MicroVMS Release Notes
Version 4.4

Order Number: AA–HL72A–TN

April 1986

This manual contains release information on the Version 4.4 software and documentation. This information is not included elsewhere in the MicroVMS Version 4.4 documentation kit.

Revison/Update Information:	This new manual supersedes previous MicroVMS Release Notes.
Operating System and Version:	MicroVMS Version 4.4
Software Version:	MicroVMS Version 4.4

digital equipment corporation
maynard, massachusetts

HOW TO ORDER ADDITIONAL DOCUMENTATION
DIRECT MAIL ORDERS

Contents

Chapter 3   Problems, Restrictions, and Notes

Index

Examples

Preface

This manual contains a summary of all Version 4.4 changes to MicroVMS software and previously released MicroVMS documentation. It does not contain any information about the MicroVMS upgrade procedure. See the appropriate MicroVMS installation manual for a description of the Version 4.4 upgrade procedure.

This manual supersedes all release notes documentation to previous versions of the MicroVMS operating system. It includes, or updates, any previous release notes that are still pertinent to the Version 4.4 release.

References are made to the MicroVMS documentation wherever appropriate.

In some cases, references are made to manuals in the VAX/VMS document set. The MicroVMS documentation, because of its condensed nature, does not describe some system features; release notes about these features refer to the appropriate VAX/VMS documentation. Moreover, many MicroVMS users have copies of VAX/VMS documentation available; to simplify the transition to Version 4.4 for these users, we have included corrections and notes to VAX/VMS documentation describing features also available with MicroVMS.

You must already be running MicroVMS Version 4.2 or later to be able to upgrade your system to Version 4.4. The MicroVMS Version 4.4 upgrade procedure will not work properly if your system is presently running at a version less than Version 4.2.

If you are running a MicroVMS Version that is prior to Version 4.2, you **must install** the Version 4.4 software on your system.

Refer to the appropriate MicroVMS installation manual for instructions about the installation and upgrade procedures.

Intended Audience

This manual is intended for all system users. Please read this manual before you install or use MicroVMS Version 4.4.

Structure of This Document

This manual consists of three major sections:

- Chapter 1 describes the mandatory update procedure for Version 4.4.

- Chapter 2 provides a brief summary of each new and changed feature.

- Chapter 3 contains information about problems, restrictions, and notes to the published documentation.

Conventions Used in This Document

The following conventions are observed in this manual:

Convention	Meaning
RET	A symbol with a one- to six-character abbreviation indicates that you press a key on the terminal, for example, RET .
CTRL/x	The phrase CTRL/x indicates that you must press the key labeled CTRL while you simultaneously press another key, for example, CTRL/C, CTRL/Y, CTRL/O.
$ SHOW TIME 05-JUN-1985 11:55:22	Command examples show all output lines or prompting characters that the system prints or displays in black letters. All user-entered commands are shown in red letters.
$ TYPE MYFILE.DAT . . .	Vertical series of periods, or ellipsis, mean either that not all the data that the system would display in response to the particular command is shown or that not all the data a user would enter is shown.
file-spec, . . .	Horizontal ellipsis indicates that additional parameters, values, or information can be entered.
[logical-name]	Square brackets indicate that the enclosed item is optional. (Square brackets are not, however, optional in the syntax of a directory name in a file specification or in the syntax of a substring specification in an assignment statement.)
quotation marks apostrophes	The term quotation marks is used to refer to double quotation marks ("). The term apostrophe (') is used to refer to a single quotation mark.

Chapter 1

Installing the Mandatory Update

After you have installed all MicroVMS options and the DECnet layered product, but before you install other layered products, you must install the mandatory update. The MicroVMS mandatory update is distributed on either one diskette or one tape cartridge and is installed using the VMSINSTAL command procedure.

Take the following steps to install the mandatory update.

1. **Start up the system if it is not already running.** See Section 2.2.4 of *Installing or Upgrading MicroVMS From Diskettes* or *Installing or Upgrading MicroVMS From a Tape Cartridge* for instructions.

2. **Log in to the system manager's account, SYSTEM, at the console terminal.**

3. **Notify users to log out.** Make sure that nobody but you is logged in to the system (notify any other users that they must log out), and then prevent users from logging in to the system by entering the following DCL command:

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

4. **Shut down the network if it is running.** Invoke NCP to shut down the network as shown in the following example:

   ```
   $ RUN SYS$SYSTEM:NCP
   NCP> SET EXECUTOR STATE OFF
   NCP> EXIT
   ```

5. **Stop all batch queues** by entering the following DCL command once for each batch queue that is set up for your system:

   ```
   $ STOP/QUEUE/NEXT queue-name
   ```

   where **queue-name** is the name of a queue that is set up on your system. The /NEXT qualifier allows the current job to complete before stopping the queue.

6. **Invoke the VMSINSTAL command procedure.**

   - If you are installing the mandatory update from a diskette, type the following command and press RETURN:

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

- If you are installing the mandatory update from a magnetic tape cartridge, type the following command and press RETURN.

  `$ @SYS$UPDATE:VMSINSTAL {insert-name-here} $TAPE1:`

7. **Reply to VMSINSTAL prompts.** You will be prompted as follows:

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

   Press RETURN if your backups are current. If not, type N and press RETURN to exit from VMSINSTAL; then back up the fixed disk (see Section 3.2.1.2) of *Installing or Upgrading MicroVMS From Diskettes* or *Installing or Upgrading MicroVMS From a Tape Cartridge*) and restart the installation procedure.

8. **Insert the first volume into the drive.**

   - If you are installing the mandatory update from a diskette, the system will prompt with the following message and question:

     `Please mount the first volume of the set on $FLOPPY1:`
     `Are you ready?`

     Insert the diskette containing the mandatory update in drive 1. Then type the letter Y (uppercase or lowercase) and press RETURN.

   - If you are installing the mandatory update from a tape cartridge, the system will prompt with the following message and question:

     `Please mount the first volume of the set on $TAPE1:`
     `Are you ready?`

     Insert the tape cartridge containing the mandatory update in the tape drive. Then type the letter Y (uppercase or lowercase) and press RETURN.

   As the installation proceeds, you will receive messages telling you that items are being processed. The VMSINSTAL procedure will shut down the system when the installation of the mandatory update has completed.

9. **Reboot the system.** First press the Halt button twice, then type the following command at the console-mode prompt:

   `>>> B ddcu`

   where **ddcu** is the physical device name of the system disk. See Appendix B of the *Installing or Upgrading MicroVMS From Diskettes* or *Installing or Upgrading MicroVMS From a Tape Cartridge* if you do not know the physical device name of the system disk.

Chapter 2

New and Changed Features

This chapter contains information pertaining to the new features that have been added for MicroVMS Version 4.4. A brief description of each new feature is provided, including a reference to where more information can be found on the topic.

Topics in this chapter can be found under the following categories:

- Section 2.1—General User Information

- Section 2.2—System Manager Information

- Section 2.3—Application Programmer Information

- Section 2.4—System Programmer Information

To find specific topics, consult the index in the back of this manual.

2.1 General User Information

This section contains information about new features in Version 4.4 of interest to the general user.

2.1.1 Command Line Recall Capability—Expanded

For interactive utilities and layered products that use the MicroVMS screen management software, you can recall up to the last 20 command lines by entering `CTRL/B` or by using the up-arrow and down-arrow keys. Examples of MicroVMS utilities that permit this feature are Mail, Debug, EDT Editor, and VAXTPU Editor.

2.1.2 DCL (Digital Command Language) Commands—New Commands

New commands are:

- CALL

- GOSUB

- RETURN

- SET SYMBOL/SCOPE

- SUBROUTINE

- ENDSUBROUTINE

- SET RIGHTS_LIST

Refer to the *MicroVMS User's Manual* for more information about these commands.

2.1.3 File Specifications and Logical Names—Hyphen Permitted

The hyphen (-) is permitted in the file name, file type, and directory fields of a DCL file specification. Hyphens may also be used in a logical name that appears as an unpunctuated file specification. Hyphens cannot be used in node names or device names.

Since the hyphen is also the DCL command-continuation character, entering a file name ending with a hyphen can be awkward. Therefore, creating files with a trailing hyphen in the name or type field is not recommended.

2.1.4 VAX Text Processing Utility (VAXTPU)—Changes

This section outlines changes that have been made to the VAX Text Processing Utility (VAXTPU).

2.1.4.1 Recompiling Section Files

In making VAXTPU faster and more functional, the format of the section files was changed. All section files must be rebuilt.

2.1.4.2 Default Section File Type Change

The default file type for VAXTPU section files was changed with this release. The default file type is now TPU$SECTION; in previous versions, it was GBL.

2.1.4.3 EVE$INIT_KEY and EVE$CLEAR_KEY in EVE

The internal, predefined EVE procedures EVE$INIT_KEY and EVE$CLEAR_KEY are no longer used with Version 4.4. If you used these procedures to extend the EVE interface, substitute the VAXTPU built-in procedures, DEFINE_KEY and UNDEFINE_KEY, respectively.

Be aware of key maps and key-map lists when defining keys. EVE does not use VAXTPU's normal default key map, TPU$KEY_MAP. The key map you probably want to use is EVE$USER_KEYS. EVE$USER_KEYS is the default key map in EVE which has precedence over EVE's other key maps. Other key maps in EVE include either EVE$VT100_KEYS or EVE$VT200_KEYS for the keypad keys, and EVE$STANDARD_KEYS for all other keys.

2.1.4.4 Callable Interface

The callable interface has the following two changes.

When calling the user I/O routine with the code TPU$K_GET, VAXTPU now passes a valid dynamic string descriptor as the DATA parameter. Your I/O routines should pass records to VAXTPU with the Run-Time Library string copy routines. Otherwise, you may get an access violation error (ACCVIO) when VAXTPU attempts to free the memory allocated to the string.

The global symbols TPU$COMMAND_TABLE, TPU$FACILTY_NAME, and TPU$MESSAGE_FLAGS in the callable interface are no longer used.

2.1.4.5 GET_INFO (SYSTEM, "timed_message")

In previous versions, when the timer was set to OFF with the built-in procedure SET (TIMER, OFF), the built-in procedure GET_INFO (SYSTEM, "timed_message") returned a null string. With this release, the string that was last specified with the built-in procedure SET (TIMER, ...) is returned. If no string was specified, the default string "working" is returned.

2.2 System Manager Information

This section contains information about the MicroVMS Version 4.4 Release intended for system managers.

2.2.1 System Security—New Command and Attributes

New features include a new DCL command, SET RIGHTS_LIST, and a new attribute, DYNAMIC.

SET RIGHTS_LIST adds and removes identifiers from the process and system rights list. You can assign the DYNAMIC attribute to identifiers to enable unprivileged users to add or remove identifiers they hold from their process rights list.

For more information on changes to the security system services, see the "DCL Appendix" section of the *MicroVMS Programming Support Manual*.

2.2.2 BATCH/PRINT Facility—New Features

The following new and changed features are documented in Section 9 of the *VAX/VMS System Manager's Reference Manual* for Version 4.4. Note that this manual was previously entitled *Guide to VAX/VMS System Management and Daily Operations*.

- The DCL command START/QUEUE/MANAGER has a new /[NO]RESTART qualifier. This qualifier causes the queue manager to automatically restart on recovery from a job controller abort. In addition, batch and output queues are restored to the status that existed prior to the interruption of service. The default is /NORESTART. For more information, see the *VAX/VMS DCL Dictionary*.

- You can now define a queue-specific default form for a printer, terminal, or server queue. To do this, first define a form using the DEFINE/FORM command. Then use the new FORM=type keyword with any of the following commands:

      INITIALIZE/QUEUE/DEFAULT=(FORM=type)
      START/QUEUE/DEFAULT=(FORM=type)
      SET QUEUE/DEFAULT=(FORM=type)

  A job submitted without an explicit form name in the PRINT command line will be processed using the default form specified for the queue.

- The /FORM qualifier has been renamed to /FORM_MOUNTED for the following commands:

      INITIALIZE/QUEUE/FORM_MOUNTED=type
      START/QUEUE/FORM_MOUNTED=type
      SET QUEUE/FORM_MOUNTED=type

The FORM_MOUNTED qualifier associates the paper stock of a form with that of the output queue. The stock types must match or the job will enter a pending state.

- The new /NOAFTER qualifier can be used with the SET QUEUE/ENTRY command to immediately release a job held by previously specifying the /AFTER qualifier.

- The new /[NO]PAGE_SETUP qualifier of the SET QUEUE/ENTRY command specifies one or more device control library modules, which perform special printer functions before the printing of each page.

2.2.3 Authorize Utility—Changes

This section contains information pertaining to the Authorize Utility.

2.2.3.1 /ATTRIBUTES—New Keyword

The Authorize Utility has a new keyword for the /ATTRIBUTES qualifier. You can specify the [NO]DYNAMIC keyword with the following commands:

ADD/IDENTIFIER/ATTRIBUTES
GRANT/IDENTIFIER/ATTRIBUTES
MODIFY/IDENTIFIER/ATTRIBUTES

Specifying the [NO]DYNAMIC keyword indicates whether unprivileged holders of the identifiers may add or remove them from the process rights list. The default is NODYNAMIC.

2.2.3.2 /ACCESS Qualifier—Enhanced

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"
```

2.2.3.3 /DEFPRIVILEGES and /PRIVILEGES Qualifiers

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

2.2.3.4 Secondary Passwords—Change

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 MicroVMS, anyone could essentially render an account useless by simply adding a secondary password that the account's owner did not know. If a user now tries to initiate a secondary password, the system will respond as follows:

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

2.2.3.5 AUTOLOGIN Flag—New Feature

A flag named AUTOLOGIN is 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 user name 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

2.2.4 VAX Text Processing Utility (VAXTPU) - Changes

The following changes to the VAX Text Processing Utility (VAXTPU) affect system managers.

2.2.4.1 VAXTPU Packaging

Previously, VAXTPU was packaged in one shareable image, TPUSHR.EXE. With this release, the screen management routines are placed in their own shareable image, TPU$CCTSHR.EXE. Install each image as follows:

```
INSTALL> SYS$SHARE:TPUSHR.EXE /OPEN/HEADER/SHARE

INSTALL> SYS$SHARE:TPU$CCTSHR.EXE /OPEN/HEADER/SHARE
```

2.2.4.2 Changing the Editing Interface from EVE to the EDT Keypad Emulator

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

2.2.4.3 Using Section Files

Section files are now installable as shared images. At the Install Utility prompt, enter
the following:

```
INSTALL> SYS$SHARE:TPUSECINI.TPU$SECTION /OPEN/HEADER/SHARE
```

The preferred method of invoking a section file other than the default section file is
to define the logical TPUSECINI to point to the appropriate section file. For example:

```
$ DEFINE TPUSECINI device:[dir]mysection.TPU$SECTION
```

where

device is the device name

dir is the directory name

mysection is the file name for the section file to be invoked.

2.2.5 MOUNT/CACHE=TAPE_DATA

The MOUNT command /CACHE=option qualifier has a new option, TAPE_DATA.
The /CACHE=TAPE_DATA qualifier enables the write cache for a tape device if
the tape controller supports a write cache. /NOCACHE is the default for mounting
on tape devices. If the tape controller does not support a write cache, the option is
ignored.

Note that the other options for the /CACHE=option qualifier pertain only to disks,
while the TAPE_DATA option is used only with magnetic tapes.

```
$ MOUNT/CACHE=TAPE_DATA MUA0:   TAPE
%MOUNT-I-MOUNTED, TAPE  mounted on _NODE$MUA0:
```

This command mounts the volume TAPE on device MUA0 and instructs MOUNT to
enable the tape controller's write cache for MUA0:.

2.2.6 AUTOGEN Command Procedure—New Features

This section describes Version 4.4 changes to the AUTOGEN.COM command procedure.

Calculation of LRPSIZE Parameter

In Version 4.4, AUTOGEN sets a value of 1504 for the LRPSIZE parameter if Ethernet is present on the system. This value is set on the assumption that Ethernet is the preferred DECnet communication mode. If you are using another mode (primarily CI, for example), edit the file SYS$SYSTEM:MODPARAMS.DAT and specify a value of 576 for LRPSIZE. When you rerun AUTOGEN, all related values will be set appropriately.

Calculation of VAXCLUSTER Parameter

AUTOGEN now calculates the correct value for the VAXCLUSTER parameter. It is no longer necessary to set the value explicitly in MODPARAMS.DAT.

Calculation of Primary Page and Swap File Sizes

If, to meet special needs on your system, you have manually allocated secondary page or swap file space, you should specify a value of 0 for the PAGEFILE and SWAPFILE parameters in MODPARAMS.DAT. AUTOGEN will then not attempt to recalculate file sizes.

2.3   Application Programmer Information

This section contains information about new and changed features available to application programmers.

2.3.1 QIO System Service—New and Changed Features

New and changed QIO features are available in Version 4.4. See section QIO of the *MicroVMS Programming Support Manual* for further information.

2.3.1.1 New Item Code

A new item code has been added:

    TRM$_ESCTRMOVR

2.3.1.2 New Bit Offsets

New bit offsets have been added:

    TT$V_BREAK
    TT2$V_DECCRT2
    TT2$V_SYSPWD

2.3.1.3 Status Code Change

The status code SS$_EXQUOTA no longer exists.

2.3.1.4 New Controller Values

New values for the controller have been added:

DT$_DMZ32
DT$_DHV
DT$_DHU

2.3.2 Linker Utility—Debugging Permitted for Shareable Images

You can now link a shareable image by issuing the following command line:

```
$ LINK/SHAREABLE/DEBUG image-name,...
```

The qualifiers /[NO]TRACEBACK and /DEBUG will be processed for a shareable image exactly as they are for an executable image. Previously, the /DEBUG qualifier was prohibited and the /[NO]TRACEBACK qualifier was ignored when linking a shareable image.

2.3.3 Debugger—New Features

New debugger features, summarized below, are described in the *MicroVMS Programming Support Manual* and, in more detail, in the *VAX/VMS Debugger Reference Manual*.

2.3.3.1 Screen Mode Enhancements

Screen mode enhancements are as follows:

- New PROMPT predefined display.

- New display attributes for SELECT command (/INPUT, /ERROR, /PROGRAM, /PROMPT).

- SET WINDOW, SET DISPLAY, and DISPLAY commands let you divide windows vertically.

- New built-in window definitions.

- New built-in symbols %PAGE and %WIDTH.

- New MOVE, EXPAND, and EXTRACT commands.

- Displays are now dynamic by default (display windows are resized as you change screen height or width).

- New qualifiers for DISPLAY and SET DISPLAY commands: /[NO]DYNAMIC, /[NO]POP, /[NO]PUSH.

- SHOW DISPLAY and SHOW WINDOW commands accept lists of parameters, wildcards, and /ALL qualifier.
- New keypad key definitions.

2.3.3.2 Other New Features and Changes

Other new features and commands are as follows:

- The debugger supports VAX DIBOL and VAX SCAN.
- You can debug shareable images (new [SET, SHOW, CANCEL] IMAGE commands).
- New ATSIGN commands (SET, SHOW) let you set/display the default file specification for command procedures.
- New EDITOR commands (SET, SHOW) let you define/display the editor invoked by the EDIT command.
- New SHOW STACK command provides detailed information about the call stack.
- You can qualify the SET BREAK, SET TRACE, and STEP commands with the /[NO]SHARE and /[NO]JSB qualifiers.
- You can control the interpretation of data in untyped locations by specifying EXAMINE/TYPE=exp, DEPOSIT/TYPE=exp, and SET TYPE TYPE=exp.
- New default scope for symbol lookup (equivalent to SET SCOPE 0,1,2,3, . . . ,n).
- You can invoke the debugger from a program by signaling SS$_DEBUG.

2.3.4 ANALYZE/RMS_FILE Utility—New Commands Added

The following commands are new:

- NEXT
- BACK
- POSITION/BUCKET
- POSITION/RECORD

These commands make it easier to examine file structures interactively. Also, new integrity features check more thoroughly for file structure errors. Refer to the *VAX/VMS Analyze/RMS File Utility Reference Manual* for further information.

2.3.5 Error Log Utility—New Features and Changes

The new features and changes that have been added to the Error Log Utility are outlined in the following two sections.

2.3.5.1 Enhancements to the User Interface

The ANALYZE/ERROR_LOG command has been enhanced to support the following:

1. New device class keywords for /EXCLUDE and /INCLUDE:

   WORKSTATION   Include or exclude workstation error log entries.
   LINE_PRINTER   Include or exclude line printer error log entries.

2. The BUSES keyword that is supported by /INCLUDE and /EXCLUDE has been enhanced to include BI bus error log entries.

3. The DEVICE_ERRORS keyword that is supported by /INCLUDE and /EXCLUDE has been enhanced to include BI adapter error log entries.

The error log entries for workstations and line printers can also be specifed by indicating the device name or the type of entry that is logged for the new hardware to /INCLUDE and /EXCLUDE.

2.3.5.2 /EXCLUDE Qualifier Added

By default, whenever an "unknown" device, CPU, or error log entry is encountered by ANALYZE/ERROR_LOG, it will output the entry in a hexadecimal longword format. The /EXCLUDE=UNKNOWN qualifier excludes these entries from the report.

2.3.6 Sort/Merge Utility—Changes

The following changes relate to the Sort/Merge Utility.

2.3.6.1 Sort/Merge Message Symbols Made Universal

The following Sort/Merge message symbols were made universal for Version 3.0 compatibility:

    SOR$_BADLOGIC
    SOR$_CLOSEDEL
    SOR$_CLOSEIN
    SOR$_CLOSEOUT
    SOR$_INSVIRMEM
    SOR$_OPENIN
    SOR$_OPENOUT
    SOR$_READERR
    SOR$_SYSERROR
    SOR$_WRITEERR

2.3.7   Print Symbiont—User Defined I/O Routines Supported

A synchronous return of status from a user-defined output routine is supported in Version 4.4. Previously, if a user-supplied output routine returned synchronous status, the modified symbiont was not supported for checkpointing. Futhermore, the output of the DCL command SHOW QUEUE displayed the state of the output queue associated with the modified symbiont as "stalled."

In addition, user-defined input filter routines correctly allow the modification of the carriage control of the input record.

User-defined main input routines that do not support the function code PSM$K_REWIND are automatically called with function code PSM$K_CLOSE followed by the function code PSM$K_OPEN. This call sequence provides the intended function of the PSM$K_REWIND code. This allows user-modified symbionts to support search and alignment operations provided by the standard MicroVMS print symbiont.

The Print Symbiont Utility routine PSM$READ_ITEM_DX is now supported for user-supplied output routines. Previously, PSM$READ_ITEM_DX was not properly supported when called from a user-defined output routine.

2.3.8   System Services—New Features Added

New system services routines and attributes have been added for version 4.4. A description of new and changed system service routines is available in Version 4.4 of the *MicroVMS Programming Support Manual*.

2.3.8.1 New System Services

The routine names and their descriptions appear in the table that follows.

New Routine	Description
SYS$CHECK_ACCESS	Checks third-party accessor's rights against the object's protection.
SYS$GETUAI	Returns authorization information about a specified user.
SYS$RMSRUNDWN	Closes all files opened by RMS on behalf of the process or image.
SYS$SETDDIR	Allows you to change the default directory string for the process.
SYS$SETDFPROT	Allows you to change the default file protection for the process.
SYS$SETUAI	Modifies the user authorization file (UAF) record for a specified user.

2.3.8.2 New Attribute

In MicroVMS Version 4.4, the DYNAMIC attribute is being added to the list of attributes (currently a list of one) found in many of the security-related system services. The following system services are affected:

- $ADD_HOLDER
- $ADD_IDENT
- $ASCTOID
- $FIND_HELD
- $FIND_HOLDER
- $GRANTID
- $IDTOASC

The format for DYNAMIC is as follows:

Bit	Meaning When Set
KGB$V_DYNAMIC	Allows the unprivileged holder to add or remove the identifier from the process rights list.

2.3.9 BASIC—Support Changes

This section explains changes relating to support for BASIC.

2.3.9.1 Error Messages—Modifications and Additions

- The severity of BASIC error 116 (PRIUSIFOR, PRINT-USING format error) has been changed from FATAL to SEVERE. This allows the error to be trapped by a BASIC error handler.

- Two new errors have been introduced: BASIC error 190 (ILLNETOPE, Illegal network operation) and BASIC error 191 (ILLTFFOPE, Illegal terminal-format file operation).

- BASIC error 190 is returned when a program attempts to mix the use of the GET or INPUT statements with PUT or PRINT statements on a terminal format file located on a remote node.

- BASIC error 191 is returned when a program uses the GETRFA built-in function on a terminal format file. Both errors can be avoided by opening the file with ORGANIZATION SEQUENTIAL VARIABLE.

- Previously, when executing on a terminal format file, the GETRFA function exhibited different behavior based on whether the file was remote or local. For a remote file, the function signaled BASIC error 131 (NO_CURREC, No current record). For a local file, the function successfully returned and the GET statement using the RFA returned signaled BASIC error 141 (ILLOPE, Illegal operation). In MicroVMS Version 4.4, the GETRFA function always returns BASIC error 191 (ILLTFFOPE, Illegal terminal-format file operation).

- In previous releases, mixed usage of GET or INPUT statements with PUT or PRINT on a remote terminal format file caused BASIC error 12 (FATSYSIO, Fatal system I/O failure). This condition now causes BASIC error 190 (ILLNETOPE, Illegal network operation).

2.3.9.2 PRINT USING Function—Behavior Change

The behavior of the PRINT USING function for backslash characters has been changed. In prior versions of MicroVMS, if a backslash was not followed by another backslash and was not followed by another format sequence, the backslash and other string constants following it were ignored. If it was followed by another format sequence, BAS$_PRIUSIFOR was signaled.

The behavior is now compatible with BASIC-PLUS. In both cases, the backslash is treated as a string constant (just as if it had an underscore in front of it).

2.3.9.3 FIND Statement—Function Change

A problem in the FIND statement with a RECORD clause that caused the lock flags (for example, ALLOW xxxx, REGARDLESS) to be ignored has been corrected. The flags are now interpreted correctly, which causes a change in behavior. A FIND used with a ALLOW MODIFY statement does not keep the current record pointer set. Hence, an update cannot be performed.

2.3.9.4 Sort/Merge—/INCLUDE and /OMIT Qualifiers

Previously, Sort/Merge incorrectly processed conditions specified by the /INCLUDE or /OMIT qualifiers that tested packed decimal fields near the end of a record. The affected record was never included in the output file. In Version 4.4, the record is correctly included or omitted based on the condition specified in the /INCLUDE or /OMIT qualifier.

2.3.9.5 Sort/Merge—%D Radix Designation Error

In prior versions of MicroVMS, use of the %D radix designation caused an error. It now correctly causes the constant to be treated as a decimal constant.

2.3.10 Run-Time Library—New Features Incorporated

The following sections contain information about new features that have been added to the Run-Time Library. These features include new procedures, new arguments, and a new facility.

A complete description of the new features for the Run-Time Library routines is located in the *MicroVMS Programming Support Manual*.

2.3.10.1 New Procedures

There are several new procedures available in Version 4.4. These procedures and their functions are as follows:

New Procedure	Function
New LIB$ Procedures	
LIB$PAUSE	Suspends program execution
New SMG$ Procedures	
SMG$COPY_VIRTUAL_DISPLAY	Creates a copy of a virtual display
SMG$DISABLE_BROADCAST_TRAPPING	Disables the trapping of broadcast messages
SMG$GET_KEYBOARD_ATTRIBUTES	Retrieves information about a virtual keyboard
SMG$GET_PASTING_INFO	Retrieves information about a virtual display
SMG$REPLACE_INPUT_LINE	Replaces lines in the recall buffer with a specified string
SMG$RETURN_INPUT_LINE	Returns a line from the recall buffer
SMG$SET_CURSOR_MODE	Turns the physical cursor on and off

2.3.10.2 New Arguments

New arguments have been added to the following existing routines:

- SMG$CREATE_VIRTUAL_KEYBOARD

- SMG$MOVE_VIRTUAL_DISPLAY

- SMG$PASTE_VIRTUAL_DISPLAY

- SMG$PUT_LINE

- SMG$READ_COMPOSED_LINE

- SMG$READ_KEYSTROKE

- SMG$READ_STRING

- SMG$READ_VERIFY

- SMG$REPASTE_VIRTUAL_DISPLAY

- SMG$SNAPSHOT

2.3.10.3 Obsolete Routines

The following procedures are now obsolete:

- The Run-Time Library routine LIB$SYS_TRNLOG is obsolete because it is a jacket routine for the now obsolete system service SYS$TRNLOG. It is suggested that users directly invoke the new SYS$TRNLNM system service for logical name translation. Due to the increased capabilities of higher-level languages in constructing item lists, the Run-Time Library has no current plans to provide a jacket routine to this new system service.

- SMG$PUT_WITH_SCROLL. The routine SMG$PUT_LINE now supports scrolling, therefore the SMG$PUT_WITH_SCROLL routine is obsolete.

- SMG$ALLOW_ESCAPE. This routine was created solely for the purpose of translating old application programs that send escape sequences to SMG$, and is no longer supported.

These routines remain in the MicroVMS Version 4.4 documentation, but will not be documented in future editions.

2.3.10.4 Other Changes

This section contains assorted changes to the Run-Time library.

- The capability for nonminimal updating is now supported. Nonminimal updating redraws only those lines affected by a change, beginning at the first changed character and proceeding to the end of the line.

- The output for LIB$SHOW_TIMER has changed. Previously, the "elapsed time" output was of the form HHHH:MM:SS.CC. In Version 4.4, this has changed to DDDD HH:MM:SS.CC. Therefore, the output of the example program in the documentation should now read:

```
ELAPSED:     0 00:00:00.22  CPU: 0:00:00.06  BUFIO: 1  DIRIO: 0  FAULTS: 18
```

- The description for the p-kit argument in the SMG$GET_KEYBOARD_ATTRIBUTES routine should be the same as the pb-info-table argument description in SMG$GET_PASTEBOARD_ATTRIBUTES.

- As of Version 4.4, RMS provides full record interlocking for shared sequential organization files. Prior to the FT Update (Y4.4), the Run-Time Library would set the FAB$B_SHR option bit FAB$V_UPI if the user wished to share a sequential file. In the FT Update, the RTL no longer sets that bit in order to take advantage

of full RMS record interlocking for shared sequential files. (This affects programs written in VAX FORTRAN, VAX PASCAL, and VAX PL/I.)

Setting the UFO bit in the file options field requires that UPI also be set. Thus, when you set the FAB$V_UFO FOP bit in your USEROPEN routine, you must also now set the UPI bit in the SHR field as well. The RTL will no longer set this bit for you.

- The function of LIB$STAT_TIMER has changed for Version 4.4. The elapsed time returned is now a delta time. Previously, the elapsed time was returned as a difference of two absolute times.

- In a future release of the RTL, users should notice that many parameter names will change. This is in conjunction with an effort to make all RTL parameter names more consistent throughout the various RTL facilities.

  For example, in a future release of the SMG$ RTL, the function-specific flag parameters will all be renamed to a generic parameter named FLAGS. Users employing these specific flag parameters should specify a value of 1 to set the flag and 0 to clear the flag. This will allow this parameter to be used in an upward-compatible manner. The following SMG$ routines will be affected by this parameter name change:

Routine	Parameter
SMG$CREATE_PASTEBOARD	preserve-screen-flag
SMG$DELETE_PASTEBOARD	clear-screen-flag
SMG$PUT_CHARS	erase-flag
SMG$PUT_LINE	wrap-flag
SMG$PUT_LINE_WIDE	wrap-flag
SMG$PUT_LINE_HIGHWIDE	wrap-flag
SMG$PUT_PASTEBOARD	p-ff-flag
SMG$SNAPSHOT	ff-flag
SMG$PUT_WITH_SCROLL	wrap-flag
SMG$READ_COMPOSED_LINE	function-keys-flag

2.3.10.5 Screen Management Restriction

Due to changes made to the Screen Management Facility, the following restriction now applies to the routines SMG$SET_BROADCAST_TRAPPING, SMG$ENABLE_UNSOLICITED_INPUT, and SMG$SET_OUT_OF_BAND_ASTS. For AST routines written in a language that does not support optional parameters (for example VAX BASIC), all system parameters must be specified. This restriction is illustrated in the example for the SMG$DISABLE_BROADCAST_TRAPPING routine.

2.3.11 RMS—New Features

Under MicroVMS Version 4.4, RMS now supports full file sharing, record locking, and the use of global buffers for **all** sequential files. Also, you may now define descending keys for indexed files. See the *VAX Record Management Services Reference Manual* for details.

2.3.12 Terminal Driver Support—Changes

The following changes have been made to MicroVMS terminal support.

2.3.12.1 Sending a Break

Previously, there was no $QIO System Service that allowed an application to send a break to a terminal. A break can now be sent by setting TT$M_BREAK in the P5 parity flags argument to the set mode $QIO.

Sending the break actually involves two $QIOs; one to turn it on and one to turn it off. (The break bit in the parity flags argument is set to turn on break and cleared to turn it off.) The application should use a timer in between the two $QIOs to ensure that the break has time to take effect.

2.3.12.2 Preventing Partial Escape Errors

Prior to Version 4.4, the only way to correct partial escape errors (SS$_PARTESCAPE) was for the application program to do single-character reads to parse the remaining characters to determine when the escape sequence was terminated. The terminal driver now supports an alternative approach that allows the application to specify an overflow buffer to be used only for an escape sequence terminator.

A new item code for the item-list read, TRM$_ESCTRMOVR, specifies the number of bytes in the read buffer to be reserved for the escape terminator. The P2 parameter, which specifies the size of the read buffer, should include both the number of bytes to receive data and the number of bytes reserved for the escape terminator overflow. Normally the overflow area can be small, perhaps about 10 bytes, since that is sufficient to hold any escape sequence generated by a DIGITAL terminal.

When the terminator overflow area is specified, any bytes from an escape sequence terminator that will not fit in the data area of the buffer will be allowed to occupy the overflow area in the buffer. For instance, a user would be able to type a single character terminated by a keypad key and not get the SS$_PARTESCAPE error, even when the data area is limited to 1 byte.

2.3.12.3 SET HOST/DTE Can Generate a Break

In order to log in on lines that expect a break rather than carriage return, you can now generate a break in SET HOST/DTE by pressing the CTRL and right bracket (]) keys.

2.3.12.4 SET HOST/DTE/DIAL—Problem and Solution

SET HOST/DTE/DIAL does not work with the DMF-32 controller. The problem is that the modem sends a response character to the host when it detects a carrier signal, but the DMF-32 drops any input until it sees the carrier signal.

One solution is to modify the example autodialer provided in SYS$EXAMPLES:DTE_DF03.MAR to perform a IO$_SENSEMODE!IO$M_RD_MODEM $QIO to check for a carrier signal. If set, the autodialer should assume success and continue.

2.3.12.5 Other Changes

In Version 4.0, lines with the MODEM characteristic would hang up 30 seconds after sensing a CARRIER signal if a channel was not assigned to the device. This feature was implemented as a security feature to prevent unused lines from being tied up. It is now possible to disable this hangup on a systemwide basis by setting bit 2 (value = 4) in the SYSGEN parameter TTY_DIALTYP.

2.3.13 Logical Names Associated With Mailboxes and Mounted Volumes—Changes

In Version 4.4, logical names associated with mailboxes and mounted volumes are no longer automatically deleted unless they are shared logical names. Because all default cases result in the creation of shared logical names, this change is likely to affect only a small number of applications that have deliberately redirected the logical names associated with mailbox creation or the mounting of a volume.

When a mailbox is created, an optional logical name can be associated with the mailbox. Names associated with temporary mailboxes are placed in the logical name table located by the following name:

```
LNM$TEMPORARY_MAILBOX
```

Names associated with permanent mailboxes are placed into the table located by the name:

```
LNM$PERMANENT_MAILBOX
```

The default assignments for these names are LNM$JOB and LNM$SYSTEM respectively.

When a volume is mounted, a logical name is placed into a table whose name depends on the type of mount operation performed.

Qualifier	Name Table
none	LNM$JOB
/GROUP	LNM$GROUP_xxxxxx
/SYSTEM	LNM$SYSTEM

Because all of these logical name tables translate to shared tables, the default behavior applies. When a mailbox disappears (the last process deassigns its channel to the mailbox) or a volume is dismounted, the shared logical name is deleted. The change in Version 4.4 only applies when one of these table names (such as LNM$TEMPORARY_MAILBOX or LNM$JOB) is redirected to a process-private table (such as LNM$PROCESS_TABLE). In the case of redirection, the logical name associated with a mailbox or mounted volume remains when the mailbox disappears or the volume is dismounted.

2.4   System Programmer Information

This section contains information about the new and changed features of Version 4.4 that are of interest to system programmers.

2.4.1   System Dump Analyzer—New and Changed Features

The following new commands have been added to the System Dump Analyzer:

- ATTACH

- SPAWN

Also, the following new qualifiers are available for the EVALUATE, EXAMINE, and SEARCH commands:

- EVALUATE /PSL

- EVALUATE /PTE

- EVALUATE /SYMBOLS

- EXAMINE /NOSUPPRESS

- EXAMINE /PTE

- SEARCH /LENGTH=length_specifier

- SEARCH /STEPS=step_factor

The following commands are new or changed for Version 4.4:

- The ATTACH command allows you to switch control of your terminal to another process in your job. The /PARENT qualifier allows you to switch control of your terminal to the parent process of the current process.

- The SPAWN command creates a subprocess from the current process. The context is copied from the current process to the spawned process.

- The EVALUATE/PSL command evaluates the specified expression in the format of a processor status longword.

- The EVALUATE/PTE command interprets and displays the expression as a page table entry (PTE). The individual fields of the PTE are separated and an overall description of the PTE's type is provided.

- The EVALUATE/SYMBOLS command specifies that all symbols that are known to be equal to the evaluated expression are to be displayed.

- The EXAMINE/NOSUPPRESS command inhibits the suppression of zeros when displaying memory with one of the following qualifiers: /ALL, /P0, /P1, /SYSTEM.

- The SEARCH/LENGTH command specifies the size of the expression value to be used for successful matching during searches of memory. The possible values of this qualifier are: BYTE, WORD, and LONGWORD.

- The SEARCH/STEPS command controls the granularity of searching through the specified memory range. As each comparison of memory occurs, the value of this qualifier determines the next memory location to be searched. The possible step_factors are: BYTE, WORD, LONGWORD, and QUADWORD.

- The COPY command releases the dump pages in the paging file so that they are available for system paging. Note that once the COPY command has released the dump pages for paging use, the dump information in these pages may be lost. Subsequent dump analysis should be carried out on the copy of the dump file that was specified in the COPY command.

- Logical operations have been added to the SDA. They are the logical AND, logical OR, logical XOR. The operators for these operations are the characters &, I, and \.

- The SET PROCESS and SHOW PROCESS commands can now include quoted strings in the process name in addition to the previous capital letters, numbers, dollar sign ($), and underscore (_) characters.

- The SHOW DEVICE command examples have been changed and now include shadowed devices.

• The SHOW CRASH command register list now includes the system identification register.

• The SHOW PROCESS /RMS=IFAB display has been altered to show the changes to that display.

For more information, refer to the *VAX/VMS System Dump Analyzer Reference Manual*.

2.4.2 DR11–W/DRV11-WA (XADRIVER)—New Support

The DR11–W is a MicroVMS supported 16-bit parallel direct memory access (DMA) interface for UNIBUS systems. MicroVMS includes the driver source code for the DR11–W driver so that users can tailor the driver for their own applications.

Beginning with Version 4.4, MicroVMS also supports the DRV11-WA, a 16-bit parallel DMA interface on the Q-bus. Because the DRV11-WA and DR11–W interfaces are similar and many users of the DR11–W wish to convert their applications to run with the DRV11-WA, support for the DRV11-WA interface has been folded into the DR11–W driver.

The Version 4.4 XADRIVER does not contain bug fixes or enhancements that affect the DR11–W interface. If a customer has not tailored the driver (SYS$SYSTEM:XADRIVER.EXE), the new version of the driver will function on the DR11–W as the old. However, if any changes have been made to the driver, it is suggested those changes be merged into the Version 4.4 driver.

The Version 4.4 XADRIVER documentation and source code presume the DRV11-WA is at CS Revision Level B and Etch Revision Level D or **earlier**. If subsequent revisions are made to the board, the expected behavior of the driver is unpredictable.

The DRV11-WA driver is not supported on the MicroVAX I.

For additional information, see *VAX/VMS I/O User's Reference Manual: Part II* and the *DRV11-WA General Purpose DMA Interface User's Guide*.

Chapter 3

Problems, Restrictions, and Notes

This chapter contains information pertaining to problems and restrictions of the MicroVMS Version 4.4 release. It also includes general notes and documentation notes about the Version 4.4 release. Each topic is given a brief description and a reference to where more information can be found (when applicable).

This chapter is arranged into the following categories:

- Section 3.1—General User Information

- Section 3.2—System Manager Information

- Section 3.3—Application Programmer Information

- Section 3.4—System Programmer Information

To find a specific topic, consult the index in the back of this manual.

3.1 General User Information

This section contains information about the MicroVMS Version 4.4 release that pertains to all system users.

3.1.1 Version 4.0 Release Notes Appendixes—Disposition of Material

Appendixes B through G of the *VAX/VMS Release Notes, Version 4.0,* contained information from the Version 3 documentation set that had not been integrated into the Version 4 documentation set. The following table lists the topics and their disposition as reflected in the Version 4.4 documentation.

Topic	Disposition
Introduction to VAX/VMS Input/Output	Integrated into I/O discussion in the *VAX/VMS System Services Reference Manual*
Real-Time I/O (Mapping I/O Space, Connecting to Interrupt Vector)	Included in the *Writing a Device Driver for VAX/VMS* manual, Appendix H
User-Written System Services (Privileged Shareable Images)	Included in the *VAX/VMS System Services Reference Manual*, Appendix B
Using Shared Memory	Included in the *VAX/VMS System Services Reference Manual*, Appendix C
Obsolete Run-Time Library Routines	No longer documented
Summary of Run-Time Library Entry Points	Use the RTL section of the *VAX/VMS Mini-Reference*

3.1.2 Mini-Reference—Supersedes Quick-Reference Booklets

The *VAX/VMS Mini-Reference*, a new manual in the VAX/VMS document set for Version 4.4, contains comprehensive quick-reference information. It supersedes the following previously published quick-reference booklets:

- *VAX/VMS DCL Commands and Lexical Functions*

- *VAX EDT Quick Reference Guide*

- *VAX DSR Quick Reference Guide*

- *VAX/VMS System Services and Run-Time Library Routines*

3.1.3 *MicroVMS User's Primer*—Error Correction

There is an error in the example in Section 7.2.3 on page 7-4 of the Version 4.2 *MicroVMS User's Primer*. In the primer, the hyphen after the word "JONES" is missing.

This book is not being reissued for 4.4. The example should appear as follows:

```
$ SET DEFAULT SYS$SYSTEM
$ RUN AUTHORIZE
UAF> ADD JONES-
_/PASSWORD-ROCKET-
_/UIC=[21,1]-
_/DEVICE=$DISK1-
_/DIRECTORY=[JONES]
```

3.1.4 Terminal Driver Line Editing—Clarification

The following information clarifies the documentation for CTRL/V in Table 1–2 of the *VAX/VMS DCL Concepts Manual* and in Table GEN–2 of the *VAX/VMS Mini-Reference*.

At DCL level, CTRL/V turns off line-editing features that were new with Version 4.0. For example, if you type CTRL/V followed by a control key such as CTRL/D, a CTRL/D is generated instead of the cursor moving left one character. Note, however, that CTRL/D is a terminator at the DCL level. Thus, when you type CTRL/V followed by CTRL/D, a carriage return is simulated. DCL uses the default RMS terminator set. The characters that are part of this set are described in Chapter 8 of the *VAX/VMS I/O User's Reference Manual: Part I*.

When combined with CTRL/V, characters that are not terminators to DCL will have no effect since a backspace or line feed in the middle of a line would result in an invalid command. Examples are CTRL/H and CTRL/J.

Certain control keys perform the same function with Version 4.0 as they did in previous versions of MicroVMS. If you type one of these keys (including CTRL/U) after a CTRL/V, the key will behave as it did prior to Version 4.0.

3.1.5 *VAX/VMS Backup Utility Reference Manual*—Guide for New User's Added

The Description section of the *VAX/VMS Backup Utility Reference Manual* now includes a subsection called "Using BACKUP: A Guide for New Users". This section is an introduction to the Backup Utility for inexperienced users. It includes descriptions of the BACKUP function, operation types and modes, qualifiers, and save sets. Users who are new to BACKUP, or who use it infrequently, may benefit by reading the guide first and then reading the reference sections of the manual.

3.1.6 *VAX/VMS Mail Utility Reference Manual*—Text Addition

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 RUSCIO, 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 RUSCIO
```

3.1.7 *Guide to Using DCL and Command Procedures on VAX/VMS*—Documentation Changes

The following corrections apply to the *Guide to Using DCL and Command Procedures on VAX/VMS*.

- Page 1–11. The last two lines in the example LOGICALS.COM file should read as follows:

```
$ DEFINE JON DAISY::HARRIS
$ DEFINE JANE DAISY::MOORE
```

- Page 4–16. In the example in Section 4.6.2, the statement

```
$ WRITE "Result is",RES
```

  should be replaced as follows:

```
$ WRITE SYS$OUTPUT "Result is ",RES
```

- Page 5–14. A line of code is missing from the example at the top of the page. The line

```
$      NUM = NUM + 1
```

  should be inserted under PROCESS_LOOP as follows:

```
$ PROCESS_LOOP:
$      FILE = F$ELEMENT(NUM,"/",FILE_LIST)
$      IF FILE .EQS. "/" THEN GOTO DONE
$      COPY 'FILE'.MEM MORRIS::DISK3:[DOCSET]*.*
$      NUM = NUM + 1
$      GOTO PROCESS_LOOP
```

- Page 5–15. The first statement in the example should have a hyphen (-) at the end of the line as follows:

```
$ COMMAND_LIST = "DELETE/DIRECTORY/EXIT/" + -
```

- Page 6–7. In the example at the top of the page, the statement

```
$ INQUIRE RECORD "Enter name"
```

  should be replaced as follows:

```
$ INQUIRE NAME "Enter name"
```

- Page 8–5. In the text following the third bullet, the qualifier /NOPRINT should be /NOPRINTER.

3.1.8 Extended File Names/Types—Caution

Although file names and file types of up to 39 characters are permitted starting with VMS Version 4.0, for some files you may need to use the VMS Version 1.0 maximum lengths (9 characters for the file name and 3 characters for the file type), or other maximum lengths as appropriate.

For example, you must use restraint in naming files that will be accessed by:

- Operating systems that cannot support longer file names and file types, such as VMS Version 1.0 systems and systems for PDP–11 processors

- Applications software that will not accept longer file names and file types

Care should be taken when naming files that will be copied or accessed by remote systems. The file-naming abilities of VMS after Version 4.0 exceed those of most other computer systems, including VAX systems running VMS Version 1.0. For example, a system running VMS Version 1.0 will return a syntax error when a file specification contains a file name (including a directory name) longer than 9 characters, a file type longer than 3 characters, a dollar sign ($) or an underscore (_). Valid file specifications of VMS after Version 4.0 that are invalid on a VMS Version 1.0 system include the following:

```
NODE::DBA2:[YOUR_DIR]FILE.DAT
NODE::DBA2:[DIR]FILETOOLONG.DAT
NODE::DBA2:[DIR]FILE_TEST.DAT
NODE::DBA2:[DIR]FILE.DATA
```

A user of a Version 4.0, or later, system would have to rename (or copy) these files before the remote system could access them. Alternatively, the user could copy these files to the remote system by using valid VMS Version 1.0 output file specifications.

File name restrictions are generally determined by the file name capabilities of the remote system(s) that require access to the file. Such restrictions should be considered as part of the overall application design when network access is required.

Applications that parse file specifications using the pre-Version 4.0 file specification conventions should be modified to use the services or routines that can parse or scan file specifications using the new extended file specifications conventions. These services and routines include the RMS Parse service and the Scan String for File Specification system service (see the *VAX Record Management Services Reference Manual* and the *VAX/VMS System Services Reference Manual*) and the LIB$FIND_FILE and LIB$FILE_SCAN routines (see the *VAX/VMS Run-Time Library Routines Reference Manual*).

3.2 System Manager Information

This section contains information about the MicroVMS Version 4.4 release pertaining to system managers.

3.2.1 STANDALONE BACKUP—Mandatory Rebuild

You must rebuild standalone BACKUP after you have upgraded or installed Version 4.4 of VMS. It is recommended as normal procedure. DIGITAL wishes to draw special attention to this requirement. Please refer to the *MicroVMS User's Manual* for a complete description of how to rebuild standalone BACKUP.

3.2.2 Image Activation, Search Lists, and Known Images—Note

One of the steps involved in image activation uses 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 that 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.

3.2.3 *VAX/VMS Verify Utility Reference Manual*—Text Correction

On page VER–7, the example should read /READ_CHECK, not /[NO]READ_
CHECK. This correction will be incorporated in the next revision of the manual.

3.2.4 *VAX/VMS Developer's Guide to VMSINSTAL*—Text Correction

The VMSINSTAL CHECK_NET_UTILIZATION callback documented in Section
5.2 of the *VAX/VMS Developer's Guide to VMSINSTAL* (a new optional manual) is
described as follows:

"This callback determines whether the net number of free blocks on the VMI$ROOT
device is sufficient to successfully complete the installation."

The description should state "peak number" rather than "net number" of free blocks.
This correction will be incorporated into the manual in a future revision.

3.2.5 *VAX/VMS Install Utility Reference Manual*—Additions and Corrections

This section describes information not included in the Install Utility documentation.

3.2.5.1 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

3.2.5.2 /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.

3.2.5.3 Corrections to Text

Make the following corrections to the *VAX/VMS Install Utility Reference Manual*.
These corrections will be incorporated into the next revision of the manual.

*   On page INS-1, the format for invoking INSTALL is given as:

    ```
    RUN SYS$SYSTEM:INSTALL
    ```

    This command line format became obsolete with Version 4.0 when the foreign
    command format was implemented. To establish the INSTALL command as
    the default for your site, you must define the global symbol INSTALL in your
    SYLOGIN.COM file as follows:

    ```
    $ INSTALL = "$INSTALL/COMMAND_MODE"
    ```

Once this symbol is defined, you can invoke the Install Utility by entering INSTALL as a DCL command.

In a future release, this format will become the default.

- Page INS–2: Footnote 2 under Example INS-2 should read "with the /OPEN qualifier," not "with the /SHARED qualifier."

- Page INS–6: the privilege listed as SYSLCKL should read SYSLCK.

- Page INS–7: the file name GRPCOMMEXE should read GRPCOMM.EXE.

- Page INS–15: in the third paragraph, 00038E should read 0003E8.

3.2.6 *VAX/VMS Accounting Utility Reference Manual*—Corrections

Make the following corrections to the *VAX/VMS Accounting Utility Reference Manual*. These corrections will be incorporated in the next revision of the manual.

- Page ACC–4: in the example, the keyword ELAPSES should read ELAPSED.

- Page ACC–49: Figure ACC–7 is incorrect. There should be an empty, unused byte at offset 25. ACR$W_USERNAME should be at offset 26. Each item in the figure should be moved forward by 1 byte, starting with the USERNAME field.

3.2.7 *VAX/VMS Mount Utility Reference Manual*—Addition to Manual

The documentation for the jobwide MOUNT support was omitted from the VMS documentation. It should read as follows:

In VMS, 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.

3.2.8 *VAX/VMS DECnet Test Sender/DECnet Test Receiver Utility Reference Manual*—Text Changes

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.

3.2.9 Accounting Utility—Abbreviated Qualifier Values Restriction

The use of abbreviating qualifier values in the Accounting Utility can produce erroneous results from nonabbreviated qualifiers. For example, the following command produces a display of all summary information and LOGFAILs:

```
$ ACCOUNTING/SUMMARY=US !(US abbreviation for USER)
```

However, the identical command used with the nonabbreviated qualifier (USER) produces a display of all summary information without the LOGFAILs:

```
$ ACCOUNTING/SUMMARY=USER
```

This problem will be corrected in a future release. Until then, do not use abbreviated qualifiers in the Accounting Utility.

3.2.10 System Generation Utility (SYSGEN)—Notes and Restrictions

This section contains information related to the System Generation Utility (SYSGEN).

3.2.10.1 UDABURSTRATE Parameter

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

3.2.11 *VAX/VMS System Generation Utility Reference Manual*—Text Changes

The following notes document errors and omissions in the Version 4.2 manual:

- The SHARE command is incorrectly documented as SHARE/CONNECT.

- On pages SGN-19 and SGN-20, the examples shown for the CONNECT command are incorrect and should be as follows:

```
SYSGEN> CONNECT LPAO /ADAPTER=3/CSR=%0777514 -
SYSGEN> /DRIVERNAME=LP2DRIVER/VECTOR=%0200
        .
        .
        .
SYSGEN> CONNECT NET /NOADAPTER/DRIVER=NETDRIVER
```

- On page SGN-58, the final sentence in the description of the ACP_SHARE parameter should be as follows: "This parameter should be set on when ACP_MULTIPLE is on."

- On page SGN-62, the parameters FREEGOAL and FREELIM are listed as dynamic. These parameters are not dynamic.

- On page SGN-66, the description of the LNMHASHTBL parameter should indicate that the values specified for this parameter are always rounded up to the nearest power of 2. The same is true for the LNMPHASHTBL parameter.

- On page SGN-73, the parameters listed as PQL_DJJQUOTA and PQL_MJJQUOTA are misspelled and should be PQL_DJTQUOTA and PQL_MJTQUOTA respectively.

- On pages SGN-77 and SGN-78, the descriptions of the RMS_DFMBC and RMS_DFNBC parameters should be as follows:

RMS_DFMBC (D)

    RMS_DFMBC specifies the default disk block size used by RMS in accessing sequential files.

    Normally the default value is adequate.

RMS_DFNBC (D)

    RMS_DFNBC specifies a default block count for network access to remote, sequential, indexed sequential, and relative files.

    The network block count value represents the number of blocks that RMS is prepared to allocate for the I/O buffers used to transmit and receive data. The buffer size used for remote file access, however, is the result of a negotiation between VAX RMS and the remote File Access Listener (FAL). The buffer size chosen is the smaller of the two sizes presented.

    Thus, RMS_DFNBC places an upper limit on the network buffer size that will be used. It also places an upper limit on the largest record that may be transferred to or from a remote file. In other words, the largest record that can be transferred must be less than or equal to RMS_DFNBC multiplied by 512 bytes.

    Normally the default value is adequate.

- On page SGN-79, the following information should be included in the description of the SCSNODE parameter:

  Specify the parameter value as an ASCII string enclosed in quotation marks ("). Note that the string may not include dollar sign ($) or underscore (_) characters.

- On page SGN-84, the description of the TTY_DIALTYPE parameter should be as follows:

  TTY_DIALTYPE
    TTY_DIALTYPE provides flag bits for dial-ups. Bit 0 is 1 for United Kingdom dial-ups and 0 for all others. Bit 1 controls the modem protocol used. Bit 2 controls whether modem lines will hang up 30 seconds after seeing CARRIER if a channel is not assigned to the device. The remaining bits are reserved for future use. See the *VAX/VMS I/O User's Reference Manual: Part I* for more information on flag bits.

3.2.12 Batch/Print Facility—Notes

The following notes pertain to the Batch/Print facility.

3.2.12.1 SET QUEUE/ENTRY Command—Behavior Change

Specifying the /NOHOLD qualifier with the SET QUEUE/ENTRY command no longer releases jobs specified with /AFTER. You can now use the /NOAFTER qualifier to release a job submitted with /AFTER.

3.2.12.2 Tab Expansion Determined at Start of Queue

When the output queue is started, the Version 4.4 print symbiont determines if tab expansion is required by accessing the current device characteristics. The Version 4.4 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, DIGITAL recommends 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, DIGITAL recommends 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.

3.2.12.3 Generation of Blank Pages When Set-up or Reset Sequence is Specified

In Version 4.0, it was possible to create library setup/reset modules that were output to the device during the processing of the current print job. Setup/reset modules could be output before a specific file, before all files, or after the current job is completed. The Version 4.0 print symbiont incorrectly inserted form feeds after all setup or reset modules regardless of content. In Version 4.4, 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.

DIGITAL realizes 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. DIGITAL believes this area of the symbiont may require additional flexibility beyond that provided in this functional update. DIGITAL is currently investigating mechanisms by which additional flexibility may be provided.

3.2.12.4 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.

3.2.13 VAX RPG II Version 2.0—Restriction

If you currently have VAX RPG II Version 2.0 installed, you must reinstall it after installing or upgrading your system to Version 4.4.

3.2.14 VAX LISP Version 1.2—Incompatibility with MicroVMS Version 4.4

When a VAX LISP user resumes a suspended system, it must be mapped into memory at an exact address. This address is n pages beyond where the MicroVMS RTL ends. In MicroVMS Version 4.4, the run-time library (RTL) grows beyond the point where a VAX LISP Version 1.2 suspended system needs to start. VAX LISP exits with the fatal error, "VAX LISP image and suspended do not match."

This problem will be corrected in VAX LISP Version 2.0.

Customers who need to use VAX LISP Version 1.2 in the MicroVMS Version 4.4 environment should contact Margaret Meehan at (617) 568-6515 for more information.

3.2.15 VAX BASEVIEW—BYTLM Quota

Before installing VAX BASEVIEW Version 1.0 on VMS Version 4.4, it is necessary to raise the BYTLM quota in the User Authorization File (SYSUAF.DAT).

The steps to do this are:

```
$ SET DEFAULT SYS$SYSTEM
$ RUN AUTHORIZE
UAF> SHOW SYSTEM
```

If the SYSTEM account has a BYTLM value of less than 18100, then please raise the value to at least this number. DIGITAL recommends that this be raised slightly higher.

```
UAF> MODIFY SYSTEM/BYTLM=18200
UAF> EXIT
```

You must log out and log back into the SYSTEM account prior to performing the installation, or the raised value will not be in effect, and the installation will fail.

3.2.15.1 VMSINSTAL Option N—Compatibility Problem

Use of the new VMSINSTAL Option N to display or print layered product release notes is not compatible with earlier saved auto answer files created via the VMSINSTAL Option A.

As Option N did not exist on previous versions of MicroVMS, there was no way that it could be stored in the auto answer file.

As a result, use of an existing auto answer file with Option N will produce the following:

```
%VMSINSTAL-F-AUTOSYNC, Auto-answer file is not in synch with questions.
-VMSINSTAL-F-AUTOSYNC, question: * Select option [3]:
-VMSINSTAL-F-AUTOSYNC, file: * Do you want to purge files replaced by this
installation [YES]?
%VMSINSTAL-F-UNEXPECTED, Installation terminated due to unexpected
event.
VMSINSTAL procedure done at 14:00
```

The solution is either not to use an auto answer file with Option N or to recreate the answer file and use Option N concurrently by specifying both options appended together.

The N option allows the installer to view or print, or both view and print the online release notes for those optional layered software products that support online release notes. Note: Currently, not all layered products support online release notes. Use of Option N in these cases will produce no difference in the flow of the installation procedure.

3.2.16 *Guide to VAX/VMS Performance Management*—Additions to Manual

For Version 4.4, the *Guide to VAX/VMS Performance Management* includes the following new materials:

- A description of operations that the system manager may perform after installation to improve overall system performance.

- A new chapter, Managing System Resources. This chapter describes procedures for evaluating the performance of the CPU, memory, and disk I/O resources using the Monitor Utility and (to a lesser extent) other standard VAX/VMS utilities. Discussions focus on the utilization of each hardware resource by major VAX/VMS software components and on the measurement, analysis, and possible reallocation of the hardware resources. Suggestions for corrective actions are provided, in case evaluation indicates that improvements are possible.

  The chapter includes a tabular summary of important MONITOR statistical items along with rules of thumb for using MONITOR data to evaluate performance of the CPU, memory, and disk I/O resources.

3.2.17 Monitor Utility—Fails in Certain Cases

The Monitor Utility fails when monitoring more than one multiprocessor using the following classes: (CLUSTER, SYSTEM, or MODES).

The Monitor Utility also fails when PROCESSES/TOPxxx and the SYSTEM class are monitored at the same time.

Both of these problems will be fixed in the next release of MicroVMS.

3.2.18 *Guide to VAX/VMS System Security*—Text Changes

Please make the following corrections to the *Guide to VAX/VMS System Security*. These corrections will be included in the next revision of the manual.

3.2.18.1 Defining Ownership Privileges

Section 4.4.2 defines the conditions needed to convey ownership privileges to a user. The numbered list should be replaced with the following:

1. Hold the resource attribute to the identifier that owns the file

2. Running with BYPASS or SYSPRV

3. Running with GRPPRV and i n the same group as the file owner

3.2.18.2 Establishing and Changing File Ownership

Section 4.4.5 describes the steps MicroVMS uses to determine the default owner of a file. These steps should be replaced with the following list:

1. An attempt is made to propagate the ownership from a previous version of the file. This will only succeed if the user is privileged (holds BYPASS, SYSPRV, or GRPPRV privilege) or has ownership rights to the owner of the previous version.

2. If the attempt to propagate from the previous version fails (either because there is no previous version, the creator lacks ownership rights to the previous version, or the creator is not privileged), then an attempt is made to propagate ownership from the parent directory. This will only succeed if the user is privileged or has ownership rights to the owner of the parent of the directory.

3. If the attempt to propagate from the parent directory fails, then the owner of the created file will be the same as the creator of the file.

3.2.18.3 Default ACL Protection

The second sentence in Section 4.5.2.2 states the following:

In addition, when you create a file whose owner identifier is not your UIC, an ACE is added to your ACL for the file that grants full access to your UIC.

This sentence should be replaced with the following corrected version:

*In addition, when you create a file whose owner identifier is not your UIC, an ACE is added to your ACL for the file that grants CONTROL access plus the access available to the owner of the file (the **Owner** field of the SOGW protection mask.)*

A similar change will also be made to Section 5.2.6.2 and the flowcharts in Figures 4-4 and 5-5. These changes will be incorporated in the next revision of the manual.

3.2.18.4 Example Change

In Figure 5-10, the line *$ read /end . . .* should be placed following the line *$ delete /symbol /local /all.* This correction will be included in the next revision of the manual.

3.2.19 ACMS—Restriction

VAX ACMS Version 1.2 will cause your MicroVMS system to be unusable if you try to run on MicroVMS Version 4.4. Because of a problem latent in Version 1.2, ACMS fails to release locks on SYSUAF.DAT after checking user authorization. Until now, RMS has been silently handling the problem at image rundown. With MicroVMS Version 4.4, however, RMS will leave the ACMS locks in place, so that once a user has logged into ACMS, no subsequent users will be able to log into MicroVMS.

This problem has been fixed in ACMS Version 2.0, and ACMS Version 2.0 kits will be available before MicroVMS Version 4.4.

To avoid difficulty, follow these guidelines:

- Install ACMS Version 2.0 **before** you install MicroVMS Version 4.4.

- If you do not plan to upgrade to ACMS Version 2.0, **do not** install MicroVMS Version 4.4.

3.2.20 TMPJNL and PRMJNL—Removed

The TMPJNL and PRMJNL privileges, which were never used by MicroVMS, have been removed from MicroVMS Version 4.4.

Any documentation that may still mention these privileges will be updated to reflect this change in the next release.

3.2.21 AUTOGEN Command Procedure—New Features

This section describes Version 4.4 changes to the AUTOGEN.COM command procedure.

Calculation of LRPSIZE Parameter

In Version 4.4, AUTOGEN sets a value of 1504 for the LRPSIZE parameter if an Ethernet is present on the system. This value is set on the assumption that Ethernet is the preferred DECnet communication mode. If you are using another mode (primarily CI, for example), edit the file SYS$SYSTEM:MODPARAMS.DAT and specify a value of 576 for LRPSIZE. When you rerun AUTOGEN, all related values will be set appropriately.

Calculation of Primary Page and Swap File Sizes

If, to meet special needs on your system, you have manually allocated secondary page or swap file space, you should specify a value of 0 for the PAGEFILE and SWAPFILE parameters in MODPARAMS.DAT. AUTOGEN will then not attempt to recalculate file sizes.

Reset PHYSICALPAGES on VAXstation I After Adding Memory

If you add memory to a VAXstation I system, you must reset the value for the PHYSICALPAGES parameter and rerun AUTOGEN. Follow these steps:

1. Shut down the system and perform a conversational bootstrap operation.

2. At the SYSBOOT> prompt, enter the following command:
   ```
   SYSBOOT> SET PHYSICALPAGES DEFAULT
   ```

3. When the system comes up, invoke AUTOGEN from the SAVPARAMS phase to reboot the system and set the correct PHYSICALPAGES value for the new system:

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

Secondary Page and Swap Files Deleted During MicroVMS Version 4.4 Upgrade

The MicroVMS Version 4.4 upgrade procedure will delete all secondary page and swap files in SYS$SYSTEM on MicroVMS system disks, and AUTOGEN will create primary files of appropriate size.

3.2.22 Multiuser License Kit—Must Be Re-installed

Because MicroVMS Version 4.4 is packaged as an upgrade that replaces all system executable images, your multi-user license kit must be re-installed after upgrading your current MicroVMS system to Version 4.4. If you initially purchased MicroVMS Version 4.1M or Version 4.2, your existing multiuser license kit will not install correctly on MicroVMS Version 4.4. It must be upgraded using a conversion procedure supplied with MicroVMS Version 4.4.

After completing the installation of MicroVMS Version 4.4, invoke the conversion procedure from the SYSTEM account with the following command.

   ```
   $ @SYS$UPDATE:CVTLICENSE
   ```

The procedure will prompt you for a device on which to read your present license kit. It will read the kit, make the necessary changes, and write a new copy of the converted kit, which you may then install using VMSINSTAL.

The conversion is necessary only for multi-user licesne kits that were shipped with MicroVMS Version 4.1M and Version 4.2. License kits shipped with Version 4.3 and Version 4.4 do not require conversion.

3.2.23 Authorize Utility Error Messages

Make the following corrections to the *VAX/VMS Authorize Utility Reference Manual*. These corrections will be incorporated in the next revision of the manual.

- Page AUTH–2: the summary of AUTHORIZE commands should include the following qualifiers:

      /ASTLM (for the ADD command)
      /GENERATE_PASSWORD (for the MODIFY command)

- Page AUTH–11: in Table AUTH–2, the /FLAG=[NO]PWDEXPIRED function should read /FLAG=[NO]PWD_EXPIRED. Please include the underscore (_).

- On page AUTH-13: the /PWDEXPIRED and /PWDLIFETIME qualifiers should appear as /[NO]PWDEXPIRED and /[NO]PWDLIFETIME, respectively.

- Page AUTH–14: in the description of the /UIC qualifier, the documentation states that the value of the member number must be in the range of 0–1777776. The correct range is 0–177776.

- Pages AUTH–21, AUTH–37, AUTH–42: the documentation states that the rights identifier values must be in the range 32,768 to 268,435. Note that user-defined identifiers must be in the range 65,536 to 268,435,455. Identifier values of less than 65,536 are reserved.

- Tables AUTH–2 and AUTH–4: the recommended values for process resource limits should read as follows:

Limit	Value
ASTLM	24
BIOLM	18
BYTLM	8192
ENQLM	30
PGFLQUOTA	12800
WSDEFAULT	200
WSQUOTA	500
WSEXTENT	1000

3.2.24 *VAX/VMS Authorize Utility Reference Manual*—Error Messages

The Authorize Utility has the following error messages that have not previously been documented.

BADNODFORM,   improper node::remoteuser format

Facility: AUTHORIZE, Authorize Utility

Explanation: You specified the format for the remote node and user incorrectly. The correct format consists of a node name, a pair of colons, and the user name of the remote user. A node name may consist of 1–6 alphanumeric characters and must contain at least one alphabetic character. If you use a wildcard character for either the node or user, you must still include the colons.

User Action: Reenter your command with the correct format.

BADUSR,   username does not exist

Facility: AUTHORIZE, Authorize Utility

Explanation: The user name you specified does not exist in the system user authorization file (SYSUAF.DAT).

User Action: Correct the user name and reenter your command. You can display the records in the user authorization file by using the AUTHORIZE command SHOW.

CLIWARNMSG,   Warning: /CLITABLES field may need to reflect changes to /CLI field

Facility: AUTHORIZE, Authorize Utility

Explanation: If you modify the command language interpreter (CLI) field of a record in the system user authorization file, you may have to modify the CLITABLES field to reflect the change. If you have set the CLI field to DCL or MCR, however, the CLITABLES field defaults to the correct value.

User Action: If you have changed the CLI field to a value other than DCL or MCR, use the AUTHORIZE command MODIFY/CLITABLES to set the CLITABLES field to the corresponding tables. Refer to the description of the LOGIN procedure in the *VAX/VMS DCL Dictionary* for further information about specifying CLI tables.

CMDTOOLONG,   command line exceeds maximum length

Facility: AUTHORIZE, Authorize Utility

Explanation: The length of your command, after any symbols and logical names have been expanded, exceeds the maximum allowable length.

User Action: Reenter a shorter form of the command.

EXTRAPARM,   superfluous parameter detected

Facility: AUTHORIZE, Authorize Utility

Explanation: You have specified too many parameters in the command line. The extra parameter is identified in the message.

User Action: Reenter your command without the excess parameter.

GRANTERR, unable to grant identifier 'id-name' to 'user name'

Facility: AUTHORIZE, Authorize Utility

Explanation: The specified identifier cannot be granted to the specified user. This message should be accompanied by a second message showing the specific reason why the identifier could not be granted.

User Action: Correct the condition identified by the second message and reenter your command.

GRANTMSG, identifier 'id-name' granted to 'user name'

Facility: AUTHORIZE, Authorize Utility

Explanation: The specified general identifier has been granted to the specified user. The user has access to all of the rights associated with the identifier.

User Action: None.

HELPERR, error finding or outputting HELP information

Facility: AUTHORIZE, Authorize Utility

Explanation: An error occurred trying to access the AUTHORIZE HELP file.

User Action: Check that the AUTHORIZE HELP file, by default named UAFHLP.HLB, is located in the proper directory and is not protected against read access.

IDOUTRNG, identifier value is not within legal range

Facility: AUTHORIZE, Authorize Utility

Explanation: The value you specified for an identifier is not within the permissible range. A general identifier may have an integer value between 32,768 and 268,435,455. A UIC identifier takes a value in standard UIC format.

User Action: Reenter your command with an identifier value that is within the permissible range.

INVCMD, invalid command

Facility: AUTHORIZE, Authorize Utility

Explanation: The command you have entered is not a valid AUTHORIZE command.

User Action: Refer to the *VAX/VMS Authorize Utility Reference Manual* for a description of the command you are trying to use and then reenter the command correctly.

INVUSERNAME,   username syntax error

Facility: AUTHORIZE, Authorize Utility

Explanation: The user name you specified is invalid due to incorrect syntax. If you are adding a new user name to the system user authorization file with the AUTHORIZE command ADD, the new user name may be 1-12 alphanumeric characters, and it may include underscores.

User Action: Correct the user name and reenter your command.

INVUSERSPEC,   error in user specification

Facility: AUTHORIZE, Authorize Utility

Explanation: Your command included an incorrect user specification. In a user specification, you may use a numeric UIC format (for example, [007,007]), an alphanumeric format (for example, [COMPOSERS,HAYDN]), or a user name (for example, HAYDN). You can use wildcards to specify multiple users. Refer to the *VAX/VMS Authorize Utility Reference Manual* for specific syntax rules for the command you are using.

User Action: Correct the user specification and reenter your command.

NAFADDERR,   unable to add record to NETUAF.DAT

Facility: AUTHORIZE, Authorize Utility

Explanation: The record you specified could not be added to the network user authorization file (NETUAF.DAT). This message should be accompanied by a VAX RMS message that identifies the specific reason for the error. For example, this error occurs if you try to add a record authorizing a remote user to access more than one local account. Each user at a remote node is allowed access to the files of only one user on the local node.

User Action: If possible, correct the condition identified by the RMS message and reenter your command. Otherwise, examine the network user authorization file to determine why the record could not be added. You can display the contents of the file by using the AUTHORIZE command SHOW/PROXY. You can write the contents of NETUAF.DAT to a listing file by using the AUTHORIZE command LIST/PROXY. If you want to delete a current record from NETUAF.DAT in order to add a new one, use the AUTHORIZE command REMOVE/PROXY.

NAFAEX, NETUAF.DAT already exists

Facility: AUTHORIZE, Authorize Utility

Explanation: A network user authorization file (NETUAF.DAT) already exists for the local node.

User Action: If you want to create a new network user authorization file, either delete or rename the current one (if you have sufficient privilege to do so). Once the current file has been deleted or renamed, reenter the AUTHORIZE command CREATE/PROXY. Note that you must have sufficient privilege to create a new file.

NAFCREERR, unable to create NETUAF.DAT

Facility: AUTHORIZE, Authorize Utility

Explanation: A network user authorization file (NETUAF.DAT) could not be created. This message should be accompanied by a VAX RMS message that identifies the specific reason why the file could not be created. For example, this error occurs if you do not have sufficient privilege to create the file.

User Action: Correct the condition identified by the RMS message and reenter your command.

NAFDNE, NETUAF.DAT does not exist

Facility: AUTHORIZE, Authorize Utility

Explanation: A network user authorization file (NETUAF.DAT) does not exist on the local node.

User Action: If you have sufficient privilege, use the AUTHORIZE command CREATE/PROXY to create a network user authorization file. Then you can add records to the file by using the AUTHORIZE command ADD/PROXY.

NAFDONEMSG, network authorization file modified

Facility: AUTHORIZE, Authorize Utility

Explanation: The network user authorization file (NETUAF.DAT) has been modified to reflect the change directed by your command.

User Action: None.

NAFNOMODS,   no modifications made to network authorization file

Facility: AUTHORIZE, Authorize Utility

Explanation: No modifications have been made to the network user authorization file (NETUAF.DAT).

User Action: None.

NAFUAEERR,   entry already exists in NETUAF.DAT

Facility: AUTHORIZE, Authorize Utility

Explanation: The record you have tried to add to the network user authorization file is already in the file; it has not been duplicated.

User Action: None.

NAONAF,   unable to open NETUAF.DAT

Facility: AUTHORIZE, Authorize Utility

Explanation: The network user authorization file (NETUAF.DAT) could not be opened. This message should be accompanied by a VAX RMS message that identifies the specific reason for the error. Possible reasons are insufficient privilege, file protection violation, or location of the file in the wrong directory.

User Action: If you do not have sufficient privilege to open NETUAF.DAT, there is nothing you can do except to ask a privileged user, such as your system manager, to access the file for you. If you do have sufficient privilege, make sure the file is located in the proper directory and is not protected against read or write access. Then reenter your command.

NETLSTMSG,   listing file NETUAF.LIS complete

Facility: AUTHORIZE, Authorize Utility

Explanation: The contents of the network user authorization file (NETUAF.DAT) have been written to the listing file named NETUAF.LIS.

User Action: None.

NOARG,   missing argument for option

Facility: AUTHORIZE, Authorize Utility

Explanation: You specified a qualifier without a required argument.

User Action: Reenter your command and include the required argument.

NODTOOBIG, node name too long

Facility: AUTHORIZE, Authorize Utility

Explanation: VAX/VMS node names cannot exceed six characters. A node name may consist of 1–6 alphanumeric characters; at least one character must be alphabetic.

User Action: Check the node name and reenter your command with the correct name.

NOGRPWILD, wild card group numbers not allowed

Facility: AUTHORIZE, Authorize Utility

Explanation: Wildcard characters are not allowed in the UIC group number field for the command you entered.

User Action: Reenter your command with a specific UIC group number instead of a wildcard character.

NOIDNAM, no ID name was specified

Facility: AUTHORIZE, Authorize Utility

Explanation: The command you entered requires that you include an identifier name.

User Action: Check the *VAX/VMS Authorize Utility Reference Manual* for the syntax rules regarding identifier names for the command you want to use. Then reenter the command including an identifier name.

NOTIDFMT, id name parameter does not translate to ID format

Facility: AUTHORIZE, Authorize Utility

Explanation: The identifier name that you specified does not translate to a corresponding value in general identifier format. Identifier name values translate to either general identifier format or UIC format. General identifier names may be 1 through 31 alphanumeric characters and are stored with an integer value in the range of 32,768 to 268,435,455. General identifiers are created by the AUTHORIZE command ADD/IDENTIFIER.

When you use the AUTHORIZE command GRANT/IDENTIFIER, the first identifier you specify must be in general identifier format. In other words, you cannot grant a UIC-format identifier to another UIC-format identifier.

User Action: Determine why the identifier name does not translate. You can display an identifier name and its corresponding value with the AUTHORIZE command SHOW/IDENTIFIER. To change the value of an identifier name, use the AUTHORIZE command MODIFY/IDENTIFIER.

NOTUICFMT,   user id parameter does not translate to UIC format

Facility: AUTHORIZE, Authorize Utility

Explanation: The user specification in your command does not translate to a UIC identifier (an identifier in UIC format).

User Action: Determine why the user specification does not translate. You can display user names and their corresponding UIC values by using the AUTHORIZE command SHOW.

NOUSERNAME,   missing username

Facility: AUTHORIZE, Authorize Utility

Explanation: The command you are using requires a user name. A user name is the member name from the alphanumeric form of a user's UIC (user identification code).

User Action: Reenter your command including a user name.

NOUSERSPEC,   missing user specification

Facility: AUTHORIZE, Authorize Utility

Explanation: The command you are using requires a user specification. A user specification may be a user name (for example, CAESAR), or a user identification code (for example, [100,44]).

User Action: Reenter your command including a user specification.

PREMMSG,   record removed from NETUAF.DAT

Facility: AUTHORIZE, Authorize Utility

Explanation: The record specifed in the AUTHORIZE command REMOVE/PROXY has been removed from the network user authorization file.

User Action: None.

PWDNCH,   password not changed

Facility: AUTHORIZE, Authorize Utility

Explanation: An error occurred using the random password generator to generate an account password.

User Action: None.

PWDNOL,   password not on list; try again

Facility: AUTHORIZE, Authorize Utility

Explanation: The password you specified was not one of those listed.

User Action: Select another password and try again.

RDBADDERR,   unable to add 'id-name' to RIGHTSLIST.DAT

Facility: AUTHORIZE, Authorize Utility

Explanation: The identifier name you specified could not be added to the rights database file (RIGHTSLIST.DAT). This message should be accompanied by a VAX RMS message that identifies the specific reason for the error. Most likely, the identifier name already exists in the rights database file. Duplicate identifier names are not allowed in the rights database file.

User Action: Correct the condition identified by the RMS message and reenter your command. If you want to change the name of an identifier in the rights database file, use the AUTHORIZE command MODIFY/IDENTIFIER.

RDBADDERRU,   unable to add 'id-name' value: '[UIC]' to RIGHTSLIST.DAT

Facility: AUTHORIZE, Authorize Utility

Explanation: The specified identifier name and its corresponding user identification code (UIC) could not be added to the rights database file (RIGHTSLIST.DAT). This message should be accompanied by a VAX RMS message that identifies the specific reason for the error. Most likely, the identifier name already exists in the rights database file. Duplicate identifier names are not allowed in the rights database file.

This error also occurs if you copy a record in the system user authorization file (SYSUAF.DAT) without specifying a new UIC value for the copy. By default, an identifier name and corresponding UIC value for the new record are written to the rights database file (RIGHTSLIST.DAT); if the UIC has not been changed, it will conflict with the UIC of the original record, and a 'duplicate identifier' error results.

User Action: Correct the condition identified by the RMS message and reenter your command. If you want to change the UIC value of an identifier in the rights database file, use the /VALUE qualifier with the AUTHORIZE command MODIFY/IDENTIFIER. If you copy a record in the system user authorization file, and you want an identifier for the new record to be added to the rights database file, use the /UIC qualifier with the AUTHORIZE command COPY.

RDBADDERRV,   unable to add 'id-name' value: 'hex code' to RIGHTSLIST.DAT

Facility: AUTHORIZE, Authorize Utility

Explanation: The specified identifier name and its corresponding integer value (expressed as an 8-bit hexadecimal code) could not be added to the rights database file (RIGHTSLIST.DAT). This message should be accompanied by a VAX RMS message that identifies the specific reason for the error. Most likely, the identifier name or value already exists in the rights database file. Duplicate identifier names or values are not allowed in the rights database file.

User Action: Correct the condition identified by the RMS message and reenter your command. If you want to change the value of an identifier in the rights database file, use the /VALUE qualifier with the AUTHORIZE command MODIFY/IDENTIFIER.

RDBADDMSG,   identifier 'id-name' value: 'hex code' added to RIGHTSLIST.DAT

Facility: AUTHORIZE, Authorize Utility

Explanation: A general identifier with the specified name and value has been added to the rights database file (RIGHTSLIST.DAT).

User Action: None.

RDBADDMSGU,   identifier 'id-name' value: '[UIC]' added to RIGHTSLIST.DAT

Facility: AUTHORIZE, Authorize Utility

Explanation: A UIC identifier with the specified name and value has been added to the rights database file (RIGHTSLIST.DAT).

User Action: None.

RDBCREERR,   unable to create RIGHTSLIST.DAT

Facility: AUTHORIZE, Authorize Utility

Explanation: The rights database file, named RIGHTSLIST.DAT, could not be created. This message should be accompanied by a VAX RMS message that identifies the specific reason for the error. For example, you cannot create another rights database file if one already exists, unless you first delete or rename the original file.

User Action: Correct the condition identified by the RMS message and reenter your command. If you want to create a new rights database file, either delete or rename the current one (if you have sufficient privilege to do so). Once the current file has been deleted or renamed, reenter your command.

RDBDONEMSG, rights database modified

Facility: AUTHORIZE, Authorize Utility

Explanation: The rights database file (RIGHTSLIST.DAT) has been modified.

User Action: None.

RDBMDFYERR, unable to modify identifier 'id-name'

Facility: AUTHORIZE, Authorize Utility

Explanation: The specificied identifier could not be modified. This message should be accompanied by a VAX RMS message that identifies the specific reason for the error.

User Action: Correct the condition identified by the RMS message and reenter your command.

RDBMDFYERRU, unable to modify identifier '[UIC]'

Facility: AUTHORIZE, Authorize Utility

Explanation: The specified UIC identifier could not be modified. This message should be accompanied by a VAX RMS message that identifies the specific reason for the error.

User Action: Correct the condition identified by the RMS message and reenter your command.

RDBMDFYMSG, identifier 'id-name' modified

Facility: AUTHORIZE, Authorize Utility

Explanation: The record for the specified identifier in the rights database file has been modified according to the AUTHORIZE command MODIFY/IDENTIFIER.

User Action: None.

RDBNOMODS, no modifications made to rights database

Facility: AUTHORIZE, Authorize Utility

Explanation: The rights database file (RIGHTSLIST.DAT) was not modified.

User Action: None.

RDBREMERR,   unable to remove 'id-name' from RIGHTSLIST.DAT

Facility: AUTHORIZE, Authorize Utility

Explanation: The specified identifier could not be removed from the rights database file (RIGHTSLIST.DAT). This message should be accompanied by a VAX RMS message that identifies the specific reason for the error.

User Action: Correct the condition identified by the RMS message and reenter your command.

RDBREMMSG,   identifier 'id-name' value: 'hex code' removed from RIGHTSLIST.DAT

Facility: AUTHORIZE, Authorize Utility

Explanation: The general identifier with the specified name and hexidecimal value has been removed from the rights database file (RIGHTSLIST.DAT).

User Action: None.

RDBREMMSGU,   identifier 'id-name' value: '[UIC]' removed from RIGHTSLIST.DAT

Facility: AUTHORIZE, Authorize Utility

Explanation: The UIC identifier with the specified name and user identification code has been removed from the rights database file (RIGHTSLIST.DAT).

User Action: None.

REVOKEERR,   unable to revoke identifier 'id-name' from 'user name'

Facility: AUTHORIZE, Authorize Utility

Explanation: The specified identifier could not be revoked from the specified user.

User Action: Make sure that the user has been granted the identifier you are trying to revoke. Use the AUTHORIZE commands SHOW/IDENTIFIER/FULL or LIST/IDENTIFIER/FULL to display an identifier and the users who hold it.

REVOKEMSG,   identifier 'id-name' revoked from 'user name'

Facility: AUTHORIZE, Authorize Utility

Explanation: The specified identifier has been revoked from the specified user. The user no longer has the rights associated with the identifier.

User Action: None.

RLSTMSG,   listing file RIGHTSLIST.LIS complete

Facility: AUTHORIZE, Authorize Utility

Explanation: The contents of the rights database file (RIGHTSLIST.DAT) have been written to the listing file named RIGHTLIST.LIS.

User Action: None.

SHOWERR,   unable to complete show command

Facility: AUTHORIZE, Authorize Utility

Explanation: The AUTHORIZE command SHOW could not be completed. This message should be accompanied by a VAX RMS message that identifies the specific reason for the error.

User Action: Correct the condition identified by the RMS message and reenter your command.

SYSMSG2,   Error code 'hex code' not found

Facility: AUTHORIZE, Authorize Utility

Explanation: The $GETMSG system service could not find a corresponding message for the specified error code, which probably indicates that the code is incorrect. Since an incorrect error code obviously should not be generated, this message probably indicates an internal software error.

User Action: Submit a Software Performance Report (SPR) that describes the conditions leading to the error.

WLDNOTALWD,   wild card user specs not allowed

Facility: AUTHORIZE, Authorize Utility

Explanation: Wildcard characters are not allowed in the user specification for the command you are using.

User Action: Reenter your command without using wildcard characters.

ZZPRACREN,   proxies to 'user name' renamed

Facility: AUTHORIZE, Authorize Utility

Explanation: Proxy access records for the specified user have been renamed to the new user name. When a user name in the system user authorization file (SYSUAF.DAT) is renamed, any records in the network authorization file (NETUAF.DAT) for the original user name are automatically renamed to the new user name.

User Action: None.

ZZSYSPWD,   system password modified

Facility: AUTHORIZE, Authorize Utility

Explanation: The system password has been changed to the password directed by your command.

User Action: None.

3.3   Application Programmer Information

This section contains information about the MicroVMS Version 4.4 release that pertains to application programmers.

3.3.1 *MicroVMS Programmer's Manual*—Change in Focus

The focus of the *MicroVMS Programmer's Manual* has changed to supply programmers in **any** programming language with guidelines for using the development tools available with MicroVMS; therefore, 'FORTRAN ' in the title has been eliminated. However, the manual still has a FORTRAN orientation, all examples are in FORTRAN, and the programmer is assumed to have a reading knowledge of the language.

The Debugger section of the New and Changed Features section of this manual contains most of the new and changed programming features for Version 4.4.

3.3.2 *VAX/VMS Linker Reference Manual*—Correction

Please note the following corrections to the *VAX/VMS Linker Reference Manual*:

- On page LINK-1, the default for the command qualifier /[NO]USERLIBRARY should read /USERLIBRARY=ALL. This correction will be incorporated in the next revision of the manual.

- The reference to Section 6.3.6.2 on page LINK-31 (third list item at the top of the page) is incorrect. The correct reference is to Section 5.3.6.2.

- The reference to Appendix A on page LINK-61 (fourth line, second paragraph from the bottom) is incorrect. The correct reference is to Chapter 6, which contains information on the VAX Object Language.

- Example 3 on page LINK-129 is incorrect. The example should read as follows:

```
$ LINK LAMAR,SYS$INPUT/OPTION
  GRABLE/SHAREABLE
```

Note, GRABLE is the name of a shareable image file and not an options file as previously documented. The above correction also applies to Example 2 on page LINK–142.

3.3.3 *Run-Time Library Routines Reference Manual*—Additions

Documentation has been added to Chapter 3 of the *VAX/VMS Run-Time Library Routines Reference Manual* concerning user-written exit handlers for screen management routines. This documentation explains why pasteboards and virtual keyboards cannot be deleted from within a user-written exit handler.

3.3.4 Run-Time Library Screen Management Facility—Restriction

Due to changes made to the Screen Management Facility, the following restriction now applies to the routines SMG$SET_BROADCAST_TRAPPING, SMG$ENABLE_UNSOLICITED_INPUT, and SMG$SET_OUT_OF_BAND_ASTS. For AST routines written in a language that does not support optional parameters (for example VAX BASIC), all system parameters must be specified. This restriction is illustrated in the example for the SMG$DISABLE_BROADCAST_TRAPPING routine in the *VAX/VMS Run-Time Library Routines Reference Manual*.

3.3.5 Debugger—Problems and Restrictions

Note the following restrictions and problems that apply to the Debugger.

3.3.5.1 Debugging Shareable Images—Restriction

Support for debugging shareable images is new with Version 4.4 and is described in Chapter 4 of the *VAX/VMS Debugger Reference Manual* and in the *MicroVMS Programming Support Manual*.

There is one restriction you should be aware of when debugging a shareable image: it must have been linked with the /DEBUG qualifier. If the image was not linked with the /DEBUG qualifier, you will still be able to "SET IMAGE" to that image, but then you may obtain incorrect results.

The behavior is summarized in the following table for an arbitrary shareable image, X.

Command	Effect
LINK/SHARE/DEBUG X	You can SET IMAGE to X and debug it as documented.
LINK/SHARE X	You can SET IMAGE to X, but you may obtain incorrect results when you try to debug it (when using SET BREAK, EXAMINE, and so on). This problem will be corrected in a future release of the debugger.
LINK/SHARE/NOTRACE	You cannot SET IMAGE to X and, therefore, cannot debug it with debugger commands.

3.3.5.2 Using the Debugger on a MicroVAX Workstation—Problem

When you invoke the debugger on a MicroVAX workstation, the debugger comes up in its own window. There is a problem with the handling of CTRL/Y when the debugger is running in its own MicroVAX window. CTRL/Y is ignored when the keyboard is attached to the debugger window. To make CTRL/Y take effect, attach the keyboard to the window from which you invoked the debugger (by pointing at that window with the mouse), then press CTRL/Y.

This problem will be corrected in a future release.

3.3.5.3 Debugging SMG Programs—Problem

The debugger now uses the MicroVMS Screen Management Facility (SMG) to implement screen mode. If your program also calls SMG routines, and you debug it with the debugger running on the same terminal, there will probably be interference between your program and the debugger.

To avoid this problem, debug the program using two terminals. The technique is described in Appendix D of the *VAX/VMS Debugger Reference Manual*.

3.3.5.4 Debugger Changes Affecting Compatibility with Earlier Versions

This section notes any changes in the debugger for Version 4.4 that are incompatible with Version 4.2 or earlier versions. All changes concern the display of various windows in screen mode.

New Display Window Definitions

Changes to the built-in window definitions and the addition of a PROMPT predefined display have caused some incompatibilities with earlier versions of the debugger. If you use built-in window definitions, such as H2, in your debugger initialization file or in your own command or key definitions, then you should be aware of the following changes:

- Previously, the bottom sixth of the screen (lines 21-24 on a VT100 or VT200 series terminal) could not be used for defining windows. That area was reserved for the debugger prompt, debugger input, debugger diagnostic messages, and program output. Now, windows can occupy any part of the screen, and the new

PROMPT display shows the debugger prompt, debugger input, and program output.

- The boundaries of the built-in windows have been redefined to cover the greater usable screen height. For example, on a VT100 or VT200 series terminal, FS (full screen) now covers lines 1-24, H1 lines 1-12, H2 lines 13-24, and so on. And a new symbol prefix, S, denotes a multiple of one sixth of the screen.

- The PROMPT display occupies window S6 by default (bottom sixth of screen). Note that, to avoid confusion, the PROMPT display is always on top of the display "pasteboard" and, therefore, will hide the part of any display that overlaps the PROMPT window.

- By default, the OUT display is now at S45 (not, as previously, at H2), so it will not be hidden by the PROMPT display. And the keypad keys that manipulate windows have been redefined so that no display is positioned behind S6.

If your debugger initialization file contains DISPLAY or SET DISPLAY commands to locate displays near the bottom of the screen (for example, at H2, T3, or Q34) you may want to modify these window definitions so the displays will not be hidden.

New Keypad Key Definitions

The debugger's built-in definitions for keypad keys KP7 and MINUS have been changed to accommodate changes in the built-in window definitions. The new key definitions are as follows:

```
KP7         = DISPLAY SRC AT LH1, INST AT RH1, OUT AT S45, PROMPT AT S6
GOLD KP7    = DISPLAY INST AT LH1, REG AT RH1, OUT AT S45, PROMPT AT S6
BLUE KP7    = Not defined
MINUS       = DISPLAY %NEXTDISP AT S12345
GOLD MINUS  = Not defined
BLUE MINUS  = DISPLAY SRC AT H1, OUT AT S45, PROMPT AT S6 (this is the
                      default for high-level languages)
```

Refer to the previous section for information about the new window definitions.

Register Display

The predefined register display (REG) has been reformatted to take advantage of the new window capabilities. REG is now a square display that fits in one of the quarters of the screen (for example, the top left-hand window LH1 or the top right-hand window RH1). If your debugger initialization file had a command like "DISPLAY REG AT Q3", then you may want to change it to something like "DISPLAY REG AT RH1" to accommodate the reshaped register display.

Display Kind "NORMAL" Renamed to "OUTPUT"

The display kind "NORMAL" has been renamed "OUTPUT." If your debugger initialization file contains DISPLAY or SET DISPLAY commands that specify a display kind, you may want to change any occurrences of "NORMAL" to "OUTPUT." However, the display-kind name "NORMAL" will continue to be accepted as a synonym for "OUTPUT."

3.3.6 SET HOST/DTE Support—Problems

This section describes problems concerning SET HOST/DTE support.

3.3.6.1 SET HOST/DTE/DIAL Command—DMF-32 Problem

SET HOST/DTE/DIAL does not work with the DMF-32 controller. The problem is that the modem sends a response character to the host when it detects a carrier signal, but the DMF-32 drops any input until it sees the carrier signal.

3.3.6.2 SET HOST/DTE/LOG Command—Log File Problem

There is a known problem with the SET HOST/DTE/LOG command. The log file that is created may include extra linefeed characters. This problem will be corrected in a future update.

3.3.7 *VAX/VMS Command Definition Utility Reference Manual—*Example Correction

The following example is an excerpt from Example CDU-2 in the VAX/VMS Command Definition Utility Reference Manual.

To make this BASIC program execute as described in the documentation, change the following lines (comments describe the changes):

```
200 SUB EXIT_COMMAND !Same as documented.
    !exclude EXTERNAL INTEGER FUNCTION SYS$EXIT
    CALL SYS$EXIT(1% BY VALUE)      !Note addition.

290 SUBEND  !Same as documented.

1   EXTERNAL INTEGER FUNCTION CLI$DCL_PARSE,CLI$DISPATCH !Exclude LIB$GET_INPUT
    EXTERNAL INTEGER FUNCTION SEND_COMMAND,SEARCH_COMMAND,EXIT_COMMAND !Same
    EXTERNAL INTEGER TEST_TABLE,LIB$GET_INPUT       !Note addition.

2   IF NOT CLI$DCL_PARSE(,TEST_TABLE,LIB$GET_INPUT,LIB$GET_INPUT,'TEST> ')
    AND 1%
    THEN GOTO 2  !Note elimination of O above.
```

3.3.8 *VAX Text Processing Utility Reference Manual*—Documentation Changes

The following are corrections to the documentation for VAXTPU.

3.3.8.1 GET_INFO—Restriction

The material in the *VAX Text Processing Utility Reference Manual* does not include a restriction on using the built-in procedure GET_INFO. The following material should be added to the manual's description of the built-in procedure GET_INFO:

Be careful when you write programs that attempt to search one of the lists maintained by VAXTPU. VAXTPU provides only one context for traversing each list. VAXTPU maintains lists of buffers, defined keys, key maps, key-map lists, processes, and windows. You can search a list by using "first," "next," "previous," "current," or "last" as the second parameter to the built-in procedure GET_INFO.

If you create nested loops that attempt to search the same list, the results are unpredictable. For example, a program attempting to search two key-map lists for common key maps may contain the built-in procedure GET_INFO (KEY_MAP, "next", . . .) in a loop within a loop containing GET_INFO (KEY_MAP, "previous", . . .). This creates an infinite loop.

3.3.8.2 VAX BLISS—VAXTPU Example

The VAX BLISS Example TPU-1 in Section 12 of the *VAX/VMS Utility Routines Reference Manual* contains errors. The following example contains corrections to Example TPU-1.

Example 3–1   Sample VAX BLISS Template for Callable VAXTPU

```
! How to declare the VAXTPU routines

external routine
        tpu$FILEIO,
        tpu$HANDLER,
        tpu$INITIALIZE,
        tpu$EXECUTE_INIFILE,
        tpu$EXECUTE_COMMAND,
        tpu$CONTROL,
        tpu$CLEANUP;

! How to declare the VAXTPU literals

external literal
!
! File I/O operation codes
        tpu$k_close,
        tpu$k_close_delete,
        tpu$k_open,
        tpu$k_get,
        tpu$k_put,
```

(Continued on next page)

Example 3–1   (Cont.)   Sample VAX BLISS Template for Callable VAXTPU

```
!
! File access codes
        tpu$k_access,
        tpu$k_io,
        tpu$k_input,
        tpu$k_output,
!
! Item codes
        tpu$k_calluser,
        tpu$k_fileio,
        tpu$k_outputfile,
        tpu$k_sectionfile,
        tpu$k_commandfile,
        tpu$k_filename,
        tpu$k_journalfile,
        tpu$k_options,
!
! Mask for values in options
        tpu$m_recover,
        tpu$m_journal,
        tpu$m_read,
        tpu$m_command,
        tpu$m_create,
        tpu$m_section,
        tpu$m_display,
        tpu$m_output,
!
! Bit positions for values in options
        tpu$v_display,
        tpu$v_recover,
        tpu$v_journal,
        tpu$v_read,
        tpu$v_create,
        tpu$v_command,
        tpu$v_section,
        tpu$v_output,
!
! VAXTPU status codes
        tpu$_nofileaccess,
        tpu$_openin,
        tpu$_inviocode,
        tpu$_failure,
        tpu$_closein,
        tpu$_closeout,
        tpu$_readerr,
        tpu$_writeerr,
        tpu$_success;
```

Example 3-1 (Cont.)   Sample VAX BLISS Template for Callable VAXTPU

```
own
        OPTIONS:          bitvector [32];
! OPTIONS will be passed to VAXTPU

GLOBAL ROUTINE top_level =
BEGIN
!++
! Main entry point of your program
!--
! Your_initialization_routine must be declared as a BPV

local    BPV:    vector[2,long] initial (TPU_INIT,0);! Procedure block

! First establish the condition handler

LIB$ESTABLISH (tpu$handler);

! Call the intialization routine and pass it the address of the BPV
! which has the address of your initialization routine (VAXTPU
! calls this)

tpu$initialize (BPV);

! Use the following call if the options word passed to VAXTPU indicated that
! an initialization file needs to be executed and/or the TPU$INIT_PROCEDURE
! in the section file needs to be executed.

tpu$execute_inifile();

! Let VAXTPU take over.
tpu$control();

! To break out of VAXTPU, use call_user from within a VAXTPU program
! Upon return from tpu$control, the editing session is done

tpu$cleanup();

! Loop and start the sequence over or exit
return tpu$_success;

END;
```

Example 3–1 (Cont.)   Sample VAX BLISS Template for Callable VAXTPU

```
ROUTINE TPU_INIT =
BEGIN
!
!--
own     BPV:     vector[2,long] initial (TPU_IO,0);! Procedure block

Macro
        OUTFILE_NAME= 'OUTPUT.TPU'%,
        COMFILE_NAME= 'TPUINI.TPU'%,
        SECFILE_NAME= 'SYS$LIBRARY:EVESECINI.TPU$SECTION'%,
           FILE_NAME= 'FILE.TPU'%;

! Set VAXTPU options I want to enable

OPTIONS[tpu$v_display] = 1;
OPTIONS[tpu$v_section] = 1;
OPTIONS[tpu$v_create]  = 1;
OPTIONS[tpu$v_command] = 1;
OPTIONS[tpu$v_recover] = 0;
OPTIONS[tpu$v_journal] = 0;
OPTIONS[tpu$v_read] = 0;
OPTIONS[tpu$v_output] = 1;

begin              !Just for BIND
bind
! Set up item list to pass back to VAXTPU to tell it what to do
! VAXTPU calls me back later
ITEMLIST = uplit byte (

!buffer length,         item code,         buffer address,  return address

 word (4),    word (tpu$k_options),    long (OPTIONS),   long (0),
 word (4),    word (tpu$k_fileio),     long (BPV),       long (0),
 word (%charcount(outfile_name)),
           word (tpu$k_outputfile),  long (uplit(%ascii(outfile_name))),
                                                    long (0),
 word (%charcount(comfile_name)),
           word (tpu$k_commandfile), long (uplit(%ascii(commandfile_name))),
                                                    long (0),
 word (%charcount(file_name)),
           word (tpu$k_filename),     long (uplit(%ascii(file_name))),
                                                    long (0),
 word (%charcount(secfile_name)),
           word (tpu$k_sectionfile), long (uplit(%ascii(secfile_name))),
                                                    long (0),
 long (0) );
return ITEMLIST;
end;
END;                       ! End of routine TPU_INIT
```

Example 3–1 (Cont.)   Sample VAX BLISS Template for Callable VAXTPU

```
GLOBAL ROUTINE TPU_IO (P_OPCODE, FILE_BLOCK, DATA: ref block [,byte]) =
BEGIN
!
!--
local
        item: ref block [3,long],       ! Item list entry
        status;

! Look at the opcode (operation) that VAXTPU wants me to perform
! and if I don't want to do it, just call it back
! if (..P_OPCODE NEQ tpu$k_open)
!       then
!       return (tpu$fileio (.p_opcode, .file_block, .data));
!
! Else set what operation to do

selectone ..P_OPCODE of
set
[tpu$k_open]:
! Time to open a file
!
    begin
        item = .data;           ! Point to the FILENAME item list entry

        end;
        return tpu$_success;    ! End of tpu$k_open
    end;
[tpu$k_get]:                    ! If none exists, then no data
! Time to read a record
    begin
    end;
[tpu$k_put]:                    ! Time to write a record
    begin

        return tpu$_success;
    end;
[tpu$k_close]:                  !Time to close a file
    begin

        return tpu$_success;
    end;
[tpu$k_close_delete]: lib$stop (..p_opcode);
[otherwise]: lib$stop (..p_opcode);
tes;

return tpu$_success;
END;                    ! End of routine TPU_IO
```

3.3.9 Run-Time Library Support of VAX BASIC—USEROPEN Problem

In MicroVMS Version 4.4, the Run-time library support for VAX BASIC clears the RAB$V_WAT and RAB$V_TMO bits of the RAB$L_ROP field. This occurs each time a FIND or GET is executed. Applications that use a USEROPEN to set either of these bits and expect them to stay set will not work properly under MicroVMS Version 4.4. This problem will be fixed in a future release of VAX BASIC.

3.3.10 PL/I PRINT FILE Format—Line Feed Change

Prior to MicroVMS Version 4.2, VAX PL/I generated an extra line feed immediately following a PAGE directive for PRINT format files. This extra line is no longer generated when PL/I programs are run on MicroVMS versions later than 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 DIGITAL recommends that /NOFEED be used for printing formatted files, this change should allow PL/I PRINT files that are generated on a MicroVMS Version 4.2 or later system 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 upon 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.

3.3.11 VAX Ada—Compatibility Problem With MicroVMS Debugger

If users at your site heavily use VAX DEBUG with VAX Ada programs, you may want to consider delaying installation of this version of VMS until you have obtained the next maintenance release of the Ada compiler after version 1.2.

This release of VMS includes some improvements in the debugger and linker that interfere with the debugging of Ada programs compiled with VAX Ada Version 1.2, or earlier. The problem is that arrays and records whose layout is only known at run time cannot be debugged.

3.4 System Programmer Information

This section contains information of interest to the system programmer.

3.4.1 *VAX MACRO and Instruction Set Reference Manual*—Additional Information on 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
```

3.4.2 $PRDEF Symbols—Documentation Addition

The documentation for the $PRDEF symbols was omitted from previous release notes. 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-coded, 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 percent 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.

3.4.3 CPUDISP Macro—Format Restriction

One format previously supported by the CPUDISP macro prior to Version 4.4 is no longer allowed. An example taken from the XFDRIVER follows.

Old CPUDISP invocation:

```
CPUDISP <DR_780,DR_750,DR_730,DR_790> ; * Dispatch on CPU type *
```

The new CPUDISP invocation must be in the form of 2-tuples, where a 2-tuple is the CPU designator (for example 780, UV1, etc.) and the macro label that begins the code specific to that CPU (for example DR_780).

New CPUDISP invocation:

```
CPUDISP <-
<780,DR_780>,-
<750,DR_750>,-
<730,DR_730>,-
<790,DR_790>-
> ; * Dispatch on CPU type *
```

Index

A

Accounting Utility (ACCOUNTING), 3-8
  abbreviated qualifier values, 3-9
ACL protection, 3-14
ANALYZE/RMS_FILE, 2-10
Authorize Utility (AUTHORIZE), 2-5, 3-17
  AUTOLOGIN flag, 2-6
  /ACCESS qualifier, 2-5
  /PWDEXPIRED qualifier, 3-17
  /PWDLIFETIME qualifier, 3-17
  /secondary passwords, 2-6

B

BASIC error messages, 2-13
Batch/Print facility, 2-4

C

Callable interface
  for VAXTPU, 2-3
CDU (Command Definition Utility), 3-35
Cyclic Redundancy Check (CRC)
  MACRO, 3-43

D

DCL Command, 2-2
  MAIL/SELF, 3-3
Debugger
  new features, 2-9
  problems and restrictions, 3-32

DECnet–VAX
  DTS/DTR, 3-9
Default file type
  for VAXTPU section files, 2-2
Default key map
  in EVE interface, 2-3
DEFINE/FORM command, 2-4

E

EDT Keypad Emulator
  changing the the default interface, 2-7
Error Log Utility, 2-11
EVE (Extensible VAX Editor)
  changing the default interface, 2-7
  key maps in, 2-3
  predefined procedures in, 2-3
EVE$CLEAR_KEY predefined EVE
  procedure, 2-3
EVE$INIT_KEY predefined EVE procedure,
  2-3
Extensible VAX Editor
  See EVE

F

File name, 3-5
File ownership, 3-14
File type
  for section files
    change in default, 2-2
FIND Statement, 2-14
Form
  specifying for a queue, 2-4

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 _____
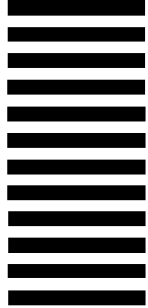
Organization _____

Street _____

City _____ State _____ Zip Code _____
                                          or Country

d i g i t a l

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