digital

VAX/VMS

Release Notes

Version 2.0

Order No. AA-D015C-TE

VAX11

March 1980

This document contains information not included elsewhere in the documentation set. Typically, this information covers software and/or documentation errors that were discovered or changes that were made late in the development cycle, plus hints concerning system installation and operation. This document should be read before the system is installed or used.

VAX/VMS
Release Notes
Version 2.0

Order No. AA-D015C-TE

SUPERSESSION/UPDATE INFORMATION: This revised document supersedes the VAX/VMS Release Notes (Order No. AA-D015B-TE) and VAX/VMS Version 1.6 Release Notes (Order No. AA-J039A-TE).

OPERATING SYSTEM AND VERSION: VAX/VMS V02

SOFTWARE VERSION: VAX/VMS V02

To order additional copies of this document, contact the Software Distribution Center, Digital Equipment Corporation, Maynard, Massachusetts 01754

digital equipment corporation · maynard, massachusetts

First Printing, August 1978
Revised, February 1979
Revised, March 1980

The information in this document is subject to change without notice and should not be construed as a commitment by Digital Equipment Corporation. Digital Equipment Corporation assumes no responsibility for any errors that may appear in this document.

The software described in this document is furnished under a license and may only be used or copied in accordance with the terms of such license.

No responsibility is assumed for the use or reliability of software on equipment that is not supplied by DIGITAL or its affiliated companies.

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

The following are trademarks of Digital Equipment Corporation:

DIGITAL	DECsystem-10	MASSBUS
DEC	DECtape	OMNIBUS
PDP	DIBOL	OS/8
DECUS	EDUSYSTEM	PHA
UNIBUS	FLIP CHIP	RSTS
COMPUTER LABS	FOCAL	RSX
COMTEX	INDAC	TYPESET-8
DDT	LAB-8	TYPESET-11
DECCOMM	DECSYSTEM-20	TMS-11
ASSIST-11	RTS-8	ITPS-10
VAX	VMS	SBI
DECnet	IAS	PDT
DATATRIEVE	TRAX	

CONTENTS

CONTENTS

CONTENTS

CONTENTS

TABLE

1.0 HOW TO INSTALL VAX/VMS V2.0

VAX/VMS V2.0 is a major release of the VAX/VMS operating system. Current customers use the V2.0 distribution kit to upgrade their VAX/VMS V1.6 system to a V2.0 system. New customers use the distribution kit to install a VAX/VMS V2.0 system.

This section is an overview of the information you need to install (or upgrade to) a VAX/VMS V2.0 system. The definitions of and procedures for system installations and system upgrades are found in the VAX-11 Software Installation Guide provided with this kit.

1.1 Identifying the VAX/VMS V2.0 Kit

Two VAX/VMS V2.0 kits are distributed: one is sent to customers who require magnetic tape distribution, the other is sent to customers who require disk distribution. Except for media differences, the kits are identical, and can be used for upgrading a system (current customers) or installing a new system (new customers).

The components of the kits are listed in Tables 1 and 2.

Table 1
VAX/VMS V2.0 Magnetic Tape Distribution Kit

Component	Label	Order Number
SPD 25.1.5	VAX/VMS Operating System Version 2.0	AE-C770E-TE
Console floppy diskette	RX 1/11780 LOCAL CNSL PKG	AS-E633I-YE
DSC floppy diskettes	STAND/ALONE 11780 DSC2 FLP1	AS-E808I-YE
	STAND/ALONE 11780 DSC2 FLP2	AS-J831A-BE
Upgrade floppy diskette	VAX/VMS V2.0 RX01 UPG	AS-J803A-BE
V2.0 system magnetic tape	VMS V20 BIN MT9	BB-D782C-BE
Upgrade magnetic tape	VMS V20 UPGRADE MT9	BB-J806A-BE
Microfiche kit	VAX/VMS V2.0 SRC LST MCRF/XXX	AH-H159B-SE
Documentation kit	VAX/VMS V2.0 DOC KIT	QE001-GZ
RSX-11S floppy diskette	RSX-11S UPDATE RX01	AS-J856A-BE

(continued on next page)

Table 1 (Cont.)
VAX/VMS V2.0 Magnetic Tape Distribution Kit

Component	Label	Order Number
EDTCAI kit	EDTCAI FLOPPY 1	AS-J813A-B3
	EDTCAI FLOPPY 2	AS-J814A-BE
	EDTCAI FLOPPY 3	AS-J815A-BE
	EDTCAI FLOPPY 4	AS-J816A-BE
	EDTCAI FLOPPY 5	AS-J817A-BE
	EDTCAI FLOPPY 6	AS-J818A-BE
	EDTCAI FLOPPY 7	AS-J819A-BE
	EDTCAI COURSE ADMIN GUIDE	AA-J823A-TE
	EDTCAI STUDENT GUIDE	AA-J824A-TE

Table 2
VAX/VMS V2.0 Disk Cartridge Distribution Kit

Component	Label	Order Number
SPD 25.1.5	VAX/VMS Operating System Version 2.0	AE-C770E-TE
System/upgrade disk cartridge	VAX/VMS V2.0 BIN RK07	AY-H020C-BE
Console floppy diskette	RX 1/11780 LOCAL CNSL PKG	AS-E633I-YE
DSC floppy diskettes	STAND/ALONE 11780 DSC2 FLP1	AS-E808I-YE
	STAND/ALONE 11780 DSC2 FLP2	AS-J831A-BE
Upgrade floppy diskette	VAX/VMS V2.0 RX01 UPG	AS-J803A-BE
Microfiche kit	VAX/VMS V2.0 SRC LST MCRF/XXX	AH-H159B-SE
Documentation kit	VAX/VMS V2.0 DOC KIT	QE001-GZ
RSX-11S floppy diskette	RSX-11S UPDATE RX01	AS-J856A-BE
EDTCAI kit	EDTCAI FLOPPY 1	AS-J813A-BE
	EDTCAI FLOPPY 2	AS-J814A-BE
	EDTCAI FLOPPY 3	AS-J815A-BE
	EDTCAI FLOPPY 4	AS-J816A-BE
	EDTCAI FLOPPY 5	AS-J817A-BE
	EDTCAI FLOPPY 6	AS-J818A-BE
	EDTCAI FLOPPY 7	AS-J819A-BE
	EDTCAI COURSE ADMIN GUIDE	AA-J823A-TE
	EDTCAI STUDENT GUIDE	AA-J824A-TE

1.2 VAX/VMS Optional Software Products

This kit does not contain updates to any VAX-11 optional product and, with one exception, contains no information on optional product installation. That exception is the installation procedure for VAX-11 COBOL-74, contained in Section 6.0 of these release notes. However, the VAX-11 Software Installation Guide does provide an overview of optional product installation.

Table 3 lists the names of VAX-11 optional software products, the versions of these products that were supported by VAX/VMS versions V1.5 and V1.6, and the versions of these products that are now supported by VAX/VMS V2.0.

Table 3
VAX-11 Optional Software Products

Product Name	Version Supported by VAX/VMS		
	V1.5	V1.6	V2.0
DECnet-VAX	1.1	1.2	1.3
VAX-11 FORTRAN	1.2	1.3	2.0
VAX-11 BLISS-32	1.0	1.0	2.0
VAX-11 2780/3780 Protocol Emulator	--	1.0	1.1
FORTRAN IV/VAX to RSX Cross Compiler	2.2	2.3	2.3
PDP-11 BASIC-PLUS-2/VAX	1.5	1.5	1.6
PDP-11 DATATRIEVE/VAX	1.1	1.2	2.0
PDP-11 CORAL 66/VAX	3.0	3.0	3.0
VAX-11 PASCAL	--	1.0	1.1
VAX-11 COBOL-74	4.0	4.1	4.1
VAX-11 BASIC	--	--	1.0

1.3 Before Installing the V2.0 Kit

To ensure successful installation of the software:

1. Read the VAX-11 Software Installation Guide.

   The Guide is the basic document for understanding how to install, upgrade, or update VAX/VMS or its optional products.

2. Read this manual, the VAX/VMS Release Notes (Version 2.0).

   This manual contains information you need to ensure a successful installation of (or upgrade to) a VAX/VMS V2.0 system.

3. Examine the VAX-11/780 System Identification Register (SID).

   This read-only register contains the following information:

Bit Field	Contains information on:
31:24	Processor type
23:19	ECO level for functional hardware changes
18:15	ECO level for nonfunctional hardware changes
14:12	Manufacturing location code
11:0	Processor serial number

   VAX/VMS V2.0 cannot be installed unless the processor type field contains a 1 and the ECO level for functional hardware changes field contains a 3. To examine the SID, you must use the console command language to print the contents of the register:

   a. Ensure that the console command floppy diskette is inserted in the floppy diskette drive.

   b. At the system console, press <CTRL/P>

   c. After the >>> prompt, type HALT<RET>

   d. After the >>> prompt, type EXAMINE/I 3E<RET>

   e. A hexadecimal number is displayed. The binary value of bits 31:24 must be equal to 1 (processor is VAX-11/780) and the binary value of bits 23:19 must be equal to 3 (the hardware system is at ECO level 3). If these fields contain any other values, contact DIGITAL field service and report the problem: you cannot install (or upgrade to) V2.0.

   f. If you are a current customer, ensure that your current system is a VAX/VMS V1.6 system. VAX/VMS V2.0 can be upgraded only from a VAX/VMS V1.6 system.

1.4 Installing the VAX/VMS V2.0 Kit

To install (or upgrade to) VAX/VMS V2.0, follow the instructions in the VAX-11 Software Installation Guide that is provided with this kit. Current V1.6 customers will perform a system upgrade, new customers will perform a system installation.

NOTE

Instructions for installing the new EDT Computor-Assisted Instruction course (the EDTCAI kit) are contained in the Introduction to the EDT Editor, Course Administrator Guide provided with this kit.

1.5 After Installing the VAX/VMS V2.0 Kit

New customers should install optional software products after the VAX/VMS V2.0 system has been installed.

After completing the system upgrade, current customers must reassemble and relink any user-coded software that had been linked with the V1.6 global symbol table of the operating system, [SYSEXE]SYS.STB, or that had been assembled using the V1.6 operating system macro library, [SYSLIB]LIB.MLB. For example, all user-coded drivers must be reassembled and relinked to the V2.0 version of SYS.STB. For more information on changes to user-coded software, refer to Section 2.10 of these release notes.

Current customers should reinstall all optional software products after completing the V2.0 upgrade, if these products have not been superseded. If a product has been superseded, an update of that product should be applied after the V2.0 upgrade is completed.

CAUTION

If you have installed VAX-11 COBOL-74 on an earlier version of VAX/VMS, do not reinstall it after upgrading to VAX/VMS V2.0. Continue to use the version of VAX-11 COBOL-74 already installed.

2.0 DIFFERENCES BETWEEN VAX/VMS V2.0 AND VAX.VMS 1.6

Because V2.0 is a major release of the VAX/VMS operating system, there are many changes and new features. The purpose of this section is to highlight some of them: changes and new features are completely described in the documentation for this release.

2.1 New Hardware Supported

VAX/VMS V2.0 now supports the following hardware. For a complete list of hardware supported, consult the Software Product Description that is provided with this kit.

- MA780 multiport memory
- RX02 floppy diskette drive (used as a data device only)
- TS11 magnetic tape transport
- LP25 line printer
- DR780 32-bit parallel interface

2.2 Changes to the Documentation Kit

The VAX/VMS V2.0 Documentation Kit includes several new manuals and revisions to current manuals. To find the information you need, read the VAX-11 Information Directory and Index that is provided with this kit. The information directory describes the intended audience and features of each manual in the documentation kit and provides an index to major VAX/VMS operating system topics.

Each manual that has been revised for VAX/VMS V2.0 contains a summary of technical changes. The purpose of these summaries is to point you quickly to portions of the documentation that were changed or added to describe new V2.0 features.

General information about VAX/VMS V2.0 is found in the following documentation:

- VAX-11 Information Directory and Index
- VAX/VMS Summary Description and Glossary
- VAX/VMS Primer
- VAX/VMS Release Notes

2.3 Command Language and System Messages

This section contains notes on changes and additions to the DIGITAL Command Language (DCL) and system message facility. Documentation describing DCL and system messages is found in:

- VAX/VMS Command Language User's Guide
- VAX/VMS Guide to Using Command Procedures
- VAX/VMS System Messages and Recovery Procedures Manual

2.3.1 **Changes to DCL** - There are many changes and additions to the DCL. Read the "Summary of Technical Changes" in the VAX/VMS Command Language User's Guide for a list of them.

2.3.1.1 **DIRECTORY Command Changes** - The DIRECTORY command has been redesigned. There are some incompatibilites between the way certain qualifiers worked in previous versions of VAX/VMS and the way these qualifiers now work. Furthermore, there are many new qualifiers. Users should study the new command descriptions in Part II of the VAX/VMS Command Language User's Guide.

If you have used the DIRECTORY command in command procedures, some reprogramming of the procedures may be necessary to achieve the desired results.

2.3.1.2 **FORTRAN Command /WORK_FILES Qualifier Now Ignored** - You should no longer specify the number of work files for the compiler to use. The VAX-11 FORTRAN compiler does not use external work files. However, if you currently have command files that specify this qualifier, they need not be changed; the /WORK_FILES qualifier is ignored.

2.3.1.3 **SET TERMINAL Command /LOCAL and /REMOTE Qualifiers Removed** - You can no longer specify the /LOCAL, /NOLOCAL, /REMOTE, or /NOREMOTE qualifiers with the SET TERMINAL command. VAX/VMS determines whether a terminal is remote or local automatically when the terminal comes on line. All tracking of remote or local status is internal. If you specify any of these qualifiers, an error occurs.

If you have used these qualifiers in command procedures, you must remove them to avoid the error messages.

2.3.2 **System HELP Files Have Changed** - The format of the HELP files have changed for V2.0. All the help text files are now maintained in a library, SYS$HELP:HELPLIB.HLB. A program, SYS$SYSTEM:CVTHELP.EXE, is supplied to aid you in converting your site-specific help files to the new format.

When you run CVTHELP, you must have the old help files in your default directory. CVTHELP will ask for the V1.6 file name. Specify only the file name, as a default type of HLP is automatically applied. CVTHELP will also ask for the key name. This is the key name that will be entered into the index of the help library. When asked for the output file name, you may specify a full file specification. A default type of DAT is used if you do not supply one.

After the help files have been converted, you can place them in the help library by using the DCL command:

```
$ LIBRARY /HELP SYS$HELP:HELPLIB helpsource
```

Note that all DIGITAL-supplied help files have been updated to include new features.

2.4 Program Development Tools

This section contains notes on changes and additions to the system components used to develop software. The program development tools are documented in the following manuals:

- VAX-11 Text Editing Reference Manual
- VAX-11 EDT Editor Reference Manual
- VAX-11 Utilities Reference Manual
- VAX-11 SORT User's Guide
- PDP-11 SORT Reference Manual
- VAX-11 MACRO User's Guide
- VAX-11 MACRO Language Reference Manual
- VAX-11 Linker Reference Manual
- VAX-11 Symbolic Debugger Reference Manual

2.4.1 **VAX-11 Symbolic Debugger Features** - The following features are new to the V2.0 debugger. Refer to the VAX-11 Symbolic Debugger Reference Manual for information on the new features.

- Search Rules
- SET SCOPE command
- Special pathnames
- Scope default
- SHOW SCOPE command
- Modes for search rules and typing
- SET TYPE command
- New command modifiers
- SET LOG command
- SET OUTPUT command
- SHOW OUTPUT command
- SHOW LOG command
- Command procedures
- Support for VAX-11 BASIC

2.5 System Services and I/O

This section contains notes on changes and additions to VAX/VMS system services and I/O. The documentation for these topics is found in the following manuals:

- VAX/VMS System Services Reference Manual
- VAX/VMS I/O User's Guide

2.5.1 Use of Colon to Separate Name of Shared Memory from Name of Event Flag Cluster, Mailbox, or Global Section

The colon serves the special function of separating the shared (multiport) memory name from the name of the common event flag cluster, mailbox, or global section. Therefore, any existing programs that include a colon in the name string will not work as they did previously (regardless of whether your system has a multiport memory unit).

For example, assume your program contains these statements:

```
MBXNAM: .ASCID  /PROJECT5:BOX/
        .
        .
        .
        $CREMBX_S       ...,LOGNAM=MBXNAM
```

VAX/VMS will not attempt to create a mailbox named PROJECT5:BOX, but instead will attempt to create a mailbox named BOX in a shared memory unit named PROJECT5. You will get the error message, "Shared memory not connected." To prevent the problem, change programs that include the colon in these name strings.

2.5.2 Magnetic Tape Driver: Tapes Mounted /FOREIGN

In V2.0, the IO$_SKIPRECORD and IO$_SKIPFILE I/O functions on magnetic tapes mounted foreign have changed. Both functions, when operating in the forward direction, now terminate upon encountering two consecutive tape marks.

The status returned in the I/O Status Block (IOSB) is SS$_ENDOFVOLUME. The actual number of records (or files) skipped during the operation prior to the detection of the second tape mark is returned in the high-order two bytes of the first longword of the IOSB. The tape remains positioned between the two tape marks.

Subsequent skip record (or skip file) requests from this tape position will terminate immediately, producing no net tape movement and returning the SS$_ENDOFVOLUME status with a skip count of zero.

To move the tape beyond the second tape mark, you must employ another I/O function. For example, the IO$_READLBLK function, if issued after receipt of the SS$_ENDOFVOLUME status return, will terminate with an SS$_ENDOFFILE status and with the tape positioned just beyond the second tape mark.

From this new position, other skip functions could be issued to produce forward tape motion (presuming that there is additional data on the tape).

If three consecutive tape marks are encountered during a forward skip function, you must issue two IO$_READLBLK functions; the first to get the SS$_ENDOFFILE return, the second to position the tape past the third tape mark.

2.5.3 **User Stack Now Automatically Expanded** - VAX/VMS now expands the user stack automatically, obviating the need for exactly specifying user stack size to the Linker and for user code to extend the stack. Note that there is no option to disable this automatic stack expansion.

2.6 Run-Time Library

This section contains notes on changes and additions to the VAX-11 Common Run-Time Procedure Library. The documentation that describes the Run-Time Library is:

- VAX-11 Run-Time Library Reference Manual
- VAX-11 Guide to Creating Modular Library Procedures

2.6.1 **Compatibility Between Different Versions of VAX/VMS** - When you link with a run-time library, compatibility between different versions of the operating system is only guaranteed from an earlier version to a later version (upward compatibility) and not from a later version to an earlier one (downward compatibility). For example, a program linked on Version 1.6 will run on Version 2.0, but not vice versa.

Note that any program that requires CMKRNL or CMEXEC privileges must be relinked to run on VAX/VMS V2.0. The reason is that programs not linked with the current system have their CMKRNL and CMEXEC privileges turned off at image activation time to prevent them from attempting to access symbols whose values have been changed in SYS.STB.

In Version 1.6 systems, the information to determine whether an image linked against SYS.STB is not available. Therefore, CMKRNL and CMEXEC privileges for all Version 1.6 images are turned off at image activation time and restored at image run-down.

2.6.2 **Routine for Determining Setting of Lines-per-Page** - Many compilers and utilities produce output files formatted for the pages of a line printer. In V2.0, the Run-Time Library contains a routine, LIB$LP_LINES, that native mode programs can call to determine the user setting of the number of lines on a printer page. The routine is documented in the VAX-11 Run-Time Library Reference Manual.

Refer to Section 2.11.11 of these release notes for related information.

NOTE

Compatibility mode programs cannot use the LIB$LP_LINES routine. For example, the page size for MACRO-11 listings cannot be varied conveniently.

2.6.3 **FORTRAN Default Logical Unit Assignments** - The implementation of VAX-11 FORTRAN default logical unit assignments has been changed to reduce the number of language-specific logical names in the VAX/VMS system logical name table. VAX-11 FORTRAN, for example, provides the following correspondence between FORTRAN I/O statements, FORTRAN-specific logical names, and system logical names:

Statement	FORTRAN Logical Name	System Logical Name
ACCEPT	FOR$ACCEPT	SYS$INPUT
TYPE	FOR$TYPE	SYS$OUTPUT
READ	FOR$READ	SYS$INPUT
PRINT	FOR$PRINT	SYS$OUTPUT
any on unit 5	FOR005	SYS$INPUT
any on unit 6	FOR006	SYS$OUTPUT

In earlier versions of VAX/VMS, default logical unit assignments were implemented as system-wide logical names. These names were defined by ASSIGN/SYSTEM commands in the DIGITAL-supplied system start-up command procedure.

In VAX/VMS V2.0, these logical names are not defined as system-wide logical names. Instead, the Run-Time Library contains code that provides logical unit defaulting. When the first I/O operation is performed on any of these units, the FORTRAN-specific logical name is translated. If the logical name translation succeeds, the translated name is used (as in earlier versions). However, if the logical name is not defined (the default case), SYS$INPUT or SYS$OUTPUT is substituted, as appropriate. Therefore, to the FORTRAN program, the default logical unit assignments appear to be the same as they were in earlier versions of VAX/VMS.

FORTRAN programs that were linked to the Version 1.6 shared Run-Time Library (the default) will work without change. However, programs that were linked using the /NOSYSSHR linker command qualifier may fail when they run under VAX/VMS V2.0. The error message returned is:

```
%FOR-F-OPEFAI, open failure
```

The appropriate logical name is also displayed. If the program used units 5 or 6, no error occurs, but the system attempts to use the files FOR005.DAT or FOR006.DAT.

To prevent problems with programs linked /NOSYSSHR, do one of the following:

- Link the program again on V2.0
- Define the required logical names at the process level before running the program
- Have the system manager insert ASSIGN/SYSTEM commands for the required logical names in the site-specific system start-up command procedure [SYSMGR]SYSTARTUP.COM

2.7 VAX-11 Record Management Services

This section contains notes that describe changes and additions to the VAX-11 Record Management Services. Documentation on VAX-11 RMS is found in the following manuals:

- Introduction to VAX-11 Record Management Services
- VAX-11 Record Management Services User's Guide
- VAX-11 Record Management Services Reference Manual

2.7.1 **RMSSHARE Utility** - Changes have been made to the RMSSHARE utility; they are described in the VAX/VMS System Manager's Guide.

2.7.2 **RMSANLZ Unsupported Utility** - An unsupported utility program, RMSANLZ, has been added to the V2.0 kit. The primary use of RMSANLZ is to provide an interactive, formatted dump of information in indexed files. Refer to Appendix B of the VAX-11 Record Management Services User's Guide for information about RMSANLZ.

2.7.3 **Magnetic Tape Label HDR3** - A new label, HDR3, will appear on VAX/VMS ANSI-format magnetic tape. The label has been added to provide a place for some binary data that has been removed from HDR2.

Until the next version of VAX/VMS is released, VAX-11 RMS will produce tapes with both binary data in HDR2 and a HDR3 label. After the next version of VAX/VMS is released, VAX-11 RMS will not place binary data in HDR2, and will rely on HDR3 to contain VAX-11 RMS attribute information.

Note that the magnetic tape ACPs in V2.0 (and the next VAX/VMS version) will be able to read tapes produced prior to V2.0.

2.7.4 **Sequential FIle Sharing** - VAX-11 RMS now supports file sharing for sequential files with fixed-length, 512 byte records.

2.8 Record Management Services for the PDP-11 (RMS-11)

There are no changes or additions to RMS-11 V1.8, which was released as a component of VAX/VMS V1.6. Documentation for RMS-11 is found in the following manuals:

- RMS-11 User's Guide
- RMS-11 MACRO Reference Manual

2.9 Compatibility Mode

This section contains notes on changes and additions to functions that affect compatibility-mode programming on a VAX/VMS V2.0 operating system. Documentation describing compatibility-mode programming is found in:

- VAX-11/RSX-11M User's Guide
- VAX-11/RSX-11M Programmer's Reference Manual

2.9.1 **Change in P1 Parameter of IO.ATT Directives** - The meaning of the P1 parameter on an IO.ATT QIO RSX-11M directive has changed in V2.0. This change now makes the AME compatible with RSX-11M Version 3.2 but incompatible with previous versions of VAX/VMS.

2.9.2 **AME Emulation of QIO$ Directives for Unknown Devices** - The AME emulation of RSX-11M QIO parameter mapping for unknown device types has changed.

In V2.0, the correspondence between RSX-11M QIO parameters P1 to P6 is equivalent to the VAX/VMS QIO parameters P1 to P6.

NOTE

Unknown device types are defined as any device that is not a disk, terminal, line printer, card reader, or magnetic tape. A null device is an unknown device.

2.9.3 **MACRO-11 CRF Now Supported** - The RSX-11M V3.2 program CRF.EXE is now supported. Add the following DCL command to [SYSMGR]SYSTARTUP.COM; then, CRF can be invoked by using the /CR qualifier in the MACRO-11 command string.

```
$ DEFINE/SYSTEM CRF SYS$SYSTEM:CRF
```

2.9.4 **Changes to MCR Command Language Interpreter** - The MCR Command Language Interpreter has been updated to include features found in Version 3.2 of RSX-11M.

2.9.5 **LIBRARY/RSX11 Library Extractions** - When using the LIBRARY/RSX11 command to extract compatibility-mode object modules from the macro library, you should specify the library type by including a file type in the library file specification. For example:

```
LIBRARY/RSX11/OUTPUT=TT:/EXTRACT=macro filename.MLB
```

Do not use the /OBJECT or /MACRO qualifiers, as in the following incorrect example:

```
LIBRARY/RSX11/OUTPUT=TT:/EXTRACT=macro/MACRO filename
```

2.9.6 **Compatibility Mode Utilities Now Native Mode** - In previous versions of VAX/VMS, the following commands invoked the compatibility-mode image PIP. In V2.0, these commands invoke native-mode images:

RENAME

UNLOCK

CREATE

CREATE/DIRECTORY

SET PROTECTION

DIRECTORY

2.10 System Programming

This section contains notes that describe changes and additions to the system components used by VAX/VMS system programmers. The documentation for system programming under VAX/VMS V2.0 is found in the following manuals:

- VAX/VMS Real-Time User's Guide
- VAX/VMS Guide to Writing a Device Driver
- VAX/VMS System Dump Analyzer Reference Manual
- VAX-11 PATCH Utility Reference Manual

2.10.1 **All Drivers Must Be Reassembled and Relinked** - Current users must reassemble and relink all user-written drivers after VAX/VMS V2.0 has been installed. Because the executive has been reassembled and relinked, drivers also must be reassembled and relinked to ensure that they use the correct data structure definitions and link with the correct addresses of executive routines.

Note that drivers must be reassembled and relinked with every version upgrade but not with version updates, because version updates include a patched executive, not a reassembled and relinked executive.

2.10.2 **Buffered I/O Count Quota Field Changes** - In earlier versions of VAX/VMS, the field that contained the process byte-count quota, PCB$W_BYTCNT, was in the software Process Control Block (PCB).

For VAX/VMS V2.0, that field has been moved to a new structure, the Job Information Block (JIB) and has been given a longword value. The new field is labeled JIB$L_BYTCNT.

Any driver using the byte-count quota field will require some coding changes. For example, the code that adjusts a process's quota could have been:

```
SUBW     R1,PCB$W_BYTCNT(R4)     ; Adjust byte count quota
```

For V2.0 that code would have to be changed to:

```
MOVL    PCB$L_JIB(R4),R0      ; Get JIB address
SUBL    R1,JIB$L_BYTCNT(R0)   ; Adjust byte count quota
```

NOTE

It may be necessary to convert the addend in R1 from word to longword.

When the changes have been made, reassemble and relink the driver. Typically, the only drivers that need make this change are drivers that perform buffered I/O.

2.10.3 **Interrupt Dispatch Table Can Be Set for More Than 8 Units** - In earlier versions of VAX/VMS, the Interrupt Dispatch Block (IDB) contained a field, IDB$B_UNITS, that contained the value 8, effectively limiting the number of units attached to one controller to eight.

In VAX/VMS V2.0, that field has been changed to a word value, IDB$W_UNITS. Now, you can define the number of units allowed in the IDB by including the following qualifier in the Driver Prologue Table creation macro:

```
DPTAB
   MAXUNITS=n,...
```

Then, reassemble and relink the driver. If the MAXUNITS=n value is not specified, it defaults to the Version 1.6 value of 8.

After the number of units is so defined, you can override the value by using the following qualifier in the SYSGEN utility CONNECT command:

```
/MAXUNITS=n
```

The qualifier is optional: use it only when you want to change the maximum number of units from that specified in the DPT creation macro.

2.10.4 **New Macro for Purging a UNIBUS Datapath** - In earlier versions of VAX/VMS, code used to purge a UNIBUS adapter (UBA) datapath had to be included in each driver that used a buffered datapath.

In V2.0, a macro call can be used to purge the datapath. For example, the RK06/RK07 driver code used to purge the UNIBUS datapath was similar to the following:

```
MOVL    UCB$L_CRB(R5),R3                              ; Get CRB address
MOVL    @CRB$L_INTD+VEC$L_ADP(R3),R2                  ; Get config. reg. addr.
EXTZV   #VEC$V_DATAPATH,#VEC$S_DATAPATH,-             ; Extract datapath
        CRB$L_INTD+VEC$B_DATAPATH(R3),R1
ASHL    #UBA$V_DPR_BNE,#1,UBA$L_DPR(R2)[R1]           ; Purge datapath
ASHL    #31-UBA$V_DPR_XMTER,UBA$L_DPR(R2)[R1],R0      ; Transfer error?
BGEQ    20$                                           ; if GEQ none
```

For V2.0, the code that performs the same function consists of a macro call and an error check:

```
PURDPR              ; Purge datapath, check and clear errors
BLBS      R0,20$    ; Branch if no datapath error
```

The PURDPR macro has been included to allow drivers to call a new executive routine, IOC$PURGDATAP, instead of having to contain the code themselves.

2.10.5 **Code Change Required in Some V1.6 Drivers** - Any V1.6 drivers that use the status bits IRP$V_PRIVIO or IRP$V_EXQUOTA in IRP$W_STS to check process privileges must be changed.

A new IRP field, IRP$L_ARB, has been defined that points to the process Access Rights Block (ARB) in the Process Control Block. The ARB contains both the process privilege mask and its UIC. Documentation for the ARB is contained in the VAX/VMS Guide to Writing a Device Driver.

2.10.6 **PATCH Utility Turns Off "Last P0 Cluster" Option** - The PATCH utility can add patch area to an image file by finding the last virtual address used by the image and then creating an image section that uses the next available virtual address space. Both executable and shareable image files can have patch area added by this method.

When you link an executable image that contains one or more shareable images, the linker gives the "Last P0 Cluster" attribute to all the image sections from the shareable image that is positioned last in the virtual address space. This attribute tells the image activator that there are no image sections with higher virtual address ranges, thus allowing the image sections to grow without needing to relink the executable image. The image activator will automatically map any new space needed when the executable image is run.

The earlier versions of the PATCH utility did not turn off the "Last P0 Cluster" option when the patch resulted in new patch area. Thus, a new version of a shareable image having this option and growing in size (using more virtual address space) would result in the image activator first mapping the new space and then mapping the patch area over the new space.

The V2.0 PATCH utility turns off the "Last P0 Cluster" option for the last shareable image linked into an executable image when patch area is created for that executable image. The image activator checks that the "Last P0 Cluster" option is off when it activates an executable image in which the shareable image has grown. By finding where the patch area begins, it can calculate that the shareable image is too large to fit into the virtual address space allocated to it, and returns the error message:

```
SHARTOOBIG, Shareable image too big, please relink all images
```

The executable image should be relinked against the new copy of the shareable image (and then repatched if necessary). When two or more shareable images are linked together into a new shareable image and an executable image is linked to the combined image, the combined shareable image and the executable image both need to be relinked if the SHARTOOBIG message is returned.

2.11 System Management and Operation

This section contains notes that describe changes and additions to system features that affect VAX/VMS system management and operations. The documentation for these features is found in:

- VAX-11 Software Installation Guide
- VAX/VMS System Manager's Guide
- VAX/VMS Operator's Guide
- VAX/VMS UETP User's Guide

2.11.1 **System Security: SET PASSWORD Command** - One of the ways in which the security of a system can be improved is by allowing users to change their passwords. The SET PASSWORD command has been added to DCL for this purpose. To help maintain the secrecy of passwords, this command does not echo user input. To help ensure that a mistake is not made, users must enter their password twice. In the following example, the text enclosed in angle brackets represents user input that is not echoed.

```
$  SET PASSWORD
Old password:  <old password>
New password:  <new password>
Verification:  <new password>
```

Users can change their password at any time. However, the system manager can prevent users from changing their passwords by setting the login flag LOCKPWD in the user's record of the user authorization file. Refer to the VAX/VMS System Manager's Guide for the procedures used.

NOTE

All passwords will be encrypted using a new, more secure algorithm when set or changed using either AUTHORIZE or SET PASSWORD.

2.11.2 **System Security: Terminal Allocation** - The ability of a nonprivileged program to allocate an interactive terminal that is not currently in use can cause a potential security problem.

One way to prevent the problem is to protect the allocation of non shareable, non-filestructured devices. The SET PROTECTION command format SET PROTECTION=code/DEVICE provides the capability to set the owner UIC and protection for a nonshareable device. The READ protection flag is used to allow or deny the desired allocation.

Two new SYSGEN parameters have been added to set the protection and owner UIC of all terminals on a system: TTY_PROT and TTY_OWNER.

Currently, the protection does not restrict who can log in at a terminal (due to the way the Job Controller and the Login procedure interact). Rather, the protection mask should be set to allow no access to any class. Since the Login procedure is started with the BYPASS privilege all users will be able to log in, but none will be able to randomly allocate the terminal.

2.11.3 **System Security: Privileged Images and CTRL/Y** - In earlier versions of VAX/VMS, a user at a terminal could disrupt a privileged image by pressing <CTRL/Y> then entering commands, such as EXAMINE and DEPOSIT, that would affect the image before the image's exit handling routines could take effect.

Now, images that have been installed privileged perform an unconditional rundown (with a call to the image's exit handlers) when a CTRL/Y interrupt is detected. The effect is to prevent casual interruption of a privileged image's exit handling routines when <CTRL/Y> is pressed.

Note that the image can be installed with any privilege (not necessarily a "meaningful" one) to prevent these CTRL/Y interrupts.

2.11.4 **User Authorization File Has New Format** - The format of the user authorization file (UAF) has been changed in V2.0. The UAF file on the system disk (SYSUAF.DAT) is automatically converted to the new format during the upgrade process.

If you have created other UAFs, you must convert them, using the [SYSUPD]CVTUAF command procedure, since all the VAX/VMS programs that access UAF files (LOGIN, AUTHORIZE, etc.) require the new format. To make the conversion, type:

```
$  SET DEFAULT SYS$SYSTEM
$  @[SYSUPD]CVTUAF
```

The command procedure will prompt you for the device on which the old SYSUAF.DAT resides. You must have read access to the old SYSUAF.DAT and the ability to create files on SYS$SYSTEM. The scratch file SYSUAF.TMP is used to sort the old SYSUAF.DAT thereby compacting the initial population of the new SYSUAF.DAT.

The AUTHORIZE utility has two commands that allow the system manager to display of a particular user's record in the UAF. The LIST command is used to create a file, SYSUAF.LIS, which then can be printed; the SHOW command is used to obtain a display at the terminal. Both commands have been enhanced; refer to the VAX/VMS System Manager's Guide for details.

2.11.5 **New Device Name for Console Floppy** - The console floppy diskette name has been changed from DXA1 to CSA1.

The driver is now loadable. To load the driver use the commands:

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

These commands require the CMEXEC and CMKRNL privileges. Note that the console is not autoconfigured.

If you often configure the console floppy, you should protect it from unauthorized access by mounting it with the /FOREIGN and /SYSTEM qualifiers.

2.11.6 **Importance of Using the V2.0 VMB** - The VAX/VMS primary bootstrap file, VMB.EXE, is in [SYSEXE] on the V2.0 system binary distribution and on the console floppy diskette (label RX 1/11780 LOCAL CNSL PKG). When installing V2.0, use the V2.0 console floppy diskette.

However, customers who have purchased a field service maintenance contract that includes the remote diagnosis service receive another console floppy diskette (AS-E989n-YE), which is to be used as the console floppy diskette after the installation is completed.

If you have the remote diagnosis console floppy diskette, you may need to copy the [SYSEXE]VMB.EXE file from the system, and the files DSCGEN, CRASH, and DSCBOO.CMD from the V2.0 console floppy diskette to the remote diagnosis console floppy diskette.

After V2.0 has been installed, use the V2.0 command procedure [SYSUPD]DXCOPY to copy the ASCII files (DSCGEN, CRASH, and DSCBOO.CMD) to the remote diagnosis floppy, and the V2.0 command procedure [SYSUPD]BOOTUPD.COM to copy [SYSEXE]VMB.EXE from the V2.0 system disk to the remote diagnosis floppy.

NOTE

V2.0 can be bootstrapped using the V1.6 VMB, but system crash dumps will not contain the contents of shared memory and the V2.0 stand-alone DSC will not boot.

2.11.7 **VMS-to-VMS Network Command Terminal Capability** - VMS-to-VMS Network Command Terminal capability is now available. The requirements are:

- DECnet-VAX V1.3 is installed on each node
- REMDEVICE.DAT in SYS$SYSTEM
- RTTLOAD.COM in SYS$SYSDISK:[SYSMGR]

To bring up network command terminals, execute this command:

```
@SYS$SYSDISK:[SYSMGR]RTTLOAD
```

This command procedure loads the RTT driver for network command terminals and runs the image SYS$SYSTEM:REMACP.EXE as a detached process. REMACP will not come up if:

- DECnet-VAX is not up.
- There is no REMDEVICE.DAT.
- RTT driver has not been loaded.

To remove REMACP, type:

```
RUN SYS$SYSTEM:STOPREM
```

REMACP will delete itself when all network command terminal activity terminates.

If the network is brought down, REMACP will automatically delete itself, as though STOPREM had been run. After the network is brought up, you must execute the SYS$SYSDISK:[SYSMGR]RTTLOAD command procedure to bring up network command terminals again.

If you press <CTRL/Y> repeatedly, you will be asked if you wish to terminate the network command terminal session; otherwise, terminate the session by logging out.

If you intend to use the SET HOST command, the image RTPAD must be installed. The commands to install RTPAD are:

```
$ RUN SYS$SYSTEM:INSTALL
  RTPAD /OPEN/SHARED/PRIV=(NETMBX,TMPMBX,LOG_IO)
```

Note that the maximum number of inbound remote logins possible on a single DECnet-VAX node is 15.

2.11.8 **Using the New Stand-alone DSC2** - The operating procedures for the new version of stand-alone DSC2 are described in Section 2.2 of the VAX-11 Software Installation Guide. More information on the DSC utilities is contained in the VAX-11 Utilities Reference Manual, in Chapter 4.

2.11.9 **New Feature for ERRFMT Process** - The ERRFMT process now deletes itself if it encounters too many errors while writing the error log file. It also reports more specific error messages and prints a warning message before deleting itself.

In earlier versions of VAX/VMS, the ERRFMT process would attempt to write to the error log file in spite of error conditions and would continually broadcast error messages to the operator console. The operator had to intervene to stop the process.

In addition, the new command procedure ERFSTART.COM allows the operator to restart the ERRFMT process after the problem that caused the deletion is fixed. To restart, log in to the system manager's account and type:

```
@[SYSMGR]ERFSTART
```

The ERRFMT process will be restarted.

2.11.10 **New CRASH Command Procedure** - V2.0 contains a new command procedure on the system console medium. CRASH enables you to halt the system in the event that the system hangs and does not respond to an orderly system shutdown.

To execute this command procedure, perform the following:

1. Press <CTRL/P> on the system console.

   The system responds by displaying the >>> prompt.

2. At the system console, type: HALT<RET>

   The system responds by displaying the >>> prompt.

3. At the system console, type: @CRASH

   The system then sends the shutdown information to the console. Save this console listing and send it (and the corresponding dump file) with a Software Performance Report (SPR) to DIGITAL for examination.

2.11.11 **New Method to Change Default Lines-per-Page Value** - Many native mode programs, such as VAX-11 MACRO, VAX-11 BASIC, the linker, and VAX-11 FORTRAN, paginate listing files. The default value for physical lines-per-page in VAX/VMS is 66. In earlier versions of VAX/VMS, installations that wanted to change this default value had to use the command procedure [SYSUPD]LINEPAGE.COM to patch in a new default value.

The LINEPAGE.COM command procedure has been removed from the V2.0 kit, and a new system-wide logical name has been defined: SYS$LP_LINES. Assign a value to this logical name to produce a physical lines-per-page value for native mode utilities to use in routines that calculate the number of lines to be printed on output listings. Refer to the VAX/VMS Run-Time Library Reference Manual for more information.

2.11.12 **Effects of File Manager Performance Enhancements** - The Files-11 file manager in VAX/VMS V2.0 contains several optional enhancements that more than double its performance. However, their presence requires that greater care be taken in mounting and dismounting disks and in shutting down the system. The optional performance enhancements are described in the VAX/VMS System Manager's Guide.

CAUTION

Because the optional caching features cause a mounted disk to appear inconsistent, running VFY2 on a disk that has been in use may result in a variety of error messages. To get a consistent report from VFY2, you must flush the caches by dismounting and remounting the disk (and making sure that no file activity occurs while VFY2 is running).

2.11.13 **New AUTOCONFIGURE Device Support** - The table of UNIBUS devices recognized by AUTOCONFIGURE has been extended to include new peripheral devices manufactured by DIGITAL. Also, the CSR and vector assignments for the DR11B have been revised. Refer to Chapter 14 of the VAX/VMS Guide to Writing a Device Driver for a list of current CSR and vector assignments.

Note the new devices expected by AUTOCONFIGURE in floating CSR and vector space may interfere with customer device assignments in floating space.

2.11.14 **AUTOCONFIGURE Lineprinter Errors Corrected** - In earlier versions of VAX/VMS, the AUTOCONFIGURE command assumed incorrect CSR addresses for all but the first line printer in a configuration. In V2.0, the SYSGEN AUTOCONFIGURE command has been corrected to look for a second line printer CSR at 774004 (instead of 764004). Also, three line printer CSRs have been added.

The following table lists the CSRs and vectors for the five line printers:

Line Printer	CSR	Vector
1	%O777514	%O200
2	%O764004	%O170
3	%O764014	%O174
4	%O764024	%O270
5	%O764034	%O274

2.11.15 **Floppy Disks as Output from RSX-11S System Generation** - If you perform RSX-11S system generations on your V2.0 system, either the console floppy device or an RX02 floppy device can be (among other media) the device to which the RSX-11S system image is written.

If the console floppy device is used, an RX01 bootstrap is included in the RSX-11S system image. If an RX02 device is used, an RX02 bootstrap is included in the RSX-11S system image and the RX02 is written in single density format.

NOTE

It is not prudent to use the console floppy device for these operations. DIGITAL recommends use of other output devices (such as RX02s) if possible.

Note that the bootstraps are different: an RSX-11S system image that was written to the console floppy device cannot be bootstrapped from an RX02 device; an RSX-11S system image that was written to an RX02 device cannot be bootstrapped from an RX01 device.

2.11.16 **Use of Directory [10,40]** - Although use of the directory [10,40] has been allowed in earlier versions of VAX/VMS, it should not be used after V2.0 is installed. Instead, use the [SYSEXE] directory.

The reason for this caution is that support for the use of directory [10,40] will be terminated in a later VAX/VMS release.

2.11.17 **SYS Parameter BUGCHECKFATAL** - The SYS parameter BUGCHECKFATAL enables or disables the making of all bugchecks fatal bugchecks. In earlier versions of VAX/VMS, the BUGCHECKFATAL parameter default was set on, thereby making all bugchecks fatal.

In V2.0 the default setting for the BUGCHECKFATAL parameter (and the DIGITAL-supplied setting) is off. Thus, only a fatal bugcheck will require that the system be rebootstrapped; a nonfatal bugcheck just places an entry into the error log.

2.12 Unsupported Software

This section contains notes pertaining to unsupported software.

2.12.1 **DIGITAL Standard RUNOFF (DSR) Replaces RNO.EXE** - In earlier versions of VAX/VMS, the unsupported text formatting utility program RNO.EXE was included in the [SYSEXE] directory.

In V2.0, RNO.EXE has been replaced by the program RUNOFF.EXE, an unsupported VAX/VMS version of DIGITAL Standard Runoff (DSR).

NOTE

Although a few known incompatibilities exist, DSR is mostly compatible with RNO.EXE. Please do not submit problem reports.

Three DSR utility files are included in [SYSEXE]:

TOC.EXE	Unsupported RUNOFF Table of Contents utility
TCX.EXE	Unsupported RUNOFF Two-column Index utility
NDXDMP.EXE	Unsupported RUNOFF Index Dump utility

To use DSR, type the DCL command:

```
$  RUNOFF /qualifiers... filespec,...
```

To use the utilities provided with DSR, type the DCL RUN command:

```
$  RUN SYS$SYSTEM:TOC
$  RUN SYS$SYSTEM:TCX
$  RUN SYS$SYSTEM:NDXDMP
```

To obtain information about using DSR and its utilities order the manual:

DEC Standard Runoff (DSR) User's Guide (Order No. AA-J268A-TK)

Note that this manual is not part of the VAX/VMS V2.0 documentation kit, and must be separately ordered.

Currently, there are some known problems with the RUNOFF.EXE program:

1. Restriction on using the /SIMULATE and /PAUSE qualifiers.

   If you use these qualifiers, a prompt is not returned and nothing seems to happen. Note, however, that DSR is waiting for a response. The work around for this bug is to type a space, then press <RET>. If you just press <RET>, RUNOFF will hang.

2. You cannot specify a character for underlining if you have specified /UNDERLINE:SEPARATE or /UNDERLINE:NOSPACING. There is no way to work around this problem.

An undocumented feature of DSR is the ability to control the number of blank lines generated by an .END LIST or .END NOTE command.

To specify the number of blank lines to be generated, add a number to the .END LIST or .END NOTE command. For example:

```
.END LIST 0        no blank lines are generated
.END NOTE 9        nine blank lines are generated
```

You can specify any value that would be acceptable to the .SKIP command.

The NDXDMP utility accepts an input file produced by the /INDEX qualifier in DSR and creates an output file with the type XMP. This file contains a dump of your index file similar to that produced by the /DEBUG:INDEX qualifier in DSR, but in an easier-to-read format.

2.12.2 **Page Fault Monitor** - The Page Fault Monitor is an unsupported facility that can be used to generate page fault information about a program. The following help file contains information about the use of the facilty:

```
SYS$HELP:PFMON.HLP
```

The procedures you use to insert HLP files into SYS$HELP:HELPLIB.HLB are found in the LIBRARIAN chapter of the VAX-11 Utilities Reference Manual. PFMOM requires the SETPRV, ALTPRI, and PSWAPM privileges.

When page fault monitoring is enabled by an image, a subprocess, PFMFILWRT, is created. If the image exits without disabling page fault monitoring, the PFMFILWRT subprocess does not exit.

NOTE

On a heavily-loaded system, you can lose monitoring samples if the system takes a significant amount of time to create the PFMFILWRT subprocess. This is because the $SETPFM system service does not wait until the subprocess is created before returning to the calling image.

2.12.3 **Unsupported Utilities** - Included in the V2.0 kit are several unsupported utilities. The utilities that perform video terminal screen formatting use the screen package in the VAX-11 Run-Time Library. You can install the common utilities, such as TALK and USERS, with the required privileges so all users have access to them.

2.12.3.1 **INFO** - The INFO.EXE program displays constantly updated information about a process. When INFO is run as a foreign command, the first parameter specifies the name of the process to be displayed.

While INFO is running, you can press the V key to display a map of the pages in the virtual address space of the process. Valid pages in the working set are represented by an asterisk (*); the current PC is represented by an at sign (@). You can press the space bar to return to the first INFO display. INFO requires CMEXEC privilege.

2.12.3.2 **SEARCH** - The SEARCH.EXE program searches source files for a given string. When SEARCH is run as a foreign command, the first parameter specifies the name of the file(s) to search (wildcards are allowed) and the second parameter specifies the string to search for. If the parameters are not specified, SEARCH prompts for them.

2.12.3.3 **SETNAME** - The SETNAME.EXE program sets the current process name. When SETNAME is run as a foreign command, the first parameter specifies the new process name.

2.12.3.4 **SFA** - The SFA.EXE program is a formatting file dump utility. There is a help file, SYS$HELP:SFA.HLP, that can be added to SYS$HELP:HELPLIB.HLB, and that describes all the functions of SFA.

2.12.3.5 **TALK** - The TALK.EXE program allows users on two different video terminals to type messages interactively to each other. TALK requires OPER, WORLD, and PRMMBX privileges.

2.12.3.6 **USERS** - The USERS.EXE program displays a list of all the interactive users currently logged into the system. When USERS is run as a foreign command, the first parameter specifies the user names to display; if unspecified, all users are displayed. USERS requires WORLD privilege.

2.12.3.7 **WHO** - The WHO.EXE program converts a UIC to a user name or a user name to a UIC. WHO has a help command for further information. WHO requires the SYSPRV privilege.

2.12.4 **New TECO-11 Manual** - The PDP-11 TECO User's Manual has been revised; its new order number is DEC-11-UTECA-B-D. The manual is included in the kit.

3.0 PROBLEMS RESOLVED BY VAX/VMS V2.0

All restrictions and problems noted in the release notes for VAX/VMS V1.5 and V1.6 have been resolved, with the exceptions mentioned in Section 4.0 of these release notes.

4.0 CURRENT RESTRICTIONS AND PROBLEMS

This section lists known restrictions and problems.

4.1 Command Languages and System Messages

This section contains restrictions and problems known to exist in V2.0.

4.1.1 **DIRECTORY Command /EXCLUDE Qualifier** - The /EXCLUDE qualifier in the DIRECTORY command does not work with zero version numbers.

A version number of zero should exclude the latest version of the file. Instead, specifying a version number of zero excludes all versions of the file.

4.1.2 **Restriction on Use of Wild Card Characters** - Wild card characters are not allowed in the version field of a file specification. The one exception allowed is a single character * (asterisk). If you enter an invalid version using the asterisk character (for example ;5*7), an error is not returned and unpredictable results may occur.

4.1.3 **Effect of Wild card Operations on Devices Mounted Foreign** - Any wild card operation on a device that has been mounted foreign causes an executive mode exception.

If the SYS parameter BUGCHECKFATAL is set off, the process is deleted; if BUGCHECKFATAL is set on, a fatal bugcheck occurs.

Refer to Section 2.11.17 of these release notes for related information.

4.1.4 **Incorrect HELP Text for MOUNT Command** - The HELP text for the MOUNT command incorrectly describes the command's syntax. The text shows the device name and volume label lists enclosed in parentheses. The presence of parentheses with these parameters is incorrect. The VAX/VMS Command Language User's Guide correctly describes the command's syntax.

4.2 Program Development Tools

This section contains restrictions and problems known to exist in V2.0.

4.2.1 **LIBRARIAN Restrictions and Problems**

4.2.1.1 **Using /NAMES and /LIST Qualifiers** - Always use the /NAMES and /LIST qualifiers together. If you specify the /NAMES qualifier without also specifying the /LIST qualifier, the LIBRARIAN will terminate abnormally.

4.2.1.2 **Using /OUTPUT Qualifier** - If you issue a command to modify a VAX/VMS V1.6 (or earlier) library, always specify the /OUTPUT qualifier.

If you do not specify /OUTPUT, the LIBRARIAN will write the new V2.0 format library into the current default directory.

Use of the /OUTPUT qualifier ensures that the new library will be written to the device and directory of the library you are modifying.

4.2.1.3 **LIBRARIAN Failures on Update and Insert** - The LIBRARIAN will not insert a zero-length object, help, or macro file into a library, and will give no indication of the failure.

One situation in which the failure can occur is when an object file with zero length has been assembled (for example, as a result of issuing CTRL/Y to the assembler). Attempts to insert or replace this object module in an OLB library will appear to be successful, while in fact, the LIBRARIAN will fail to do either.

4.2.2 **<CTRL/Y> May Corrupt Library** - If you issue a CTRL/Y while modifying a library, the library may be corrupted. Later attempts to use the corrupted library will result in the message:

```
%LIBRAR-E-WRITEERR, error writing <filespec>
-LBR-E-RFAPASTEOF, VBN in map block request past eof. Submit SPR
```

4.2.3 **Broadcast Message Interrupts SOS Alter Mode Screen** - When a broadcast message interrupts the altering of a line in SOS Alter Mode, the screen is not restored properly (to look as it did before the broadcast). The cause is the character-at-a-time nature of the terminal I/O requests done by SOS in Alter Mode.

To restore the screen to its correct state, you must enter CTRL/R. Then, the line is redisplayed up to its current position.

4.2.4 **Linker Restrictions and Problems**

4.2.4.1 **LINKER Effect on Address Space of Executable Image** - If you are creating a shareable image that contains relocatable address data (using the VAX-11 MACRO .ADDRESS directive or its equivalent), the linker keeps the shareable image position-independent by a mechanism called deferred relocation (see the VAX-11 Linker Reference Manual for details).

To guarantee that the relocation is valid, the linker forces the shareable image to have the default global section match (the match equal to the middle 32 bits of the shareable image creation time). This match forces relinking of all executable images linked with the shareable image every time the shareable image is relinked, thereby enforcing the rule that all symbol references within a shareable image must be resolved when that shareable image is created.

4.2.4.2 **Linking Multiple Shareable Images** - If you are creating independent shareable images that may be bound together in an executable image, link these shareable images using the LINK command qualifier /NOSYSSHR. By doing so, you will prevent the problem of multiple copies of the shared Run-Time Library appearing in the address space of the executable image.

4.2.4.3 **Shareable Images Linked with VMSRTL** - Shareable images linked with VMSRTL, the default run-time shareable image library, must be relinked for V2.0, along with any executable images linked against these shareable images.

In previous versions of VAX/VMS, the Linker did not place position-independent shareable images last in the virtual address space. The V2.0 linker does this correctly.

4.2.5 VAX-11 Symbolic Debugger

The following restrictions exist in V2.0.

4.2.5.1 **Deposits of H_floating Data** - Deposits of H_floating data are restricted to constants. For example, DEPOSIT H = 1.405 will work as expected, while DEPOSIT H = HVARIABLE will not work.

4.2.5.2 **Long Literals** - The debugger will not accept literals longer than 32 bits. For example, the instruction:

```
EXAMINE/INSTRUCTION LABEL
```

results in truncation to the low-order 32 bits and an appropriate diagnostic message if the label was defined as:

```
LABEL:     MOVQ    XFFFFFFFFFFFFFFF2,A
```

4.2.5.3 **String Variables** - String variables cannot be passed in a debugger CALL command.

4.2.5.4 **Short Floating-Point Literals** - Short floating-point literals are not correctly decoded by the debugger. For example, MOVF #4.0,R0 will decode as MOVF #18,R0.

4.2.5.5 **Real Variable Names** - When you use the format

```
evaluate/addr "expression"
```

and "expression" contains a real (floating-point) variable name, an incorrect value is displayed. Note however that the following format works correctly:

```
evaluate/addr "real variable"
```

4.2.5.6 **Global Structures** - Global structures declared at the module or routine level are not always recognized as having structure attributes.

You can circumvent this problem by specifying the explicit pathname desired:

```
examine module\global_structure [n]
```

4.2.5.7 **Condition Handlers** - You may get unexpected behavior when you debug a program that uses a condition handler that intercepts faults (for example, SS$OPCDEC and SS$ACCVIO) and looks at the opcode whose execution resulted in the fault. In other words, if you set a breakpoint on such an opcode and then attempt to execute it, your program's condition handler will execute a BREAKPOINT opcode in its place.

Three DIGITAL-supplied condition handlers have this property: the Run-Time Library procedures LIB$EMULATE, LIB$FIXUP_FLT, and LIB$SIM_TRAP. Note that LIB$FIXUP_FLT and LIB$SIM_TRAP are implicitly enabled by any FORTRAN program that uses the PDP-11 FORTRAN compatibility-mode routines ERRSET or ERRTST.

The restriction does not prevent you from setting breakpoints on such instructions; it does prevent you from proceeding or stepping away from a breakpoint while the breakpoint is in effect.

For example, consider a program which has LIB$EMULATE enabled as a condition handler. This procedure emulates execution of G_floating, H_floating, and Octaword instructions by intercepting the fault (opcode reserved to DIGITAL) generated when these instructions are executed on VAX-11 processors which do not support them. You may set breakpoints at such instructions, and you may stop at the breakpoints, but should you proceed or step away from the breakpoint, the handler will not execute correctly.

To work around this restriction, do either of the following:

- When you reach a breakpoint on such an instruction, use the CANCEL BREAK command to cancel the breakpoint; STEP past the instruction; and use SET BREAK to reenable the breakpoint.
- Use the command form SET BREAK/AFTER:0 to set breakpoints. This form clears the breakpoint once it is reached. You must reenable the breakpoint after stepping past the instruction.

4.2.6 **VAX-11 MACRO .IF Arguments** - The arguments of a VAX-11 MACRO .IF IDENTICAL or .IF DIFFERENT directive do not function properly when they are enclosed in:

- Argument delimiters (within angle brackets)
- User-defined delimiters

For example, the directive .IF IDENTICAL ABC,abc will be properly resolved as identical, while the directive .IF IDENTICAL <ABC>,<abc> will be considered as different.

4.3 System Services and I/O

There are no known restrictions or problems.

4.4 Run-Time Library

This section contains restrictions and problems known to exist in V2.0.

4.4.1 **DEBUG Restriction When Using Condition Handlers** - If you are using the Run-Time Library procedures LIB$EMULATE, LIB$FIXUP_FLT, or LIB$SIM_TRAP, there is a restriction on breakpoints in the Debugger.

Refer to Section 4.2.5.7 of these release notes for details.

4.4.2 **LIB$ADDX and LIB$SUBX Produce Incorrect Default Length** - The procedures LIB$ADDX and LIB$SUBX perform multi-longword addition and subtraction. Currently, they both incorrectly default the number of longwords to be processed if you omit the fourth parameter (length).

To prevent the problem, specify the number of longwords in the argument list. When the length parameter is specified, the procedures function correctly.

4.4.3 **Lowercase TPA$ Symbols Not Recognized** - When creating state tables for LIB$TPARSE, a VAX-11 MACRO problem can prevent TPA$ symbols from being recognized if they are entered in lowercase characters. For example, the symbol TPA$_EXIT is recognized, but the symbol tpa$_exit is not recognized.

To prevent the problem, always specify TPA$ symbols in uppercase characters.

4.5 VAX-11 Record Management Services

This section contains restrictions and problems known to exist in V2.0.

4.5.1 **Restriction on Copying Relative and Indexed Files** - By use of the COPY command, VAX-11 RMS relative and indexed files can be copied only to disk. To copy relative and indexed files to magnetic tape or to record-oriented devices, the RMS-11 CONVERT utility should be used. CONVERT is invoked by the command, MCR CNV.

This restriction also was present in V1.6.

4.5.2 **Device Characteristics Returned on Parse** - A successful call to the SYS$PARSE service returns with the device characteristics for the target device in the DEV field of the FAB. If the target device for the SYS$PARSE routine is a process-permanent file, however, no device characteristics are returned and the DEV field of the FAB remains unchanged.

This restriction also was present in V1.6.

4.5.3 **Incorrect BLISS-32 Definition in SYS$LIBRARY:STARLET.REQ** - In SYS$LIBRARY:STARLET.REQ, the VAX-11 BLISS-32 definition for the file protection XAB incorrectly defines these file protection access bits:

XAB$V_NOREAD

XAB$V_NOWRITE

XAB$V_NOEXE

XAB$V_NODEL

If you are coding VAX-11 BLISS-32 programs that use protection XABs, you can correct this problem by substituting the correct definitions in the REQ file before creating the L32 file (as described in the VAX-11 BLISS-32 documentation):

```
MACRO     XAB$V_NOREAD  = 0,0,1,0%;
MACRO     XAB$V_NOWRITE = 0,1,1,0%;
MACRO     XAB$V_NOEXE   = 0,2,1,0%;
MACRO     XAB$V_NODEL   = 0,3,1,0%;
```

These definitions are used to reference any of the system, owner, group, or world subfields as independent data structures.

Note that the associated mask values, XABM_NOREAD, XABM_NOWRITE, XAB$M_NOEXE, and XAB$M_NODEL, are correct.

4.5.4 **Current File Access Sharing Problem** - VAX-11 RMS currently has a problem that allows other processes independent access to files that VAX-11 RMS has opened for shared access.

When sharing is specified, VAX-11 RMS should restrict access to processes that either have been opened for file sharing using VAX-11 RMS and are performing record access operations, or are using Block I/O and have UPI (user provided interlocks) set and specify only read access for sharing.

Currently, there are two ways an independent process can improperly open a file that should be restricted to those processes performing shared access:

- A process using Block I/O and specifying UPI can open and write to a shared file.
- A compatibility-mode program can open a shared file and read or write it. This presents a problem, since there is no way for the compatibility-mode program to synchronize with VAX-11 RMS to correctly change the file.

4.5.5 **Incorrect Status Return when File Opened for Block I/O** - When you open a relative or ISAM file for Block I/O (the BIO FOP option is specified), area and key XABs are not filled in. The status returned should be OK_NOP. However, the status SUC is returned, instead. Both $OPEN and $DISPLAY, if the BIO FOP option is specified, will return the incorrect status.

The XABs are not filled in because the information required is in the file's prologue, which is not read when Block I/O is specified.

To prevent the problem, open the file for Block I/O and Record I/O (the BRO FOP option is specified). Opening a file in this manner allows the file's prologue to be read and all XAB information to be returned.

4.5.6 **Magnetic Tape Restrictions and Problems** - This section contains restrictions and problems known to exist in V2.0.

4.5.6.1 **MBA/TM03 ECO Prevents Magnetic Tape ACP Failures** - There are some magnetic tape ACP failures that have been caused by an MBA/TM03 hardware problem.

If a parity error occurs on a data transfer from magnetic tape, occasionally information will be written to the ACP address space outside of the data buffer. The result usually is invalid MTAACP queue addresses, an unexpected system service exception, and a system failure.

This problem can be prevented by having field service apply the following hardware ECOs:

M8276-TW004

M8277-TW007

70-13627-TW003

4.5.6.2 **Magnetic Tape ACP Error Can Cause Process to Hang** - A problem in the magnetic tape ACP can occur when a bad tape causes an error immediately after a file has been accessed on the tape.

The problem occurs when the ACP has cleared the I/O packet address before it can reference the address to return the error to the user process. The ACP then takes an exception and disappears, resulting in the user process waiting for the ACP to complete the request. Because the ACP has exited without deaccessing the file, both the process and the tape drive are hung.

If the problem occurs, you must reboot the system to eliminate the process and free the tape drive.

4.5.6.3 **Appending Records to an ANSI Magnetic Tape File** - There is only one way to append records to an existing ANSI magnetic tape file: you must open the file with put access, which positions the tape after the last record in the file.

You cannot use the NEF bit of the ROP if you want to append records to a file (without truncating) because the NEF bit causes the tape to be positioned at the beginning of the file. If you then use GETs to position the tape at the end of the file, certain EOF conditions are latched, preventing you from putting into the file at that point. Unfortunately, you will not receive an error until a number of PUTs have been executed (VAX-11 RMS blocks records to the tape, and the error is not detected until a complete block is written). As a result, put access is not allowed.

4.5.6.4 **EOF3 Label Not Written on ANSI Magnetic Tapes** - A HDR3 label has been added to ANSI magnetic tapes written by VAX/VMS. Although HDR labels and EOF labels are supposed to be symmetric, no EOF3 label has been added. This could be a problem for tape volumes that are created for interchange with foreign (non-DIGITAL) systems that do not tolerate nonsymmetric labeling.

To prevent such problems, use the MOUNT command's /NOHDR3 qualifier when mounting the volume. In this case, the HDR3 is not written and the labels within each file will be symmetric.

4.5.6.5 **Continuation Tapes Run Away if Not Initialized** - When creating multivolume tape sets using new magnetic tapes, always initialize the continuation tapes before creating the tape set. If you do not initialize the tapes, continuation tapes may run away.

The reason is that the magnetic tape ACP always validates protection information on magnetic tapes: if the tape has never been written on, it may run away.

4.5.6.6 **Foreign Tapes Run Away if Not Initialized** - Before mounting a new magnetic tape foreign, you should initialize the tape as an ANSI tape (as described in the VAX/VMS Command Language User's Guide, INITIALIZE command). If you do not first initialize the new tape, the MOUNT/FOREIGN command, which always attempts to read the tape, could cause a runaway condition.

4.5.6.7 **Support Level for ANS X3.27-1978** - VAX-11 RMS support for magnetic tape structures is based on, but not strictly equivalent to the ANS X3.27-1978 "Magnetic Tape Labels and File Structure for Information Interchange." Refer to the VAX-11 Record Management Services Reference Manual, Appendix B for a description of the format supported.

4.6 Record Management Services for the PDP-11 (RMS-11)

This section contains restrictions and problems known to exist in V2.0.

4.6.1 **Restrictions on Use of RMSIFL Utility** - The following two restrictions were known problems in VAX/VMS V1.6.

4.6.1.1 **Records Containing Duplicate Key Values May Lose Input Order** - RMS-11 routines maintain a first-in, first-out (FIFO) ordering of records containing duplicate key values. When RMSIFL sorts records, this order can change unpredictably, depending on how the records occur in the sort work files when the work files are merged.

4.6.1.2 **Large Bucket and Record Sizes Can Cause Command Line Termination** - RMSIFL does not limit bucket or record sizes for input or output files. However, unusually large bucket and record sizes may cause RMSIFL to terminate the command line (especially if RMSIFL must sort the input file).

4.6.2 **Restrictions to Use of RMSRST Utility** - The following restrictions were discovered after V1.6 was released.

4.6.2.1 **Version Number Failures with /SE Switch** - RMSRST interprets version numbers in the file specification as octal numbers. Thus, if you specify a number containing an 8 or a 9 in the file specification using the /SE switch, RMSRST terminates the command line with the error message:

```
/RST -- SYNTAX ERROR - NN
```

For example, the command line:

```
RST *.*=MT:FILE.SKP/SE:DATA.DAT;19
```

causes the following error message to be returned:

```
/RST -- SYNTAX ERROR - 19
```

One way to work around the restriction is to specify a wild card character for the version number. All versions of the file specified will be restored; you can then delete the unwanted files from the disk.

4.6.2.2 **/RC and /RA Switch Usage Prohibited** - Use of the /RC or /RA switch can cause RMSRST to fail to restore requested files. Do not use these switches.

4.7 Compatibility Mode

This section contains restrictions and problems known to exist in V2.0.

4.7.1 **FLX Loops after Receiving INVALID DEVICE Message** - Under certain conditions, FLX may loop after displaying the error message:

```
FLX -- INVALID DEVICE
```

If this occurs, press <CTRL/C> to interrupt FLX execution and return to the command interpreter.

4.8 System Programming

This section contains restrictions and problems known to exist in V2.0.

4.8.1 **SDA SET OUTPUT Command Produces Error** - When using the System Dump Analyzer (SDA) SET OUTPUT command, an error is produced when a process-permanent file (such as SYS$OUTPUT) is specified.

To reestablish your output stream to your terminal, use SET OUTPUT TT:.

4.8.2 **PATCH Utility Problems** - Currently, PATCH does not support some abbreviations allowed by the VAX-11 MACRO assembler. Unfortunately, the error message it produces in these situations:

```
%PATCH-E-NOENCODE, cannot encode instruction <instruction>
```

is the same message used when the instruction syntax is invalid.

A common cause of this error is a displacement mode address that is not preceded by an explicit displacement size operator (B^, W^, or L^). For example, the operands 4(AP) and @4(AP) should be written B^4(AP) and @B^4(AP), respectively. Note that the assembler allows the operand @0(AP) to be abbreviated @(AP): this operand should be written as @B^0(AP).

PATCH does not provide for the representation of long literals that are less than 64 in unsigned magnitude. PATCH will try to express such numbers as short literals. (The I^ and S^ operators are not supported by PATCH).

Should an assembly/disassembly problem be encountered in PATCH, it can be circumvented by patching in noninstruction mode.

4.8.3 **Restriction on Using Interlocked Queue Instructions** - A problem exists in the 11/780 microcode for the new queue instructions (INSQHI, INSQTI, REMQHI, and REMQTI) that causes incorrect operation if any page fault is incurred while referencing queue elements. The error manifests itself as a failure to correctly back up the PC to retry the instruction.

To avoid this problem, all pages containing queue elements must be locked into the working set or otherwise guaranteed valid for reference without a page fault. For the DR780 this will mean that the command buffer area must be locked in the working set by means of the $LKWSET system service. In the case of MA780 shared memory global sections, there is no need to lock those pages in the working set since they are valid at the time they are mapped and references to them do not incur page faults.

These new queue instructions are intended primarily for use with the DR780 and MA780. They should not be used in other applications until this problem is corrected.

4.9 System Management and Operation

This section contains restrictions and problems known to exist in V2.0.

4.9.1 **AUTOCONFIGURE Command of SYSGEN Utility** - The AUTOCONFIGURE command of the SYSGEN utility can disrupt I/O activity on an active controller if the command is used to reconfigure that controller. To prevent the problem, ensure that controllers are inactive when they are autoconfigured.

Because the system boot device controller is autoconfigured during system initialization, a device timeout error may be logged each time the system is bootstrapped. You can probably ignore a timeout error that is logged for the system volume immediately following the mounting of that volume: the device driver will correct the timeout error by retrying the I/O operation.

4.9.2 **Two SYSGEN Parameters Not Supported** - The SYSGEN parameters BJOBLIM and NJOBLIM are not supported. To control the maximum number of batch jobs, adjust the job limits on the batch queues. To control the amount of DECnet-VAX activity, use the NCP maximum links parameter.

4.9.3 **Generating RSX-11S and RSX-11M DECnet** - You cannot perform RSX-11S or RSX-11M DECnet generations on a VAX/VMS host system. A problem in the MCR command language interpreter prevents the successful execution of the indirect command files that are used.

4.10 EDT Computer-Assisted Instruction Course

There are two known EDTCAI problems.

In module 2, Objective 1, Subobjective B ("Repetition of Insertions"), the cursor is not positioned as EDT will position it.

When any section of the course is running, BROADCAST to the terminal is not allowed. The terminal is in PASSALL mode, which prohibits all broadcasts.

5.0 NOTES TO PUBLISHED DOCUMENTATION

This section contains notes that correct or supplement published documentation.

5.1 VAX/VMS I/O Users's Guide

5.1.1 **Foreign Magnetic Tape Skip Functions** - The following note supplements the material on skip file and skip record functions in Sections 4.4.4 and 4.4.5 of the VAX/VMS I/O User's Guide.

A skip file or skip record operation is terminated when both of the following conditions exist:

- The tape is mounted foreign
- Two consecutive tape marks are encountered in the forward direction

The tape is left positioned between the two tape marks. The I/O Status Block (IOSB) returns the status SS$_ENDOFVOLUME and the actual number of files (or records) skipped.

The status returned in the IOSB is SS$_ENDOFVOLUME. The actual number of records (or files) skipped during the operation prior to the detection of the second tape mark is returned in the high-order two bytes of the first longword of the IOSB. The tape remains positioned between the two tape marks.

Subsequent skip record (or skip file) requests from this tape postition will terminate immediately, producing no net tape movement and returning the SS$_ENDOFVOLUME status with a skip count of zero.

For more information, see Section 2.5.2 of these release notes.

5.1.2 **DR780 Support Routines** - The following information supplements the material in Chapter 11 of the VAX/VMS I/O User's Guide.

The documentation implies that the input queue of the DR32 can be preloaded (using a call to XF$PKTBLD) before starting the DR32 (using a call to XF$STARTDEV). Although this is true, note that at least one packet must be built and released onto the input queue after the DR32 has been started.

XF$PKTBLD is the only support routine that sets the DR32 "go" bit, which notifies the DR32 that there are command packets on the input queue that it must process.

Chapter 11 also refers to a "context array," an array that the user makes available to the support routines, which use it as a common storage area for information on the status of the current transfer.

The information in the context array is useful when you are debugging DR32 transfers. The first eight longwords of the context array are a copy of the fields of the most recently completed command packet. The next eight longwords make up the command table, a data structure used by the driver to define ranges of command and data memory and other parameters of the current transfer.

A useful address to know while debugging is the address of command memory at offset CTX$L_CMDBLK in the context array. The INPTQ, TERMQ, and FREEQ queue headers, which form the communication path between the DR32 and the application program, can be found in the first six longwords of the command block.

The context array has the following format:

Byte Offset	Bits	Contents
7:0	64:0	I/O status block
11:8	31:0	Device and command control
15:12	31:0	buffer size
19:16	31:0	Buffer address
23:20	31:0	Residual memory byte count
27:24	31:0	Residual DDI byte count
31:28	31:0	DR32 status longword (DSL)
35:32	31:0	Size of command block
39:36	31:0	Address of command block
43:40	31:0	Size of data block
47:44	31:0	Address of data block
51:48	31:0	Address of pre-AST routine
55:52	31:0	pre-AST parameter
	31:16	Reserved
	15:8	Flags
	7:0	Data rate
59:56	31:0	Address to receive the address of the "go" bit
63:60	31:16	Event flag number
	15:0	Number of buffers
67:64	31:0	Address of packet AST routine
71:68	31:0	Packet AST parameter
75:72	31:0	Size of each buffer in BARRAY
79:76	31:0	Address of IDEVMSG array
83:80	31:0	Address of ILOGMSG array
87:84	31:16	Size of IDEVMSG
	15:0	Size of ILOGMSG
91:88	31:0	Address of free memory list

5.1.3 **Documentation Error** - There is a documentation error in Chapter 11 of the VAX/VMS I/O User's Guide. The documentation of the Write Device Control Message in Table 11-2, Device Control Code Descriptions, should read:

This function specifies the transfer of a control message to the far-end DR device. This message is contained in the device message field of this packet. The Write Device Control Message function directs the controlling DR32 to send the following information over the Control Interconnect:

- One byte of command (command code XF$B_PKT_MSGLEN)
- One byte of device message length (offset XF$B_PKT_MSGLEN) into the command packet
- The device message field (offset XF$B_PKT_DEVMSG) into the command packet

5.2 VAX-11 Record Management Services Reference Manual

This section supplements information on the handling of foreign magnetic tapes.

Foreign magnetic tapes are considered non-file-structured devices. VAX-11 RMS treats the tape as a collection of blocks of data. The user can write multiple files on the tape, but is responsible for both tape positioning and interpreting of file information.

VAX-11 RMS will write tape marks (two EOFs) when a file is closed only when the last successful operation was a CREATE, PUT, or WRITE. Any other operation (OPEN, SPACE, READ, REWIND, or GET, for example) clears the context, preventing tape marks from being written.

VAX-11 RMS will block records if the user specifies both block size and record size. When you access the tape with a CREATE, VAX-11 RMS will take the block size and record size from the BLS and MRS fields in the FAB. When you access the tape with an OPEN, VAX-11 RMS will take the block and record size information from the MOUNT command.

Note that both block and record sizes must be specified as greater than 18 to prevent problems with reading short blocks.

5.3 VAX/VMS System Services Reference Manual

This section supplements information on the use of the $SNDACC system service.

The $SNDACC system service expects user messages to the accounting file to have the following format:

- The first word of the message buffer must be the symbolic constant ACC$K_INSMSG.
- The remainder of the buffer is a counted string, with the count in the first byte (just like an .ASCIC string).

Note that messages that are 133 to 200 bytes long will be truncated to 132 bytes without any error. Messages longer than 200 bytes cause the $SNDACC system service to return SS$_BADPARAM status in R0.

5.4 VAX-11 PATCH Utility Reference Manual

PATCH has a command qualifier that is not documented in the manual.

/VOLUME [=n]

Requests that the output file be placed on a specified relative volume of a multivolume set.

If you specify /VOLUME without supplying a value for n, the value defaults to the relative volume number of the input image file. If you do not specify the /VOLUME qualifier, the file is placed in an arbitrary position within the multivolume set.

5.5 VAX/VMS UETP User's Guide

This section supplements information on determining parameter settings for running UETP.

Table 4 contains suggested parameter settings for running UETP on average-size VAX-11/780 configurations. The parameter settings depend upon:

- The size of the system disk
- The VMS parameter file selected when the VAX system is booted
- Physical memory size

Note that the table also contains the suggested number of UETP loads to run, given the size of the system disk and physical memory.

The parameters represent a starting point and are not guaranteed to work on every system. The NPAGEDYN value can be increased in increments of 10000 if a larger configuration hangs during the UETP test.

Table 4
Suggested UETP Parameter Settings

RK07-based Systems:

Parameters	System Memory Sizes			
	256K	512K	768K	1024K (up)
PAR file	4USER.PAR	8USER.PAR	16USER.PAR	32USER.PAR
UETP loads	4	12	18	22
Swapfile size	4000	5120	7168	8192
Pagefile size	5120	7000	10000	10000
NPAGEDYN	50000	70000	90000	130000
WSMAX	150	200	200	200
SYSDUMP.DMP	1028	2052	3076	4100

(continued on next page)

Table 4 (Cont.)
Suggested UETP Parameter Settings

RP06/RM03-based Systems:

Parameters	System Memory Sizes			
	256K	512K	768K	1024K (up)
PAR file	4USER.PAR	8USER.PAR	16USER.PAR	32USER.PAR
UETP loads	4	20	30	40
Swapfile size	8192	24000	24000	24000
Pagefile size	16384	24000	24000	24000
NPAGEDYN	60000	use default	90000	130000
WSMAX	150	use default	500	500
SYSDUMP.DMP	1028	2052	3076	4100

5.6 VAX/VMS Command Language User's Guide

The following information supplements the information on the COPY command.

In general, the COPY command requires a minimum of 60 pages of working set to run. The larger the working set available for COPY, the more efficient COPY becomes.

If large records (greater than 512 bytes) are being copied, or a block size greater than 2048 bytes (on magnetic tape input or output) is specified, a working set larger than 60 pages may be required.

5.7 VAX-11 Utilites Reference Manual

The message name defined in Section 8.3.1.3 (the Message Utility) is incorrect. Replace the definition with the following:

name

> Up to 31 characters, as long as the message symbol made up of the prefix and name does not exceed 31 characters. This symbol name is combined with the symbol prefix defined in the facility definition to make up the message symbol.

6.0 INSTALLING AND TESTING VAX-11 COBOL-74

This section describes how to install and verify the installation of VAX-11 COBOL-74 on a VAX/VMS V2.0 operating system.

CAUTION

> If you have installed VAX-11 COBOL-74 on an earlier version of VAX/VMS, do not reinstall it after upgrading to VAX/VMS V2.0. Continue to use the version of VAX-11 COBOL-74 already installed.

6.1 Installing VAX-11 COBOL-74

The distribution kit for VAX-11 COBOL-74 includes three floppy diskettes labeled:

VAXC741 BIN RX01

VAXC742 BIN RX01

VAXC743 BIN RX01

You use these diskettes to install the product. Follow the optional product installation procedure described in Section 6.2 of the VAX-11 Software Installation Guide.

NOTE

> The release notes for VAX-11 COBOL-74 are in the File C74UPD.DOC on the distribution floppy diskette labeled VAXC741 BIN RX01.

To install VAX-11 COBOL-74, perform the steps described in Section 6.2.1 of the VAX-11 Software Installation Guide. However, instead of performing step 2, assign the logical name SYS$LIBRARY to the disk and directory that contains V2.0 of VAX/VMS.

The installation procedure runs under the control of the command procedure copied from the first diskette (VAXC741 BIN RX01). The command procedure copies the files from the floppy diskette, and upon completion, dismounts the device and requests the next floppy diskette (VAXC742 BIN RX01):

Please put the second COBOL-74 kit floppy in the drive

Remove VAXC741 BIN RX01 from the drive, and replace it with VAXC742 BIN RX01. The following message is displayed.

Are you ready to continue?

When you are ready, type Y<RET>.

After the command procedure has copied the files from the second floppy diskette, the following message is displayed on the terminal:

Please put the third COBOL-74 kit floppy in the drive

Remove VAXC742 BIN RX01 from the drive and replace it with VAXC743 BIN RX01. The following message is displayed.

Are you ready to continue?

When you are ready, type Y<RET>

The installation then continues to completion without further intervention.

When the procedure is completed, the following message is displayed:

Installation of COBOL-74 has completed successfully

The time required to install VAX-11 COBOL-74 is approximately 30 minutes.

When the installation is completed, the messages described in Section 6.2.3 of the VAX-11 Software Installation Guide will be displayed. You can terminate the session, or install other optional software products.

NOTE

The installation procedure places the file C74LIB.EXE in SYS$SYSTEM. You can use this shared image as an alternative to SYS$LIBRARY:C74LIB.OLB. Linking with a shared image is described in the VAX-11 COBOL-74 User's Guide; installing a shared image is described in the VAX/VMS System Manager's Guide.

6.2 Verifying Installation of VAX-11 COBOL-74

After the installation of the VAX-11 COBOL-74 compiler has been completed, the following sample test programs should be run to verify that the compiler is operating correctly.

There are three sample test programs in the distribution kit:

IO001.COB

IO002.COB

IO003.COB

IO001.COB creates a sequential disk file named PD1101 on the system disk. IO002.COB creates a relative disk file named PD1102 on the system disk. IO003.COB reads PD1101 from the system disk and lists the file in formatted form on the system disk.

There is also a file named UETPC74.COM, the command procedure for the sample test programs. All four files are copied into the directory [SYSTEST] during the COBOL installation procedure.

To use the sample test programs, perform the following steps:

1. Log in under the privileged system manager's account.

2. Issue the following SET commands to set the UIC and directory to the proper defaults:

```
SET UIC [1,4]
SET DEF [SYSTEST]
```

3. At the terminal, type:

```
@UETPC74
```

Each sample test program should run to a successful completion.

NOTE

The following ECO message may appear on your terminal:

CVTTP ECO not installed

If this message does appear, contact your local DIGITAL field service representative.

A.0 PATCHES MADE AFTER MICROFICHE KIT MANUFACTURED

This appendix contains listings of the journal files for patches applied to the V2.0 system after the source Microfiche Kit was manufactured.

1. SYS.JNL;1

2. RMS.JNL;1

```
  PATCH  Version 2.28  16-Nov-1979

IMAGE FILE BEING PATCHED:        "_DBA0:[SYSEXE]SYS.EXE;1"
JOURNAL FILE:                    "_DBA0:[SYSEXE]SYS.JNL;1"
DATE/TIME OF PATCH:               7-APR-1980 11:18:20.46

Image does not contain local symbols
Some or all global symbols not accessible
PATCH>SET ECO 1
PATCH>DEPOSIT/AS 80002650='V2.0'
old:     80002650:  'X0QK'
new:     80002650:  'V2.0'
PATCH>DEPOSIT/AS 80013F82='V2.0'
old:     80013F82:  'X0QK'
new:     80013F82:  'V2.0'
PATCH>UPDATE
Updating image file _DBA0:[SYSEXE]SYS.EXE;2
PATCH>EXIT
```

```
  PATCH  Version 2.28  16-Nov-1979

IMAGE FILE BEING PATCHED:        "_DBA0:[SYSEXE]RMS.EXE;1"
JOURNAL FILE:                    "_DBA0:[SYSEXE]RMS.JNL;1"
DATE/TIME OF PATCH:               9-APR-1980 12:12:28.96

%PATCH-I-NOLCL, image does not contain local symbols
%PATCH-I-NOGBL, some or all global symbols not accessible
PATCH>SET ECO 1
PATCH>SE PAT
NEW>  ^X0000F400
PATCH>RE /I
LOC>  ^X00002548
OLD>  'MOVL B^^X00000018(R4),B^^X00000060(R9)'
OLD>  EXIT
NEW>  'BLBS R0,X0'
NEW>  'MOVL R0,R2'
NEW>  'BICB2 #1,B^0A(R4)'
NEW>  'MOVL B^20(R9),R0'
NEW>  'BICB2 #1,B^0A(R0)'
NEW>  'CLRL R0'
NEW>  'BSBW 0AC7'
NEW>  'BRW ^X0000260D'
NEW>  'X0 :MOVL B^^X00000018(R4),B^^X00000060(R9)'
NEW>  EXI
old:    00002548:  MOVL     B^18(R4),B^60(R9)
old:    00002550:  BISB2    #02,B^0A(R4)
new:    00002548:  JMP      L^PAB
new:    00002551:  NOP
new:    00002552:  NOP
new:    00002553:  NOP
new:    PAB:  BLBS     R0,X0
new:    0000F40B:  MOVL     R0,R2
new:    0000F40E:  BICB2    #01,B^0A(R4)
new:    0000F412:  MOVL     B^20(R9),R0
new:    0000F416:  BICB2    #01,B^0A(R0)
new:    0000F41A:  CLRL     R0
new:    0000F41C:  JSB      L^00000AC7
new:    0000F422:  JMP      L^0000260D
new:    X0:  MOVL     B^18(R4),B^60(R9)
new:    0000F42D:  BISB2    #02,B^0A(R4)
new:    0000F431:  JMP      L^00002554
symbol "X0" defined as 00002565
symbol "X0" redefined from 00002565 to 0000F428
PATCH>RE /I
LOC>  ^X00002BCF
OLD>  'BBS #^X00000006,B^^X00000033(R9),^X00002C0B'
OLD>  'BISB2 #^X00000001,B^^X00000033(R9)'
OLD>  EXIT
NEW>  'BBC #^X00000006,B^^X00000033(R9),X1'
NEW>  'BBC #^X00000002,B^^X00000033(R9),^X00002C0B'
NEW>  'MOVZWL #^X00008454,R0'
NEW>  'POPR #^X0000093C'
NEW>  'RSB'
NEW>  'X1 :BISB2 #^X00000001,B^^X00000033(R9)'
```

```
NEW>  EXI
old:    00002BCF:  BBS      #06,B^33(R9),00002C0B
old:    00002BD4:  BISB2    #01,B^33(R9)
new:    00002BCF:  JMP      L^0000F437
new:    00002BD5:  NOP
new:    00002BD6:  NOP
new:    00002BD7:  NOP
new:    0000F437:  BBC      #06,B^33(R9),X1
new:    0000F43C:  BBS      #02,B^33(R9),0000F447
new:    0000F441:  JMP      L^00002C0B
new:    0000F447:  MOVZWL   #08454,R0
new:    0000F44C:  POPR     #093C
new:    0000F450:  RSB
new:    X1:  BISB2    #01,B^33(R9)
new:    0000F455:  JMP      L^00002BD8
symbol "X1" defined as 00002BE3
symbol "X1" redefined from 00002BE3 to 0000F451
PATCH>RE /I
LOC>  ^X00002C5E
OLD>  'MOVC3 R1,(R3),(R0)'
OLD>  'POPR #^X0000093C'
OLD>  'RSB'
OLD>  EXIT
NEW>  'MOVC3 R1,(R3),(R0)'
NEW>  'MOVL #^X00000001,R0'
NEW>  'POPR #^X0000093C'
NEW>  'RSB'
NEW>  EXI
old:    00002C5E:  MOVC3    R1,(R3),(R0)
old:    00002C62:  POPR     #093C
old:    00002C66:  RSB
new:    00002C5E:  JMP      L^0000F45B
new:    00002C64:  NOP
new:    00002C65:  NOP
new:    00002C66:  NOP
new:    0000F45B:  MOVC3    R1,(R3),(R0)
new:    0000F45F:  MOVL     #01,R0
new:    0000F462:  POPR     #093C
new:    0000F466:  RSB
new:    0000F467:  JMP      L^00002C67
PATCH>U
Non-contiguous image file being written
%PATCH-I-WRTFIL, updating image file _DBA0:[SYSEXE]RMS.EXE;2
PATCH>EXI
```

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 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_____

Please cut along this line.

Do Not Tear - Fold Here and Tape

digital

No Postage Necessary if Mailed in the United States

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO.33 MAYNARD MASS.

POSTAGE WILL BE PAID BY ADDRESSEE

BSSG PUBLICATIONS TW/A14
DIGITAL EQUIPMENT CORPORATION
1925 ANDOVER STREET
TEWKSBURY, MASSACHUSETTS 01876

Do Not Tear - Fold Here

Cut Along Dotted Line