMOTE qualifier has been added to CONNECT CONSOLE. This qualifier enables a remote diagnostic port for a second console or terminal connected to a VAX 8600.

5.19.2 UDABURSTRATE Parameter

The UDABURSTRATE parameter is configuration and workload dependent. Alteration of the default value can have serious side effects. Consult your DIGITAL Field Service representative before changing the default value of this parameter.

5.20 User-Created Quorum File Problem

The cluster quorum file QUORUM.DAT is automatically created by the cluster connection manager when the system is booted for the first time with the sysgen parameter DISK_QUORUM specified. The connection manager creates the quorum file with the appropriate attributes and supplies the initial template entry.

Therefore, one should not attempt to manually create a quorum file. Doing so may cause the home block of the quorum disk to be corrupted.

5.21 VAX 8600

The following describes the enhancements and corrections to the VAX 8600 System.

5.21.1 Change in VAX 8600 Console Behavior

Beginning with release 4.0 of the console RL02, version 7.0 of the console software, typing CTRL/P no longer automatically halts the VAX 8600; you must enter the console HALT command if you wish to halt the CPU. The impact of this change on VMS is that going to console command mode during timesharing no longer results in the breaking of cluster connections and the detaching of terminals of processes connected via LAT lines.

5.21.2 VAX 8600 Power Failure Corrected

In VAX/VMS Version 4.1, the CI port may not fully recover from a power failure on an VAX 8600 CPU. This problem has been corrected in VAX/VMS Version 4.2.

5.21.3 Unexpected IPL 23 Interrupts Booting a VAX 8600

VAX/VMS will sometimes get an UNXINTEXC (unexpected interrupt or exception) bugcheck when booting on a VAX 8600. The bugcheck is due to an interrupt at IPL 23 (console RL02) before the system control block (SCB) has been set up to handle it and before the system has started any user processes. To positively identify such a crash, examine the resultant dump to see if the contents of the interrupt stack resemble the following:

```
        80604BF0  80004680      ERL$UNEXP+004
        80604BF4  04170000
SP =>   80604BF8  8000A6CD      MPH$SCHED+003
        80604BFC  04080000
```

While such an interrupt would normally be logged and dismissed, systems will crash if the SYSGEN parameter BUGCHECKFATAL is enabled (set to 1). Such a crash will overwrite your system dump file before a previous dump can be copied. Until this problem is corrected, you may leave BUGCHECKFATAL disabled (set to zero). The UNXINTEXC bugcheck then will not cause the system to reboot. You can also work around the problem by ensuring that the console RL02 is not mounted when VMS is brought down.

5.21.4 Copying Machine State Error Snapshots on a VAX 8600

A restriction involving the location of the ERRSNAP.COM file in VAX/VMS Version 4.1A has been removed; ERRSNAP.COM can now be found in SYS$COMMON:[SYSERR] of a cluster common system disk.

5.21.5 Running User Environment Test Package (UETP) with Multiple DR780s

Running the User Environment Test Package (UETP) with multiple DR780s can result in an SS$_DEVACTIVE error in the DR780 microcode loader process, XFLOADER. UETP starts up a test process, invoking XFLOADER for each DR780. Each XFLOADER process then tries to simultaneously load microcode in the same DR780s. The simultaneous XFLOADER processes can also attempt to write to the same XFLDR_ERROR.LOG file as their SYS$OUTPUT, which can result in an RMS$_FLK error. This problem will be corrected in a future release of VAX/VMS.

6 Notes for the Application Programmer

This chapter presents additions and restrictions in Version 4.2 of VAX/VMS that are of interest to application programmers and others who use RTL and utility routines and system services. This information is not documented elsewhere in the Version 4.2 documentation kit.

6.1 PL/I PRINT FILE Format Change

In previous versions of VAX/VMS, PL/I generated an extra linefeed immediately following a PAGE directive for PRINT format files. This extra line has been removed for Version 4.2. Applications that require the old behavior can approximate it using a PUT SKIP command when the ENDPAGE condition is raised, or when a PAGE is explicitly output. While VAX/VMS recommends that /NOFEED be used for printing formatted files, this change should allow newly generated PL/I PRINT files to be printed on forms with the same number of lines per page as those of the print file using /FEED.

Note that the effect of this change may show up in different ways depending on the printer type. New printers and terminal devices will simply print everything one line higher on the page. Older line printers may ignore some linefeeds at the top of page so that this change will only show up when the first line of text is printed part way down the page.

6.2 Run-Time Library LIB$FIND_IMAGE_SYMBOL Routine

The library procedure LIB$FIND_IMAGE_SYMBOL allows a shareable image to be dynamically mapped into the address of a running program. The procedure returns the address of a global entry point within the image.

The documentation for this procedure (page RTL-109 of the *VAX/VMS Run-Time Library Routines Reference Manual*) states that the image should be specified by a name only, not a complete file specification. If a shareable image in a directory other than SYS$SHARE: needs to be specified, a logical name must be defined to point to the image file. This logical name is then passed to LIB$FIND_IMAGE_SYMBOL.

Although this behavior was clearly documented, the procedure was not enforcing the restriction in Versions 4.0 and 4.1 of VMS. In Version 4.2, the restriction is being enforced. This may cause programs that previously worked to fail with an error status of SS$_IVLOGNAM returned by LIB$FIND_IMAGE_SYMBOL. Programs that did not conform to the documented behavior must be changed in order to continue working under Version 4.2.

This change has also affected some layered products that were using the procedure. In particular, the documentation for the VAX LISP CALL-OUT facility (page 6-14 of the *VAX LISP User's Guide*) contains example code segments that will not work under Version 4.2. Programs that mimic these examples must either refer exclusively to shareable images in SYS$SHARE: or use logical names as in the following example.

```
        (DEFINE-EXTERNAL-ROUTINE
            (NUMBERS :IMAGE-NAME "DBA2:[SMITH]EXAMPLE"
                     :RESULT INTEGER)
               X Y)
must be changed to something like
        (DEFINE-EXTERNAL-ROUTINE
             (NUMBERS :IMAGE-NAME "EXAMPLE_IMAGE"
                      :RESULT INTEGER)
               X Y)
```

where the logical name EXAMPLE_IMAGE is equivalent to the file specification DBA2:[SMITH]EXAMPLE. Note that this logical name can be defined externally with DCL or inside the program calling LIB$FIND_IMAGE_SYMBOL.

6.3 Run-Time Library LIB$GET_VM and LIB$FREE_VM Routines

The Run-Time Library (RTL) contains a new implementation of the heap storage management procedures LIB$GET_VM and LIB$FREE_VM. See the *Run-Time Library Routines Reference Manual* for a complete description of these procedures.

The new heap storage management procedures are upwardly compatible with previous versions. However, the new version may expose latent errors in existing programs:

- LIB$GET_VM and LIB$FREE_VM each require two arguments; previous versions ignored any additional arguments. The new versions have a third optional argument; if a third argument is supplied, it must be a zone-id value.
- You can only call LIB$FREE_VM to free memory that was allocated by a previous call to LIB$GET_VM. The new version checks this condition more thoroughly and may return an error status where the older version did not.
- You cannot call LIB$FREE_VM to free a dynamic string that was allocated by the Run-Time Library dynamic string package (STR$ procedures). You must call one of the string package procedures to free a dynamic string (for example, LIB$SFREEN_DD or STR$FREE1_DX).
- You should not try to free part of a block that was allocated by LIB$GET_VM or to combine several blocks into one larger block and free that larger block in a single call to LIB$FREE_VM. The new version checks these conditions more thoroughly.
- Version 4.0 uses different data structures to manage free memory. If a program incorrectly writes outside the bounds of its allocated memory and corrupts these data structures, the program may behave differently with the new version.

6.3.1 Transportability Problem with Based Images Linked Against MTHRTL

Based images that are linked against the MTHRTL math library shareable image and run on a system where the UVMTHRTL image is the default will not activate properly. This is because the UVMTHRTL image is slightly larger than the MTHRTL image, and it will not fit into the based slot allocated for it at link time.

DIGITAL recommends that you do not create based images. One reason is that this can cause fragmentation of the virtual address space for the process. Another and perhaps more significant reason is the problem of the image being linked against Run-Time Library (RTL) shareable images.

If you link based images with RTL shareable images, you will be forced to relink your application with each new version of VAX/VMS. This is because changes and additions are made to RTL shareable images in each new version of VAX/VMS, changing the size of RTL images. In turn, you must relink in order to update base addresses for these images in your based image for it to activate properly.

One of the biggest advantages of the RTL being packaged in shareable images is that applications need not be relinked when corrections are made to the RTL code. This advantage is completely defeated with the use of based images; as outlined above, a based image must be relinked on every VAX/VMS release.

Assuming, however, that one has to use a based image and that RTL support is required as well, there are two ways you can work around this problem in order to make your image activate properly. They are as follows:

1 LINK against UVMTHRTL and run against normal MTHRTL for your system

  This ensures that a based slot large enough for the largest possible math library image is allocated in your based image at link time.

2 LINK/NOSYSSHR

  This method completely eliminates the references to the MTHRTL shareable image in the first place.

Note that both of these workarounds still require that you relink with every VAX/VMS update in order to ensure proper activation and/or up-to-date versions of RTL procedures in your image. This is nonetheless something you would have to do anyway with a based image.

DIGITAL intends to integrate MTHRTL and UVMTHRTL back into a single image in a future release of VAX/VMS, eliminating this transportability problem. Note, however, that based images will continue to need to be relinked on every VAX/VMS release in order to ensure up-to-date base addresses of RTL shareable images or RTL procedures, depending on how the based image is linked.

6.3.2 Downwards Transportability for Images Linked with the Run-Time Library

User applications linked against Run-Time Library (RTL) facilities that provide new images, or images with changed global section match identification (GSMATCH) values for VAX/VMS Version 4.2, will not run on earlier versions of VAX/VMS. Note, however, that an image linked on Version 4.0 or Version 4.1 will continue to run on Version 4.2. Specifically, the RTL images with changed GSMATCH are as follows:

- LIBRTL
- MTHRTL
- UVMTHRTL
- SMGSHR
- PASRTL
- RPGRTL

GSMATCH values for these images have changed because new RTL procedures are being provided in these images to support upgraded versions of VAX/VMS languages and other layered products.

The GSMATCH value is set to protect you from attempting to activate a shareable image that does not contain a procedure you may need (for example, an RTL image on an earlier version of VAX/VMS).

New RTL images in Version 4.2 of VAX/VMS are as follows:

- ADARTL
- SCNRTL
- VAXCRTL
- VAXCRTLG

Applications affected by these changes include those that use LIBRTL, the Math Library, and Screen Management, or are written (all or in part) in PASCAL or RPG II. Programs written in Ada and SCAN will also not run on earlier versions of VAX/VMS because the RTL support for these languages is new in Version 4.2. Run-Time Library support for VAX C is also newly bundled with VAX/VMS in this version.

There is another risk that you undertake in transporting downwards; your program may not produce expected results. Corrections and modifications are made to the RTL in nearly every VAX/VMS release; you may code an application that requires a correction that is in the version of VAX/VMS you develop it on, but is not in earlier versions. In this case, the application will run on systems with the VAX/VMS version that you developed it on, but will fail on systems with earlier versions of VAX/VMS, regardless of GSMATCH.

Nonetheless, DIGITAL plans to be able to add new procedures to the RTL and still allow for downwards transportability in the future. In the meantime, you must link images on the lowest version of VAX/VMS that you expect them to run on. That is to say, if you wish to run an image on Version 4.0, it should be linked on Version 4.0 or an earlier version. An image linked on Version 4.1 or Version 4.2 will not necessarily run on Version 4.0 as expected.

6.4 Run-Time Library Support of VAX BASIC

The following problems have been corrected in the VAX/VMS Run-Time Library support for VAX BASIC:

- Previously, the user ran out of FILLM quota when opening and closing many remote files. This problem has been corrected.
- When the memory associated with a packed decimal array was being freed, the amount of storage allocated was incorrectly computed. This caused random virtual memory corruption problems. This problem has been fixed.
- Packed decimal arrays that were dimensioned at run time were not properly initialized to zero. This problem has been resolved.
- When the user program jumps from an external function into a DEF*, VAX BASIC previously signalled the error PROLOSSOR. VAX BASIC now signals the error FNEWITFUN, which directs the user to check the program logic.
- BASIC previously signalled the error PROLOSSOR when a user program jumped to a major procedure from a DEF* within a DEF. The error is now correctly reported as ILLEXIDEF and the user is directed to examine the program logic.
- When 11 fields (separated by commas) were printed on a terminal with NOMARGIN set, the eleventh field was incorrectly spaced. This problem has been fixed.
- The PRINT USING command formerly signalled the error OTS$_ FATINTERR when the format string contained an odd number of underscores and no valid format item. The error PRIUSIFOR is now signalled, which directs the user to change the PRINT USING format string.
- Previously, when an error occurred as files were being closed during image exit, the user could inadvertently regain control. In addition, the first occurrence of an error could cause an immediate image exit. Errors are now handled properly and the remaining files are closed.
- Previously descriptors sometimes became invalid when the user interrupted a REMAP statement on an array. REMAP is now atomic.
- A user could previously interrupt the routine that allocates a run-time dimensioned array. This sometimes caused virtual memory to be freed but not marked invalid. The user could then access the virtual memory which could have been pointing at something else. This problem has been resolved.
- The PRINT USING command previously truncated packed decimal numbers. The PRINT USING command now rounds packed decimal numbers.
- When VAL returns a single-precision number, it now rounds the double-precision number it calculates.

6.5 Sort/Merge Utility Routines

The following is a description of the changes made to the Sort/Merge Utility routines for Version 4.2.

- Previously, SOR$SORT_MERGE returned the error ENDDIAGS if errors were encountered. If errors were signaled, this was a warning message. If errors were not returned, ENDIAGS had the severity of the worst error encountered. In VAX/VMS Version 4.2, if errors are not being signaled, SOR$SORT_MERGE will return the worst error encountered.
- The user context longword specified in a call to SOR$BEGIN_SORT and SOR$BEGIN_MERGE was incorrectly passed to the User_Compare, User_Equal, and User_Input routines. Instead, the address of the longword was passed. This problem has been corrected.
- Some packed decimal numbers were incorrectly converted to the appropriate floating-point number. This problem has been corrected.

6.6 Symbolic Debugger

This section notes any changes made for Version 4.2 that are incompatible with Version 4.0 or earlier versions. All such changes concern the display of various windows in screen mode.

6.6.1 Register Windows

In VAX/VMS Version 4.0, the register display occupied five lines of the screen. Special windows, named R1, R2, R3, R12, and R23, were defined to accommodate a display of this size.

For VAX/VMS Version 4.2, the register display has been reduced to occupy just four lines. This was done by making more efficient use of space and also by removing the translation of R0 as an error message. (The translation of R0 can be obtained when needed by doing EXAMINE/CONDITION R0.)

Now that the register display occupies four lines, it fits in the standard windows Q1, Q2, Q3, and Q4 (one quarter of the screen). So the special purpose windows R1, R2, and R3 have been discontinued. If you refer to them in any command files, you should change your command file to use one of the "Q" (quarter) displays instead.

For example, if your old DBG$INIT file had the following:

```
DISPLAY REG AT R1
DISPLAY OUT AT R23
```

your new DBG$INIT file should have:

```
DISPLAY REG AT Q1
DISPLAY OUT AT Q234
```

6.6.2 MACRO defaults

In VAX/VMS Version 4.0, the screen default for language MACRO was to have the register display (REG) at the top of the screen, and the output display (OUT) in the rest of the screen. In Version 4.2, because of the addition of the assembly language instruction display (INST), this default has been changed to display INST in the top half of the screen (H1) and OUT in the bottom half (H2). This change makes the MACRO default analogous to that used with the supported high-level languages, namely, SRC in H1 and OUT in H2.

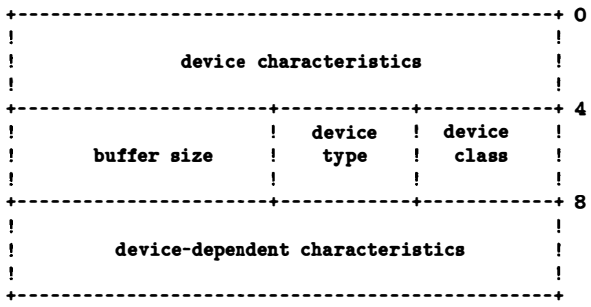
If you want to display the registers for MACRO, you now need to specify that in a DISPLAY command. For example, the following commands place REG in the top quarter (Q1) and INST in the second quarter (Q2):

```
DBG> DISPLAY REG AT Q1
DBG> DISPLAY INST AT Q2
```

6.7 System Services: $GETCHN and $GETDEV

VAX/VMS no longer supports the $GETCHN and $GETDEV system services, which were used to obtain information on the devices described in the *VAX/VMS I/O Reference Volume*.

These services returned device/channel-specific information in the following general form:



Users can obtain the same information by using the Get Device/Volume Information ($GETDVI) system service (see the *VAX/VMS System Services Reference Manual* in the *VAX/VMS System Routines Reference Volume*).

$GETDVI returns information by way of item codes, such as DVI$_DEVCHAR, DVI$_DEVDEPEND, DVI$_DEVBUFSIZ, DVI$_DEVTYPE and DVI$_DEVCLASS. The following table shows some of the set of $GETDVI item codes needed to obtain the same information returned by $GETCHN and $GETDEV.

DVI$_DEVCHAR	-	device characteristics
DVI$_DEVCLASS	-	device class
DVI$_DEVTYPE	-	device type
DVI$_DEVBUFSIZ	-	buffer size
DVI$_DEVDEPEND	-	device-dependent characteristics

The specific information returned depends on the particular device; not all devices return all item codes. For example, for disk devices $GETDVI returns information on cylinders, tracks, and sectors.

6.8 "VMS Usage" Added to Argument Descriptions

Each argument's description in the *VAX/VMS System Services Reference Manual* and the *VAX/VMS Run-Time Library Routines Reference Manual* includes the "VMS usage" of that argument.

The *Introduction to VAX/VMS System Routines* is not being revised for VAX/VMS Version 4.2, and thus its description of the documentation format for system routines does not include information on the VMS usages.

The *VAX Record Management Services Reference Manual* and the *VAX/VMS Utility Routines Reference Manual* are not being revised for Version 4.2; the next time they are revised, they will include information on the VMS usage, type, access, and mechanism for each argument.

7 Notes for the System Programmer

This chapter presents additions and restrictions in Version 4.2 of VAX/VMS that are of interest to system programmers and driver writers. This information is not documented elsewhere in the Version 4.2 documentation kit.

7.1 Drivers

This section contains device driver information.

7.1.1 Corrections to the *VAX/VMS I/O User's Guide*

The following is a list of corrections to existing documentation of the *VAX/VMS I/O User's Reference Manual: Part I* and *VAX/VMS I/O User's Reference Manual: Part II*:

- The description of Figure 8.2 in Section 8.2.3.1 on the *VAX/VMS I/O User's Reference Manual: Part I.* is changed.

  Figure 8-2 is a flow chart that shows a typical signal sequence for a terminal operation in this mode. The flow chart shows what states the modem transition code goes through to detect different types of transitions in modem state. These transitions allow the driver to detect loss of lines that have been idle for several minutes. These states do not affect the ability of the system to transmit characters.

- The description of the parameter ID NMA$C_PCLI_DES in Table 6-5 of the *VAX/VMS I/O User's Reference Manual: Part II* is changed.

  NMA$C_PCLI_DES is the shared protocol destination address. It is passed as a counted string that consists of a modifier word followed by a 6-byte (48-bit) destination physical address. NMA$C_PCLI_DES only has meaning when protocol access (NMA$C_PCLI_ACC) is defined as shared with destination mode (NMA$C_ACC_ LIM).

  Section 6.3.3.2 provides a description of protocol type sharing.

- The description of Figure 8-2 was omitted from Chapter 8 of the *I/O Users Guide: Part I* in VAX/VMS Version 4.2. It should read as follows:

  Figure 8 is a flow chart that shows a typical signal sequence for a terminal operation in this mode. The flow chart shows the states that the modem transition code goes through to detect different types of transitions in modem state. These transitions allow the driver to detect loss of lines that have been idle for several minutes. These states do not affect the ability of the system to transmit characters.

7.1.2 CI Port Drivers

VAX/VMS Version 4.2 contains a new image of the CI port driver, PADRIVER.EXE.

7.1.2.1 VAX 8600 Power Failure Corrected

In VAX/VMS Version 4.1, the CI port may not fully recover from a power failure on an VAX 8600 CPU. This problem has been corrected in VAX/VMS Version 4.2.

7.1.2.2 Supported CI780 Microcode

All sites should have either Version 3.0 or Version 4.0 of the CI780 microcode. You may identify your current microcode version by executing the SHOW CLUSTER/CONTINUOUS DCL command and then typing the ADD RP_REVIS subcommand. The first half of the field is the RAM version and the second half is the PROM version. For Version 3.0 microcode, this field will contain 30003 (hexadecimal) and for Version 4.0, this field will contain 40003 (hexadecimal).

You will need Version 4.0 of the microcode only if your error logs show frequent CI port shutdowns because of "Miscellaneous Error #12" or "Miscellaneous Error #13". On the operator's console, these shutdowns will be reported as an "Unexpected Interrupt" for VAX/VMS Version 4.0 or 4.1 systems and as "Port Error Bit(s) Set" for VAX/VMS Version 4.2.

The port driver will display the following message for sites containing Version 2.0 microcode:

```
%PAA0, - CI port ucode not at current rev level.
                                        PROM/RAM rev is 0002/0002
```

The message is purely informational and the system will continue to run under a light load. Customers, however, should upgrade their CI780 as quickly as possible because Version 2.0 of the microcode contains an error that can cause system failure.

Previous versions of the CI port driver displayed the microcode revision incorrectly on the operator's console. The correct order of the fields is "PROM/RAM" rather than "RAM/PROM".

7.1.2.3 Microcode Verification

Previously, the CI port driver failed to verify the proper loading of the CI microcode. Immediately after loading the microcode in the CI port, the CI port driver will now read back the loaded image and report any errors with the following message:

```
%PAA0, Micro-code Verification Error
```

The port driver will attempt to recover from this error by restarting the port initialization sequence. When this error appears, you should ask field service to run the CI port diagnostics.

7.1.2.4 The CI DECNET Line

When using DECNET over the CI, keep the default SYSGEN parameter SCSMAXDG and the default DECNET buffer size for the CI line. A failure to use these can lead to a system crash.

7.1.3 Source Code Changes of the DR11–W Driver

The following is a description of the changes made to the source code of the DR11–W driver. An on-line copy of the corrected code is available in SYS$EXAMPLES:XADRIVER.MAR.

In the past, because the definitions of the IO$M_INHERLOG and IO$M_RESET bits were identical, the VAX/VMS error-logging routines incorrectly referred to the IO$M_RESET bit to determine whether they should implement error logging. To correct this problem, the location of IO$M_RESET was redefined. To maintain compatibility with existing programs, the driver's read/write routine now includes code that manually moves the IO$M_RESET bit to its new location:

```
XA_READ_WRITE:
        BBCC    #IO$V_INHERLOG,IRP$W_FUNC(R3),1$
                                        ;Branch if old reset bit not set
        BISW    #IO$M_RESET,IRP$W_FUNC(R3)
                                        ;Set new reset bit
1$:     BLBC    P2(AP), 10$             ;Branch if transfer count even
2$:     MOVZWL  #SS$_BADPARAM,R0        ;Set error status code
```

The Cancel I/O routine contains a correction that allows the user to avoid a synchronization problem that previously occurred when completion and cancellation of an I/O operation coincided:

```
XA_CANCEL:
    .
    .
    .
20$:    DSBINT  UCB$B_DIPL(R5)          ;Lock out device interrupts
        BBC     #UCB$V_INT,-            ;Branch if I/O not in progress
                UCB$W_STS(R5),30$
        JSB     G^IOC$CANCELIO          ;Check if transfer going
    .
    .
    .
```

7.1.4 UQ Port Driver Support

The UQ port driver, PUDRIVER, that supports UDA50s, RC25s, and TU81s on VAX–11/750s, will check the Engineering Change Order (ECO) revision level of VAX–11/750s to determine if a particular ECO pertaining to the Unibus Adapter (UBA) has been installed. This ECO corrects a problem that affected data transfers from or to unaligned buffers using buffered data paths.

The effect of this UBA problem was to corrupt user data in certain isolated instances. Before this ECO became available, the PUDRIVER prevented the UBA problem from affecting user data integrity by always using the direct data path for unaligned data transfers. The Version 4.2 PUDRIVER will continue to use the direct data path for unaligned data transfers, but only for systems whose ECO revision level is below 70 hexadecimal. If the ECO revision level is 70 or greater, the PUDRIVER will attempt to use a buffered datapath for all transfers.

7.2 Time of Day Register Documentation

The documentation for the Time of Day Register (TODR) was omitted from VAX/VMS Version 4.0. It should read as follows:

The following internal processor registers (IPRs) are no longer common to all VAX processors. Their definitions have been removed from $PRDEF:

- NICR—Interval Clock Next Interval Register
- ICR—Interval Clock Interval Count Register
- TODR—Time of Day Register
- ACCS—Accelerator Control Status Register
- ACCR—Accelerator Reserved
- PME—Performance Monitor Enable

New CPU-specific processor register definition macros have been added to LIB.MLB to define the CPU-specific IPRs. The macro names have the format $PRxxxDEF, where xxx is the number associated with the processor (for example, $PR780DEF will define PR780$_ACCS).

The only legitimate references to these registers are in CPU-dependent code. These references must use the new CPU-dependent IPR definitions.

Note, however, that time-wait loops must NEVER directly reference the clocks. They MUST use a time-wait macro that is CPU independent. A new, CPU-independent time-wait macro called TIMEDWAIT has been added to LIB.MLB; this should eliminate any need for hand-code, time-wait loops.

There should no longer be any references to PR$_ICR or PR$_TODR to do time-wait loops. TIMEDWAIT allows for up to six special-purpose instructions to be placed in its timing loop. However, the loop timing is based on having one BITx and one conditional branch instruction embedded within the loop. Therefore, if you have a loop with no embedded instructions, you may want to adjust the TIME argument accordingly. A good rule of thumb is to add 25% to the TIME argument if the loop has no embedded instructions.

To reference PR$_TODR for logging purposes, use EXE$READ_TODR and EXE$WRITE_TODR. These two new loadable, CPU-dependent routines have been added for code that must reference this type of value.

7.3 VAX MACRO: Documentation Change

This section contains documentation changes to the *MACRO Instruction Set and Reference Manual.*

7.3.1 Cyclic Redundancy Check (CRC)

The following step should be included in the Cyclic Redundancy Check (CRC) instruction on page 9-138 of the *VAX MACRO and Instruction Set Reference Manual.*

```
Upon completion of the CRC instruction,
registers  R0, R1, R2  and R3 are left as follows:

                R0 = result of CRC
                R1 = 0
                R2 = 0
                R3 = address one byte beyond end of source string
```

Index

A

B

C

D

E

F

H

I

L

M

N

P

Q

R

S

T

U

V

VAX/VMS Release Notes
Version 4.2
AA–Z106A–TE

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

Do Not Tear - Fold Here and Tape

digital

No Postage Necessary if Mailed in the United States

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO.33 MAYNARD MASS.

POSTAGE WILL BE PAID BY ADDRESSEE

SSG PUBLICATIONS ZK1-3/J35
DIGITAL EQUIPMENT CORPORATION
110 SPIT BROOK ROAD
NASHUA, NEW HAMPSHIRE 03062-2698

Do Not Tear - Fold Here

Cut Along Dotted Line