VAX-11 RSX
Installation Guide
and Release Notes

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

The following are trademarks of Digital Equipment Corporation:

DEC	DIBOL	UNIBUS
DEC/CMS	EduSystem	VAX
DEC/MMS	IAS	VAXcluster
DECnet	MASSBUS	VMS
DECsystem–10	PDP	VT
DECSYSTEM–20	PDT	
DECUS	RSTS	
DECwriter	RSX	digital

ZK–2906

This document was prepared using an in-house documentation production system. All page composition and make-up was performed by TeX, the typesetting system developed by Donald E. Knuth at Stanford University. TeX is a registered trademark of the American Mathematical Society.

Contents

Contents

Preface

Manual Objectives

The *VAX-11 RSX Installation Guide and Release Notes* provides the system manager with a step-by-step procedure for installing VAX-11 RSX on a VAX/VMS or MicroVMS operating system. It also includes information pertaining to the Version 2.0 release of VAX-11 RSX.

Intended Audience

This manual is intended for system managers and system users familiar with both VAX/VMS and RSX-11 operating systems.

Structure of This Manual

This manual is divided into four sections and one appendix.

Section 1 describes the procedures that the system manager must follow to copy the files from the distribution kit and to install and verify the VAX-11 RSX software on a VAX/VMS system.

Section 2 describes the error messages displayed if the installation or verification fails and suggests possible user action.

Section 3 contains release notes for VAX-11 RSX. It describes the technical changes made in Version 2.0 as well as any information that supplements the *VAX-11 RSX Compatibility Mode Reference Manual*.

Section 4 contains release notes for RMS-11. It describes the technical changes made in Version 2.0 of VAX-11 RSX.

Appendix A contains information about DECUS, the Digital Equipment Computer Users Society, including details on how you can become a member.

Associated Manuals

The following manuals provide information that may help you during the VAX-11 RSX installation:

- *Guide to VAX/VMS Software Installation*
- *Guide to VAX/VMS System Management and Daily Operations*
- *VAX-11 RSX Compatibility Mode Reference Manual*
- *VAX/VMS Utilities Reference Volume*

Conventions Used in This Manual

The following conventions are observed in this manual.

Convention	Meaning
CTRL/x	A symbol that indicates the CTRL key; it must be held down while another key is pressed. For example, CTRL/C means hold down CTRL while pressing C.
xxx	A one- to three-character key symbol. For example, RET indicates the RETURN key.
n	A lowercase n; indicates a variable for a number.
,	A comma; separates parameters in commands.
.	A period; separates the file name from the file type in a file specification.
;	A semicolon; separates the file type from the file version number in a file specification.
...	Horizontal ellipsis; indicates that you can enter additional parameters, values, or other information.
. . .	Vertical ellipsis; indicates that not all the statements in an example or figure are shown.
>	The explicit prompt of the Monitor Console Routine (MCR), which is the command interpreter provided by VAX–11 RSX. The MCR prompting character appears whenever control is returned to the compatibility mode terminal and you can type input.
$	The VAX/VMS DIGITAL Command Language (DCL) prompting character. (Whenever control is returned to the native mode terminal and you can type input, this prompt appears.)
red ink	In interactive examples, what the user types is printed in red.
black ink	In examples, black ink indicates what the system prints or displays.
term	Boldface term; indicates that the term is being defined.

1 VAX–11 RSX Installation Guide

1.1 Introduction

VAX–11 RSX is a VAX/VMS and MicroVMS layered product that allows you to execute RSX–11 tasks on VAX processors. The *VAX–11 RSX Installation Guide and Release Notes* details the procedures necessary to install, verify, and maintain the VAX–11 RSX Version 2.0 software on VAX/VMS Version 4.2 and MicroVMS Version 4.2.

1.2 Identifying the VAX–11 RSX Kit

VAX–11 RSX Version 2.0 is distributed on the following types of media:

- Magnetic tape (1600 bpi)
- RK07 disks
- RL02 disks
- RC25 removable disks
- RA60 disks
- RX50 diskettes
- TK50 tape cartridges

Each distribution kit contains three files that are backup save sets. The following list describes each save set and its contents:

Save Set	Contents
RSX020.A	Contains the command files needed to perform both the software installation and the Installation Verification Procedure (IVP).
RSX020.B	Contains the task images, libraries, and other files that constitute VAX–11 RSX Version 2.0.
RSX020.C	Contains the VAX–11 RSX PDP–11 Instruction Set Emulator. This save set is installed on systems that do not have compatibility mode hardware or microcode.

NOTE: Before you install VAX-11 RSX on a MicroVMS system, the MicroVMS Base System must be installed. However, the following MicroVMS optional components may also be required, depending on your application:

- Common Utilites Option—For utilities available from MCR (for example, MAIL, DIFFERENCES, or SEARCH)

- Secure User Option—For /SP (spooling) with certain utilities (for example, PIP /SP)

- System Programming Option—To use Files-11 ODS-1 ACP for read/write operations to RSX-11 compatible disks

1.3   Getting Started

This section describes the steps you should take before installing VAX–11 RSX Version 2.0.

1 Be sure that the values in the SYSTEM account authorization record are equal to or greater than the default values in the following list:

Buffered byte count (BYTLM)	=	20480
Enqueued quota (ENQLM)	=	20
Direct I/O limit (DIOLM)	=	12
Buffered I/O limit (BIOLM)	=	12
Open file quota (FILLM)	=	20
AST limit (ASTLM)	=	20

Also make sure that the SYSTEM account does not have the DEFCLI option specified. When set, this option prevents the user of the SYSTEM account from logging in under a command interpreter (CLI) other than the DIGITAL Command Language (DCL), and causes the VAX–11 RSX Installation Verification Program (IVP) to fail.

To check these limits, run the VAX/VMS Authorize Utility by entering the following commands:

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

The system responds with the User Authorization File (UAF) prompt (UAF>). At the prompt, enter the following command:

```
UAF>  SHOW SYSTEM
```

The Authorize Utility displays the current values of the SYSTEM account's authorization record.

If necessary, you can change these values by entering the MODIFY command in response to the UAF prompt as shown in the following command line:

```
UAF>  MODIFY SYSTEM/limit=new-value
```

For example:

```
UAF>  MODIFY SYSTEM/DIOLM=12
```

To modify the DEFCLI option, enter the following command in response to the UAF prompt:

```
UAF>  MODIFY SYSTEM/FLAGS=NODEFCLI
```

Re-enter the SHOW command to verify that the new values have been set:

UAF> `SHOW SYSTEM`

To exit from the Authorize Utility, enter the EXIT command at the UAF prompt. You must then log out and log back in for the changes to take effect.

2 If you are installing VAX–11 RSX Version 2.0 on a VAX processor that supports compatibility mode hardware, be sure that your system has at least 9 unused global sections and at least 200 unused global pages.

If you are installing VAX–11 RSX on a VAX processor that does not support compatibility mode hardware or microcode, you must have at least 11 unused global sections and at least 3704 unused global pages. To determine how many global sections and global pages are available, enter the following commands:

$ `RUN SYS$SYSTEM:INSTALL`
INSTALL> `/GLOBAL`

If you have the VAX/VMS Version 4.0 Install Utility (INSTALL) defined as a foreign command in your SYSTARTUP.COM file, use the following command:

$ `INSTALL LIST/GLOBAL`

Read the last line of the screen display and note the following:

• The number of global sections **USED**

• The number of global pages **USED**

• The number of global pages **UNUSED**

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

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

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

Once you know how many unused global sections and global pages are available, use the following steps to modify the necessary parameters, as appropriate.(The procedure is similar if you are installing VAX–11 RSX on a VAX processor without compatibility mode hardware.)

• If the unused global section count is less than 9, increase the GBLSECTIONS parameter with the following command:

SYSGEN> `SET GBLSECTIONS new-value`

To compute the new value, add 9 to the number of used global sections shown by INSTALL.

- If the number of unused global pages displayed by the INSTALL /GLOBAL command is less than 200, increase the GBLPAGES parameter with the following command:

  SYSGEN>   SET GBLPAGES new-value

  To compute the new value, add 200 to the number of used global pages.

If you have modified either parameter (GBLPAGES or GBLSECTIONS), enter the following commands to write the new values into SYSGEN and exit:

    SYSGEN>   WRITE CURRENT
    SYSGEN>   EXIT

Note: **Neither GBLPAGES nor GBLSECTIONS is a dynamic parameter. You must rebootstrap your system before installing VAX-11 RSX Version 2.0 for the new value to take effect.**

3 Remove all unwanted or redundant files from the appropriate media to make sure there are enough free blocks to install VAX-11 RSX.

The following summary indicates the approximate disk block utilization requirements for VAX-11 RSX Version 2.0:

Disk Block Utilization	With Emulator	Without Emulator
Peak disk block utilization:	10000 blocks	5500 blocks
Net disk block utilization:	8900 blocks	5400 blocks

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

A significant number of system disk blocks are used to contain the work files required for the installation and the files to be read from the distribution volume. The number of blocks (5500 blocks) used during the installation is called the **peak disk block utilization**. DIGITAL recommends that you make enough blocks available on the system disk to cover peak utilization. If you choose not to do so, VMSINSTAL will operate in an alternate mode that reduces the peak utilization.

If a system failure occurs in the alternate mode, the installation will fail. To recover from this system failure, you must provide more free blocks and begin the installation again.

When the installation is complete, the system disk returns to its normal state. However, many disk blocks are permanently used up as a result of the installation process. The number of used disk blocks will be greater than the number in use before you began the installation. This number (5400 blocks) is called the **net disk block utilization**.

After you rebootstrap the system, you can recover additional disk blocks by purging system files that VMSINSTAL cannot purge during the installation.

Confirm the free block count by entering the following DCL command:

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

4   Back up your system disk using the Backup Utility. To invoke the utility, enter the following command and information:

```
$ BACKUP
From: indev:
To: outdev:
$
```

A system failure at a critical point during the installation may result in unusable files. Therefore, you should always back up your system disk before you attempt any software installation. You should use the copy created from the backup operation for the installation. For more information on the VAX/VMS Backup Utility, see the *VAX/VMS Utilities Reference Volume*.

5   Invoke an editor to modify SYSTARTUP.COM and perform the following:

*   Insert the following command line:

```
$ @SYS$MANAGER:VAX11RSX [spool_device]
```

This indirect command invokes VAX11RSX.COM. The command file VAX11RSX.COM installs VAX–11 RSX each time your system is bootstrapped, and establishes the environment in which VAX–11 RSX works.

During the installation of VAX–11 RSX, VAX11RSX.COM is placed in the system manager's account [SYSMGR]. The spool_device parameter, if specified, defines the logical device names SP and SP0. If this parameter is not specified, these logical device names are defined as LPA0.

The installation procedure also creates the command file VAX11RSX$DEVICE_NAMES.COM in the system manager's account (if the file did not already exist from a previous installation of VAX–11 RSX). This file initially defines $$n logical names for the disk and magnetic tape devices that were configured at the time of installation. It also defines $$n names for SYS$SPECIFIC and SYS$COMMON to allow references to them. The file is invoked by the VAX11RSX.COM command procedure. You may modify VAX11RSX$DEVICE_NAMES.COM to do any of the following:

—   Add additional devices or rooted directories

—   Delete definitions that are not needed (for example, $$n logical names)

—   Modify the $$n name unit number to improve clarity or efficiency (by giving the lowest unit numbers to the most frequently used devices)

You can include the systemwide page-length value that RSX–11 utilities (for example, the Task Builder (TKB) and the PDP–11 MACRO–11 Assembler (MACRO–11)) use to determine the number of lines per page for a printed listing. If you want to specify this value, include the following line in the definition table of SYSTARTUP.COM:

```
$ DEFINE/SYSTEM SYS$LP_LINES [page-length-value]
```

SYS$LP_LINES is used by the $PGLEN run-time library routine, which computes the default number of lines on a printer page. This routine can be used by any task that paginates and produces listing files.

The new page-length value range for SYS$LP_LINES is 30 to 99 lines per page. If you do not specify a value, the page-length value will be 66 lines per page by default.

- Remove any (CL, CO, and LB) logical name definitions from the file. Also, remove any start-up commands that install compatiblity mode components. VAX11RSX.COM specifies all the necessary logical name definitions and compatibility mode start-up commands.

6 Log in at the console terminal under the system manager's account (or any other account that has the SETPRV privilege).

7 Be sure that all users are logged out and that all batch jobs are completed by entering the following commands:

```
$  SHOW USERS
   .
   .
   .
$  SHOW QUEUE/BATCH
   .
   .
   .
```

8 Enter the following command to prevent users from gaining access to the system:

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

9 Make sure DECnet–VAX is turned off. If you are not sure whether DECnet–VAX is running on your system, enter the following command:

```
$  SHOW NETWORK
```

If the message "Network unavailable" appears, proceed to 1.4. If DECnet–VAX is running, turn it off by entering the following command:

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

The system responds with the Network Control Program (NCP) prompt (NCP>). At the prompt, enter the following commands:

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

1.4   Installing VAX–11 RSX

If you successfully completed all the steps in 1.3, you are ready to install VAX–11 RSX Version 2.0. The installation takes approximately 30 minutes to complete. At any time during the execution of VMSINSTAL, you can type a question mark (?) for help. Proceed in the following manner:

1 Log in at the console terminal using a privileged account (an account that has the SETPRV privilege).

2 Enter the following command line to initiate the installation:

```
$  @SYS$UPDATE:VMSINSTAL RSX020 ddcu:
```

where:

ddcu:

is the drive in which you have placed the VAX–11 RSX distribution kit.

The installation procedure is performed in two phases: the first phase copies the files and the second phase verifies that the VAX–11 RSX utilities are working (see step 4 in this section).

3 At the start of the first phase, you must answer the following question:

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

Assuming that you have performed the backup as recommended in 1.3, step 4, press the RETURN key (YES is the default response).

4 Next, VMSINSTAL prompts you as follows:

```
Please mount the first volume of the set on ddcu:
* Are you ready?
```

Physically mount the distribution software in the appropriate drive (if you have not already done so) and then answer YES.

A prompt will appear to request each additional volume. Enter YES after you insert each volume. When all the volumes have been copied, remove the last volume from the drive.

5 VMSINSTAL prompts you as follows:

```
Please select one of the following actions:
    1)  Perform the installation
    2)  Create a file with a description of the installation
    3)  Both of the above
* What would you like to do [3]:?
```

Choose option 1 if you want to perform the installation without creating a file that contains a description of the installation.

Choose option 2 if you do not want to perform the installation, but want the description file.

Choose option 3 (the default) if you want both to perform the installation and create a description file.

If you choose either option 1 or 3, VMSINSTAL responds with the following questions:

```
* Do you want to install the MCR help library [YES]?
```

This library supplies help for VAX–11 RSX MCR commands and for the RSX–11 utilities supplied with VAX–11 RSX. The library requires 2000 additional disk blocks. If you want the library, answer YES.

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

If you do not have sufficient free disk space or you do not want the old copies of the files, answer YES.

6 After you decide whether to install the library, the second phase of the installation begins with a VMSINSTAL question concerning the Installation Verification Procedure (IVP):

```
* Do you want to run the IVP after the installation [YES]?
```

IVP tests the various VAX-11 RSX components to make sure that each component was installed correctly and is working properly. IVP is not meant to be a comprehensive test of the components, but it does test some of the features of each component. For example, IVP tests LBR by having it create an object library, extract a module from that library, delete the module in the library, and then reinsert the extracted module into the library. Note that you can invoke IVP only while running VMSINSTAL.

When IVP is running and, for example, testing SLP, the following messages appear:

```
%RSX020-I-IVPSTART, Installation Verification Procedure starting
%RSX020-I-TESTSTART, SLP testing started at 10:17:18
%RSX020-S-TESTSUCCESS, SLP testing completed successfully at 10:17:27
%RSX020-S-IVPSUCCESS, the installation verification of VAX-11 RSX Version 2.0 succeeded
```

Various compatibility mode images tested during the IVP may produce status messages that are not in standard VAX/VMS format. However, if an error occurs, IVP will detect it and produce the error message in the standard VAX/VMS format.

Refer to 2.2 if an error message is displayed during the Installation Verification Procedure (IVP). There you will find a description of each error message and information on how to correct the error.

At this point, you have answered all the questions necessary to install VAX-11 RSX successfully. The VMSINSTAL procedure will complete your VAX-11 RSX Version 2.0 installation. See the *Guide to VAX/VMS Software Installation* for more information about VMSINSTAL.

7 When the installation is complete, VMSINSTAL invokes the start-up procedure SYS$MANAGER:VAX11RSX.COM (which VMSINSTAL copied from the distribution software into the system manager's account (SYSTEM)), and then exits. You can now use VAX-11 RSX Version 2.0.

8 For a complete description of the new images and files that VAX-11 RSX uses (if you chose either option 2 or option 3 in step 5), please read SYS$UPDATE:RSX020.TXT.

1.5   After Installation

This section lists the steps that you should follow after the installation of VAX-11 RSX.

1 Read the console listing carefully. Look for warning or error messages that indicate steps you must perform manually. Many informational messages will also be displayed. Be sure to read them.

2 Perform an incremental backup of the system disk.

3 Establish minimum user account quotas for individual users (for example, set quotas for open file, paging file, working set size, and subprocess creation).

1.6 A Sample VAX—11 RSX Installation

The following example contains the text of a typical VAX—11 RSX Version 2.0 installation on VAX/VMS Version 4.0, using the RK07 distribution kit.

```
Username: SYSTEM
Password: RET
Welcome to BABBLE!

   Last interactive login on Thursday, 19-SEP-1985 09:45
   Last non-interactive login on Monday, 16-SEP-1985 20:00
$ @SYS$UPDATE:VMSINSTAL RSX020 DM0:
         VAX/VMS Software Product Installation Procedure V4.0

It is 19-SEP-1985 at 10:01.
Enter a question mark (?) at anytime for help.
* Are you satisfied with the backup of your system disk [YES]? RET

Please mount the first volume of the set on DM0:
* Are you ready [YES]? RET
The following products will be processed:
  RSX  V2.0

        Beginning installation of RSX V2.0 at 10:02
%VMSINSTAL-I-RESTORE, Restoring product saveset A...

Please select one of the following actions:

            1)  Perform the installation
            2)  Create a file with a description of the installation
            3)  Both of the above

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

An optional  help library is  provided for use  with the MCR CLI.
This library  supplies help for commands supported by MCR and for
the RSX  utilities supplied  with VAX-11 RSX.  This  library will
require approximately 2000 additional disk blocks. If you wish to
install this library on your system, answer yes to the  following
question:
* Do you want to install the MCR help library [YES]?  RET
* Do you want to purge files replaced by this installation [YES]?  RET
* Do you want to run the IVP after the installation [YES]?  RET

VAX11RSX.COM, the startup command procedure, is used to set up
the environment for VAX-11 RSX.  During the installation it will
be placed in the [SYSMGR]  directory of the system root on which
this  installation  is being  performed.   Your  system  startup
procedure  should be modified to invoke this procedure  when the
system  boots.  However, it will not be  necessary to reboot the
system  after the installation, since this procedure is  invoked
as part of  the installation.  See the  VAX-11 RSX  Installation
Guide and Release Notes,  or the procedure  itself for information.

The installation  procedure has no  further questions to ask and
can continue from this point without user assistance.   The rest
of the installation will take  approximately 10 minutes.  If you
chose to run the IVP, it will take an additional 10 minutes.

%VMSINSTAL-I-RESTORE, Restoring product saveset B...
```

Note: The following text will be displayed only if you are installing VAX–11 RSX on a VAX processor (for example, a MicroVAX) that does not contain compatibility mode hardware or microcode.

```
                   ** PDP-11 Instruction Set Emulator **
Since your VAX processor does not include support for compatibility
mode within hardware or microcode, the VAX-11 RSX PDP-11 Instruction
Set Emulator is being installed.

%VMSINSTAL-I-RESTORE, Restoring product saveset C...
```

Note: The following text will be displayed only if you are installing VAX-11
RSX on a SYSCOMMON system (that is, SYS$SYSROOT is a searchlist
directory on a VAXcluster running on a common system disk).

```
                      ** VAXcluster common system disk setup **
If you intend to execute VAX-11 RSX on other nodes in your VAXcluster
which share this common system root, and you have the appropriate
software license, you must ensure that SYS$MANAGER:VAX11RSX.COM is
executed on each node to define the appropriate logical names and to
create the system directory synonyms required by RSX-11 software.
                        ** System $$n logical names **
RSX-11 tasks running under VAX-11 RSX can only access VAX/VMS rooted
directories through the use of logical names having a format which
resembles that of an RSX-11 physical device name. Similarly, many VAX/VMS
physical devices (including any device on a VAXcluster) can only be
accessed by RSX-11 tasks through the use of these logical names.

VAX-11 RSX uses several standard system-wide logical names, as well as
internal "logical names" (such as SY and TI) for this purpose. See
the VAX-11 RSX Compatibility Mode Reference Manual (Section 2.7) for
more detailed information on VAX-11 RSX device name mapping.

The system directory root (SYS$SYSROOT) is always mapped by the VAX-11
RSX logical name LB. However, other devices and root directories should
be mapped by "$$n" logical names, with the string "$$" followed by
an octal unit number from 0 to 377. Note that even though $$n
logical names are not strictly necessary (except for rooted directories)
on non-VAXcluster systems, RSX-11 task performance
(particularly the ALUN$ and ELP$ directives) will be improved through
their use. This installation will create $$n names for all the disk
and tape devices currently present on your system.

%RSX-I-$$N, the following $$n logical names will be defined:

         $$n       VAX/VMS device
         -----     ---------------
         $$0       BABBLE$DMA0:
         $$1       BABBLE$DMA1:
         $$2       BABBLE$DRA0:
         $$3       BABBLE$DRA1:
         $$4       BABBLE$DRA2:
         $$5       BABBLE$MFA0:

The file SYS$MANAGER:VAX11RSX$DEVICE_NAMES.COM contains DEFINE commands
to create each of the $$n logical names listed above. The VAX11RSX.COM
startup procedure will execute VAX11RSX$DEVICE_NAMES.COM automatically
when it runs following this installation, and during future system
startups.

If there are devices which were not displayed above, which will be
commonly accessed by compatibility mode tasks on your system, you may
add $$n names to VAX11RSX$DEVICE_NAMES.COM. You may also delete $$n
names which you do not want to be defined system-wide from this file.
Additionally, you should define $$n names for any rooted directories
which will be commonly accessed look at the current definition of LB
with the command "SHOW LOGICAL/FULL LB", for an example of this.

You may also want to consider defining the logical name
RSX$DEVICE_CACHE to control VAX-11 RSX use of $$n names; it is
described more fully in the VAX-11 RSX Installation Guide and Release
Notes, and in the VAX-11 RSX Compatibility Mode Reference Manual.

Additionally, to improve performance, you may wish to reorganize the
$$n names which have been defined, so that those which are most commonly
used have the lowest unit numbers. Currently, they have been defined
in the order in which SHOW DEVICE displays them.

VAX11RSX$DEVICE_NAMES.COM will not be modified on subsequent
installations of VAX-11 RSX. If you wish to perform a future
installation that will require changes to the VAX11RSX$DEVICE_NAMES.COM
file, the old file must be deleted first. The installation will then
create a new copy of VAX11RSX$DEVICE_NAMES.COM.

%RSX-I-DESCRIBE, The installation is described in SYS$UPDATE:RSX020.TXT
```

```
 1) BACKTRANS (new image)
 2) BCK (new image)
 3) BRU (new image)
 4) CNV (new image)
 5) CRF (new image)
 6) DEF (new image)
 7) DES (new image)
 8) DMP (new image)
 9) DSC1 (new image)
10) DSP (new image)
11) EDI (new image)
12) FLX (new image)
13) FCSRES (new image)
14) ICM (new image)
15) IFL (new image)
16) INDSYS (new file)
17) LBR (new image)
18) LINK (miscellaneous)
19) MAC (new image)
20) MCR (new image)
21) MCRHELPLIB (new file)
22) MCRTABLES (miscellaneous)
23) MFT (new image)
24) ODT (new file)
25) OLDSYLIB (miscellaneous)
26) OLDTKB (miscellaneous)
27) PAT (new image)
28) PIP (new image)
29) QIOSYM (new file)
30) RMS11ODL (new file)
31) RMSDES (new file)
32) RMSLIB (new file)
33) RMSMAC (new file)
34) RST (new image)
35) RSX (new image)
36) RSXMAC (new file)
37) RSXSHR (new image)
38) RSXUSR (new image)
39) SLP (new image)
40) SYSLIB (new file)
41) TKB (new image)
42) VAX11RSX (new file)
43) VMLIB (new file)
44) ZAP (new image)
```

Note: For VAX/VMS Version 4.0 and Version 4.1 host systems, the LINK command definition has been modified to support the new LINK/RSX11 qualifiers.

Also, if the PDP–11 Instruction Set Emulator is installed, the CEM$EMULATOR image file is also included.

```
%VMSINSTAL-I-MOVEFILES, Files will now be moved to their target directories...
%RSX020-I-IVPSTART, Installation Verification Procedure starting
```

The Installation Verification Procedure (IVP) tests that VAX-11 RSX installed correctly by testing various compatibility mode utilities. This serves as a test of the utilities themselves and of the RSX AME. Also, when DCL commands exist for compatibility mode utilities (i.e. LIBRARY/RSX for LBR), these will be used to test the translation of DCL commands to MCR syntax for these utilities.

Several of the compatibility mode utilities will log their normal actions to SYS$OUTPUT. These informational messages as well as any error messages that may be produced by these utilities, will not be in the standard VAX/VMS message format. If an error occurs, the IVP will detect it and produce an error message in the standard VAX/VMS format. The IVP will also produce informational and success messages in the standard format.

```
%RSX020-I-TESTSTART, DMP testing started at 10:12:32
DMP -- DMP Version M08.01 (ANSI)

%RSX020-I-TESTSUCCESS, DMP testing completed successfully at 10:13:27
%RSX020-I-TESTSTART, LBR testing started at 10:13:27
Modules deleted:
OBJ3

Module "OBJ1 " replaced

Module "OBJ2 " replaced

Entry points deleted:
ENT$A1
ENT$B2
ENT$A3

Modules deleted:
MCLBR3

Module "MCLBR1" replaced

Module "MCLBR2" replaced

%RSX020-S-TESTSUCCESS, LBR testing completed successfully at 10:15:52
%RSX020-I-TESTSTART, PAT testing started at 10:15:53
%RSX020-S-TESTSUCCESS, PAT testing completed successfully at 10:15:58
%RSX020-I-TESTSTART, PIP testing started at 10:15:58
PIP -- PIP Version M1600 (ANSI)

%RSX020-S-TESTSUCCESS, PIP testing completed successfully at 10:17:14
%RSX020-I-TESTSTART, FLX testing started at 10:17:14

FLX -- FLX Version 16.07

%RSX020-S-TESTSUCCESS, FLX testing completed successfully at 10:17:18
%RSX020-I-TESTSTART, SLP testing started at 10:17:18
%RSX020-S-TESTSUCCESS, SLP testing completed successfully at 10:17:27
%RSX020-S-IVPSUCCESS, the installation verification of VAX-11 RSX Version 2.0 succeeded
        Installation of RSX V2.0 completed at 10:17
        VMSINSTAL procedure done at 10:17
```

2 VAX-11 RSX Installation and Verification Messages

This section describes the messages that are displayed during the VAX-11 RSX installation and verification procedure, and suggests possible user action, if appropriate. Messages are displayed or printed in both uppercase and lowercase characters (when the output device is capable of displaying or printing lowercase characters).

In this section, the fixed and variable portions of the text of a message are distinguished by enclosing the variable portions in apostrophes (single quotation marks).

2.1 Message Format

Messages displayed by VAX-11 RSX have the following format:

```
%FACILITY-L-IDENT, TEXT
[-FACILITY-L-IDENT,TEXT]
```

where:

FACILITY

is a VAX-11 RSX facility or component name. A percent sign (%) prefixes the first message issued, and a hyphen (-) prefixes each subsequent message. For the VAX-11 RSX kit installation procedure, the facility name is RSX020.

L

is a severity level indicator with one of the following values:

Severity Level	Meaning
S	Success
I	Information
W	Warning
E	Error
F	Fatal, or severe error

IDENT

is an abbreviation of the message text; the message descriptions are alphabetized by this abbreviation.

TEXT

is the explanation of the message.

2.2 Message Summary

CONSOLE, console device CSA1: not configured; defining $$n anyway

Explanation: The installation program has assigned the console mass storage device a $$n name even though the console has not been connected to the system (using the system generation CONNECT CONSOLE command).

User Action: None.

DESCRIBE, the installation is described in SYS$UPDATE:RSXvvu.TXT

Explanation: Indicates where the installation description can be found after the installation is complete.

User Action: None.

ERRORS, of the "nn" action listed above, the following "nn" should be reviewed:

Explanation: If errors occur during the software installation, this message will summarize the points at which the errors occurred.

User Action: Review the specified actions.

IVPFAILURE, the installation verification of VAX–11 RSX "version" failed

Explanation: The installation did not pass the software Installation Verification Procedure (IVP). Previous error messages should have indicated the specific reason(s) for the failure.

User Action: Submit a Software Performance Report (SPR) describing the problem and include the console listing of the software installation.

IVPSTART, starting installation verification procedure

Explanation: IVP is beginning.

User Action: None.

IVPSUCCESS, the installation verification of VAX–11 RSX "version" succeeded

Explanation: IVP did not detect any errors in the software installation.

User Action: None.

KEEPOLD, retaining previous SYS$MANAGER:VAX11RSX$DEVICE_
        NAMES.COM

Explanation: A $$n name definition file already exists for your system. It will not be superseded.

User Action: None.

NOBLOCKS, RSX "version" requires "n" additional blocks after installation

Explanation: After installation, the specified version of VAX–11 RSX will use the specified number of blocks on the system disk. If the specified number of blocks is not available, the installation will fail.

User Action: Purge or delete unnecessary files from the system disk.

NO$$N,   unable to read device information to create $$n names

Explanation: An error occurred while attempting to access the SHOW DEVICE command output.

User Action: You must manually add $$n name definitions to the VAX11RSX$DEVICE_NAMES.COM command file.

SOME,   unable to collect all device information

Explanation: An error occurred during the processing of device information while creating $$n names.

User Action: You must manually complete the list of $$n definitions in the VAX11RSX$DEVICE_NAMES.COM command file.

TESTERROR,   "utility" fail test of "test description"

Explanation: An error occurred while performing the specified utility test during the IVP.

User Action: Submit an SPR describing the problem and include the console listing of the software installation.

TESTSTART,   "utility" testing started

Explanation: IVP has begun testing the specified utility.

User Action: None.

TESTSUCCESS,   "utility" testing completed successfully

Explanation: IVP detected no errors.

User Action: None.

VERSION,   RSX "version" must be installed on existing VAX/VMS "version" system

Explanation: This version of VAX–11 RSX can be installed only on the specified version of VAX/VMS.

User Action: You have the wrong version of either VAX–11 RSX or VAX/VMS. You must obtain the correct version.

$$N,   the following $$n logical names will be defined:

Explanation: The installation will list the $$n names created for disks and magnetic tapes on your system.

User Action: None.

3 VAX-11 RSX Version 2.0 Release Notes

3.1 VAX-11 RSX Software Release Notes

This section contains a brief description of all new and modified software features for VAX-11 RSX Version 2.0. Many of these features are more completely documented in the *VAX-11 RSX Compatibility Mode Reference Manual*.

3.1.1 VAX-11 RSX PDP-11 Instruction Set Emulator

The VAX-11 RSX distribution software consists of the following three save sets:

1 RSX020.A, which contains the command files needed to perform both the software installation and the Installation Verification Procedure (IVP).

2 RSX020.B, which contains the task images, libraries, and other files that constitute VAX-11 RSX Version 2.0.

3 RSX020.C, which contains the VAX-11 RSX PDP-11 Instruction Set Emulator.

During the installation process, the VAX-11 RSX PDP-11 Instruction Set Emulator save set will be installed on your system disk if your VAX processor does not include hardware or microcode support for compatibility mode. See Section 10 of the *VAX-11 RSX Compatibility Mode Reference Manual* for more information about the Emulator.

3.1.2 Changes to MCR

The following sections describe changes to the Monitor Console Routine (MCR) for VAX-11 RSX Version 2.0.

3.1.2.1 New Commands
DELETE/INTRUSION_RECORD

Removes an entry from the break-in database.

SHOW INTRUSION

Displays the contents of the break-in database.

3.1.2.2 New or Revised Keywords and Parameters
RUNOFF

/DEVICE=(option[,...])—Controls whether DSR (Digital Standard Runoff)
generates an output file (LNI) that is suitable for printing on LN01, LN01E, or
LN03 laser printers.

/REVERSE_EMPHASIS—Directs DSR to change the order in which flagged
text is underlined on an output device.

3.1.3 Changes to DCL

The following sections describe changes to the DIGITAL Command Language
(DCL) documentation for VAX–11 RSX Version 2.0.

3.1.3.1 Enhancement to the DCL Command MCR
The DCL command MCR enables RSX–11 components top be run in a
manner that is compatible with the RSX–11 operating system. If you wish,
may include a component name on the MCR command line to invoke
that component. However, you may enter only an RSX–11 or VAX/VMS
component and a command line for that component.

If you type the command MCR and press the RETURN key, the MCR
command interpreter (CLI) prompts for command input, placing you in
prompting mode. This mode allows access to all the commands in the MCR
CLI. To exit from MCR, press CTRL/Z. You will be returned to DCL.

Commands that are issued in this mode have their input device (SYS$INPUT)
assigned to a mailbox. Native mode images that read from SYS$INPUT (TI)
should not be used in this mode (for example, HELP or the EDT editor).
RSX–11 tasks which read from TI may be used, but only if they perform
simple line-oriented read operations (such as the command prompting mode
of most RSX–11 utilities).

3.1.3.2 New LINK/RSX11 Qualifiers
The following command qualifiers have been added to the list of qualifiers
supported by the DCL command LINK/RSX11:

/[NO]CHECKPOINT=keyword

Allows a task image to be checkpointed (that is, the task may be swapped
out). (Only one keyword, SYSTEM or TASK may be specified.)

/[NO]COMPATIBLE

Aligns memory-resident overlays on 512-byte boundaries to correspond to
VAX/VMS memory management allocations.

/ERROR_LIMIT[=n]

Causes the Task Builder to exit after n errors have occurred. When n is not
specified, the default is 1. This qualifier is a synonym for /EXIT[=n].

/LONG

Refers to map size. This qualifier is a synonym for /FULL.

/OPTIONS=filespec

Allows you to specify the name of a file containing Task Builder options. This qualifier is a synonym for /TASK_OPTIONS.

/[NO]RECEIVE

Allows the default task to receive messages from the Executive Send Data (SDAT$) directive.

/[NO]RESIDENT_OVERLAYS

Builds memory-resident overlays into the image by using the memory-resident overlay operator.

/[NO]SEGREGATE

Allocates virtual address space containing contiguous program sections first to the read-write (RW) program sections and then to the read-only (RO) program sections.

/SHAREABLE=keyword

Instructs the Task Builder to build a shareable common, a shareable library, or a shareable task image. (Only one keyword, COMMON, LIBRARY, or TASK may be specified.)

/SLOW

Invokes the Slow Task Builder, which is used to build more complex task images.

3.1.4   Changes to the Indirect Command Processor (Indirect)

The following sections describe the changes for the Indirect Command Processor (Indirect) that is supported under VAX–11 RSX. These changes include new directives and a new symbol.

3.1.4.1   New Directives
.ENABLE CLI

Passes commands that are not processed by Indirect to your command interpreter (CLI).

.ENABLE DELETE

Deletes the current command line when Indirect processes the last command line in the file.

.TRANSLATE

Allows a command file to expand a local or global logical name assignment.

3.1.4.2   New Symbol
< SYTYP>

Contains the "VAX–11 RSX" system indentification string.

3.1.5 Support for New VAX–11 RSX Executive Directives

VAX–11 RSX Version 2.0 includes support for a subset of the RSX–11 memory management (sometimes referred to as Program Logical Address Space (PLAS)) directives. VAX–11 RSX now supports the following RSX–11 memory management directives:

ATRG$

Attaches a region

CRAW$

Creates an address window

CRRG$

Creates a region

DTRG$

Detaches a region

ELAW$

Eliminates an address window

GMCX$

Gets the mapping context

GREG$

Gets the region parameters

MAP$

Maps an address window

UMAP$

Unmaps an address window

This support allows RSX–11 task images running on the VAX/VMS operating system to use RSX–11 memory management directives. VAX–11 RSX memory management includes the following RSX–11 features:

- Task and library memory-resident overlays

- Shared resident libraries and commons

- Cluster libraries

- Virtual sections

- Shared data commons with VAX/VMS native images

Additionally, VAX-11 RSX now supports the following logical name directives that are new to RSX-11M-PLUS Version 3.0 and Micro/RSX Version 3.0:

ACHN$

Assigns channels to a logical name

CLON$ (CLOG$)

Creates a logical name

DLON$ (DLOG$)

Deletes a logical name

RLON$ (RLOG$)

Recursive translation of a logical name

TLON$ (TLOG$)

Translates a logical name

GDIR$

Gets the default directory

FEAT$

Tests for a system feature

SDIR$

Sets the default directory

Because it supports these directives, VAX-11 RSX is fully compatible with RSX-11M/M-PLUS and Micro/RSX operating systems in the following areas:

- Support of static and dynamic regions

- Support of shared regions

- Support of memory-resident overlaid tasks

- Support of tasks linked to memory-resident overlaid libraries (including FCSRES)

- Support of tasks linked to cluster libraries

- Support of virtual sections

- Support of FORTRAN-77 virtual arrays

Note: **Because FCSRES is vectored, tasks can be transported between VAX-11 RSX Version 2.0 and RSX-11M-PLUS Version 3.0, Micro/RSX Version 3.0, and RSX-11M Version 4.2**

FORTRAN-77 tasks using virtual arrays may be transferred between VAX-11 RSX and any RSX-11 operating system. VAX-11 RSX places no restrictions on the use of FORTRAN-77 virtual arrays.

3.2 Enhancements to Existing VAX–11 RSX Features

The following sections describe the enhancements to VAX–11 RSX. Many of these features are also documented in greater detail in the *VAX–11 RSX Compatibility Mode Reference Manual*.

3.2.1 Support for SYSCOMMON System Disks

The Applications Migration Executive (AME) has been enhanced to support device name and file-handling operations. These operations now perform compatibly with SYSCOMMON system disks or with any VAX/VMS search as long as the logical name (and its translation strings) has been repetitively translated until either of the following conditions occur:

- Translations are marked with the TERMINAL attribute (LNM$M_TERMINAL).

- Strings that have no translations are reached on all branches of the search list.

Each of these final strings must contain only a physical device name, or a physical device name combined with a rooted directory name.

Note that if a file is opened by file identification rather than by file name, or if a file is spooled using the PRINT$ routine, there is a small possibility that an incorrect file will be found. This may occur if the terminal translations of the VAX/VMS logical name used as a "device" are on different physical devices and if the specified DID and FID exist on a device prior to the intended one. To avoid such an occurrence, DIGITAL recommends that you do not use search lists spanning multiple devices. However, file access by directory and file name (the normal case) is no more susceptible to this problem than native mode utilities using search lists. (In other words, if the directory and file name exist on a device prior to the intended one, both RSX–11 and VAX/VMS programs will find the same "incorrect" file.)

3.2.2 Performance Enhancements to the ALUN$ and ELP$ Directives

For VAX–11 RSX Version 2.0, the software that assigns, parses, and maps device names has been redesigned and reimplemented, resulting in a significant performance improvement for most tasks that use the Assign Logical Unit Number (ALUN$) and Extended Logical Parse (ELP$) directives.

3.2.3 $$N Translation Caching

Individual users should define $$n names for their own use with high $$n unit numbers (starting at 347 (8) and descending to lower numbers). However, because $$n names defined dynamically by programs have unit numbers beginning at 350 (8) and ascending to higher unit numbers, you may have to attempt many translations before you arrive at these values.

As a result, the VAX–11 RSX Version 2.0 AME will cache the translation of LB, SP, and $$n names the first time it translates them. After caching the translation, the AME will no longer have to translate those particular names. The AME will always attempt to translate $$n names with units higher than 350 (8) since these may be dynamically defined by other programs while the

task is running, and even names that were already translated may change during task execution.

The implication of this caching for users is that, with the exception of dynamic $$n names, any changes to $$n names after the task begins may not affect the operation of the task. However, this behavior may be altered. See the *VAX–11 RSX Compatibility Mode Reference Manual* for a description of RSX$DEVICE_CACHE for more information.

3.2.4 Terminal Driver Enhancements

Enhancements have been made to the VAX–11 RSX terminal driver, which allow the driver to operate like the RSX–11 terminal driver. The terminal driver contains the following new or modified features:

Break-through Write Function

Maps the RSX–11 function code IO.WBT to the VAX/VMS function, IO_WRITELBLK!IO$M_BREAKTHRU.

TC.RAT Terminal Characteristic

Maps the RSX–11 characteristic TC.RAT (Type-ahead buffer) to the VAX/VMS characteristic TT$M_NOTYPEAHD.

IO.RAL Function Code

Maps the RSX–11 function code IO.RAL to the VAX/VMS function code IO$_TTYREADALL. By using IO$_TTYREADALL, no interpretation of data will occur.

IO.RLB Function Code

When this code is combined with both the subfunction bits TF.RAL (Read All) and T.RST (Read Special Terminators), VAX–11 RSX will not ignore the TF.RST subfunction. The special characters terminate the read operation.

3.2.5 Reactivating RSX–11 Images

On VAX–11 RSX Version 2.0, the AME restores all necessary initial context so that it is possible to reactivate RSX–11 task images (for example, when using the RUN/INTERVAL=delta-time command).

3.2.6 Reporting CRF Subprocess Errors

VAX–11 RSX Version 2.0 provides a way of reporting Cross-Reference Processor (CRF) subprocess errors. A procedure has been created that allows a subprocess task to write to a mailbox owned by the main process. This procedure is referred to as output chaining. The Applications Migration Executive (AME) then writes this data to its SYS$OUTPUT file, which, in turn, allows the CRF task to report errors.

Note that any task that is spawned with the parent's terminal input pseudodevice (TI), when the output half of TI is a Record Management Services (RMS–11) file, will automatically and transparently make use of output chaining to write data to TI.

3.2.7   EXST$ Directive Behavior Change

In VAX–11 RSX, the Exit With Status (EXST$) directive now uses an RSX–11 status passback procedure to provide the parent process with the full RSX–11 status word. However, because the VAX–11 RSX passback procedure is a protocol initialized between the AME running in the parent process and the MCR CLI running in the subprocess, this full word status passback is used only for tasks activated as the result of a spawned MCR CLI command. The enhancement, to the Exit With Status directive allows Indirect command procedures to use the <EXSTAT> symbol to access the full RSX–11 status word from any MCR commands executed.

3.2.8   DECnet–11M and DECnet–11M–PLUS NETGEN Support

Because VAX–11 RSX supports the return of the full RSX–11 status word of a spawned CLI command to an Indirect command procedure, it is possible to perform NETGEN operations for DECnet–11M and DECnet–11M–PLUS. To do this, however, you must have DECnet–11M Version 4.2 or DECnet–11M–PLUS Version 3.0

3.2.9   Changes to the Indirect .TESTPARTITION and GPRT$ Directives

The .TESTPARTITION directive allows a command file to acquire information about any region known to VAX–11 RSX. The Get Partition Parameters (GPRT$) directive allows a task to acquire information about any region known to VAX–11 RSX. Any of the following are considered known regions:

*   The GEN partition, which is the task region of the current task

*   A group-global section created by VAX–11 RSX

*   A group-global section created by a VAX/VMS native mode image

*   A system-global section created by the VAX/VMS Install Utility

*   A valid image file of a resident library or common located in SYS$LIBRARY

*   A valid image file of a resident library or common to which the region's name translates (if the latter is defined as a logical name)

3.2.10   Enhancements to VAX–11 RSX Error Messages

Along with changes made to some existing VAX–11 RSX error messages, there are some new error messages that have been added for Version 2.0 of VAX–11 RSX. See Section 10 of the *VAX–11 RSX Compatibility Mode Reference Manual* and Section 2 of this manual for a description of these messages.

Unlike the previous version of VAX–11 RSX, in which most errors caused the AME to display a single error message, the AME now displays a sequence of one to three error messages. The first line of the message indicates the general area of the error. The second line gives provides more detailed information, and the final line notes the VAX/VMS error status that caused the failure. The following is a typical illustration of this sequence:

%RSX-F-ERRINILIB,   unable to initialize library/common \FCSRES\

-RSX-F-CRSCERR, error creating the "RSX$FCSRES" global section

-SYSTEM-F-GPTFULL, global page table is full

3.2.10.1 New Error Message

Under VAX-11 RSX Version 2.0, the incorrect use of a high-level language subroutine or an error detected during the loading of an overlay segment results in task termination by means of a breakpoint (BPT) instruction. The task aborts and one of the following error messages is displayed:

%RSX-F-ERRLANGIN, error in high level language interface

%RSX-F-ERRLODOVR, load overlay failure

Both errors are followed by the message:

%RSX-F-IMAGETERM, \Process name,\ terminated, \date\ \time\

3.2.11 BOM$ Directive Error Messages

For compatibility with RSX-11M-PLUS Version 3.0 and Micro/RSX Version 3.0, VAX-11 RSX includes the overlay control error (ALERR) and high-level language interface error (XERR) routines in SYS$LIBRARY:SYSLIB.OLB. These routines use the Breakpoint Or Message (BOM$) directive which is not documented. The BOM$ directive may simulate a breakpoint (BPT) instruction or cause the printing of an error message before task termination. One of the following error messages is returned by the BOM$ directive:

%RSX$S_ERRLANGIN, Error in a high level language interface

%RSX$S_ERRLODOVR, Load overlay failure

%RSX$S_INVBOMCOD, Invalid BOM$ directive parameter

3.2.12 Named Directory Support

Software enhancements made to RSX-11M-PLUS Version 3.0 for directory formats of Files-11 Structure Level 1 devices are now supported by VAX-11 RSX and all its utilities except the Disk Save and Compress Utility (DSC1) and the File Transfer Utility (FLX).

Directories may have the following format:

[ufd]

where:

ufd

is 1 to 9 alphanumeric characters (A to Z, 1 to 9) as well as the UIC directory format ([g,m]).

3.3 VAX-11 RSX Software Restrictions

The following sections describe the software restrictions for VAX-11 RSX Version 2.0. Many of these restrictions are also documented in greater detail in the *VAX-11 RSX Compatibility Mode Reference Manual*.

3.3.1 Defining SYS$LOGIN

The system logical name SYS$SYLOGIN points to the systemwide login command file SYS$MANAGER:SYLOGIN.COM (unless specified otherwise). This file is invoked as part of the login procedure used by the DCL command interpreter (CLI). SYLOGIN.COM contains DCL commands that usually cannot be executed by Indirect (ICM). However, if you log in on the system using the Monitor Console Routine (MCR) command interpreter, you may wish to create a systemwide login command file that contains Indirect commands. You can do this by creating a SYS$MANAGER:SYLOGIN.CMD file and then defining the logical name SYS$LOGIN as follows:

```
$ DEFINE/SYSTEM/EXECUTIVE SYS$SYLOGIN SYS$MANAGER:SYLOGIN
```

By doing this, SYLOGIN.COM is invoked when you log in using DCL and SYLOGIN.CMD is invoked when you log in using MCR.

3.3.2 Differences Between VAX–11 RSX Memory Management Directives and RSX–11 Memory Management Directives

The following sections describe differences between VAX–11 RSX memory management directives and RSX–11 memory management directives.

3.3.2.1 Region Creation

VAX–11 RSX Version 2.0 static regions may be created in one of the following ways:

- By using the VAX/VMS Install Utility to make a compatibility mode image known to the system and by creating a global section (if the image has read-only sections or is a memory-resident library or common). For example:

```
$ INSTALL CREATE USER$:[MIKEE]RESLIB.EXE/SHARED
```

NOTE: **To create a static region with read-only access, enter the following command:**

```
$ INSTALL CREATE USER$:[MIKEE]RESLIB.EXE/NOWRITEABLE/SHARED
```

- By defining the name of a region as a logical name. This translates to the file specification of a compatibility mode image of a resident library or common. For example:

```
$ DEFINE RESLIB USER$:[MIKEE]RESCOM.EXE
```

VAX–11 RSX Version 2.0 dynamic regions can be defined by issuing the Create Region (CRRG$) directive in a compatibility mode image.

3.3.2.2 Region Deletion

VAX–11 RSX also allows you to delete a static region. To do this, use the VAX/VMS Install Utility (if the region was created by the Install Utility) by entering the following command:

```
$ INSTALL DELETE USER$:[MIKEE]RESLIB.EXE
```

To deassign the logical name for the static region, enter the following:

```
$ DEASSIGN RESCOM
```

3.3.2.3   Region Attachment

A compatibility mode task image can attach to any region created by VAX–11 RSX. It can also attach to any existent global section created by the VAX/VMS Install Utility or by a VAX/VMS native mode image, if the desired access is allowed. For regions created by VAX–11 RSX, the protection check is the same as it is for RSX–11 systems. For global sections created by the VAX/VMS Install Utility or native mode images, the check is the same as it is for VAX/VMS systems.

3.3.2.4   Displaying Region Information

The following list describes how to display information about RSX–11 regions.

- The VAX/VMS Install Utility can be used to display the names and the characteristics of the global sections that are associated to static or dynamic RSX–11 regions by entering either of the following:

  $ INSTALL LIST /GLOB

  or

  $ INSTALL LIST /GLOB/FULL

NOTE: For compatibility with RSX–11 systems, VAX–11 RSX creates regions that have a zero length (R.GSIZ=0 in the Region Definition Block (RDB)), but it does not asscociate a global section to it. Therefore, names of such regions cannot be displayed with the VAX/VMS Install Utility.

- The DCL command SHOW can be used to display the image files of resident libraries or commons that were defined as regions by using logical names. For example:

  $ SHOW LOG RESCOM

  RESCOM = USER$:[DIREC.LIBRARIES]RESCOM.EXE

  or

  $ SHOW LOG RES*

  RESLIB = USER$:[DIREC.LIBRARIES]RESLIB.EXE
  RESCOM = USER$:[DIREC.LIBRARIES]RESCOM.EXE

3.3.2.5   Region Checkpointing

VAX–11 RSX Version 2.0 does not support the Checkpoint Common Region (CPCR$) directive. However, region checkpointing does exist. The only difference is when checkpointing occurs, which cannot be controlled. VAX–11 RSX memory management creates a global section on behalf of the shared common region, using the compatibility mode image file as a section file. If the common region is accessed read/write and was built with checkpointing allowed (using the /CP switch or the /CHECKPOINT qualifier), the section file is updated each time pages from the common region are swapped out. At the end, when the common is detached and the region deleted, the section file contains all the modifications made in the common region.

NOTE: To make sure that a compatibility mode image file of a common region accessed for read/write operations is not being updated during the mapping session, build the common region as noncheckpointable. For a noncheckpointable common region file, VAX–11 RSX creates a Copy on Reference global section, and the pages are swapped into the system's page file rather than into the section file.

3.3.2.6   **Memory Management Granularity**

VAX–11 RSX memory management must comply with the VAX/VMS memory management. Thus, under VAX–11 RSX, a **page** is a set of 256 contiguous word locations which VAX/VMS uses as a unit for memory mapping and protection. The following requirements apply:

- Window alignment—Compatibility mode task images running on VAX/VMS and issuing memory management mapping directives must specify the window offset W.NOFF in the Window Definition Block as a multiple of 8 (256-word window alignment). If this requirement is not satisfied, VAX–11 RSX rejects the mapping attempt, returning an IE.ALG error code.

- /CM switch—Compatibility mode images of memory-resident overlaid libraries (that are accessed for read/write operations) must task build with the /CM switch or link with the /COMPATIBILTY qualifier. In this way, the Task Builder generates 256-word aligned windows for the memory-resident overlays. When VAX–11 RSX detects that a resident library does not satisfy this requirement, it displays an error message on SYS$ERROR and aborts the task. A task has read/write access to a memory-resident library or common when one of the following options was specified at task-build time:

  — RESCOM = name /RW

  — RESLIB = name /RW

  — COMMON = name :RW

  — LIBR = name :RW

  — CLSTR = name, name:RW

NOTE: The VAX–11 RSX Task Builder has the default set to /CM.

3.3.2.7   **Compatibility Between VAX–11 RSX Memory Management and RSX–11 Memory Management**

Tasks that use memory management and that are developed and tested on VAX–11 RSX can be transported to RSX–11, and conversely. Additionally, VAX–11 RSX provides better performance for shared regions accessed as read-only than it does for regions accessed for read/write operations. Therefore, tasks that do not write data in resident libraries or commons should access them as read-only, by specifying any one of the following TKB options while the task is being built:

- LIBR = library:RO

- RESLIB = library/RO

- CLSTR = library:RO

3.3.2.8   **VAX–11 RSX Memory Management Privilege Requirements**

To emulate the region attach/detach mechanism, the global sections created by the VAX–11 RSX memory management must be permanent. Therefore, the PRMGBL privilege is required for compatibility mode task images using memory management directives, or linking against memory-resident libraries or commons. To satisfy this requirement, RSX.EXE is installed at VAX–11 RSX startup time with the PRMGBL privilege.

3.3.3 Removal of Restrictions to Installing Compatibility Mode Tasks

To share read-only sections of tasks built with the /MU qualifier and installed with the VAX/VMS Install Utility, VAX–11 RSX Version 1.0 required that the task name be the same as the task file name. VAX–11 RSX Version 2.0 removes this restriction. Tasks built with the /MU qualifier can now have any name and still share read-only sections.

3.3.4 EXTK$ Directive Behavior Change

VAX–11 RSX does not change the current address mapping of a memory-resident overlaid task that issues an Extend Task (EXTK$) directive. However, if the task has no memory-resident overlays, VAX–11 RSX attempts to modify the mapping of the task to its region by the specified number of 32-word blocks.

NOTE: **Because VAX/VMS memory management acts on 256-word pages instead of on 32-word blocks, VAX–11 RSX will map an entire page when the task requests an extension equal to a fraction of a page. VAX–11 RSX also leaves the mapping unchanged if the task requests a contraction but still needs access to a fraction of the last page.**

The following restrictions apply to the Extend Task directive:

- A task cannot extend itself beyond 32k words.

- A task that has memory-resident overlays cannot reduce its size to less than the size of the highest window in the task region.

3.3.5 Changes Required to User-Created RSXUSR.EXE Images

The following information applies only to systems that have their own RSXUSR.EXE image if it was not changed for VAX–11 RSX Version 1.0. Note that if the image was changed, you do not have to do it again for VAX–11 RSX Version 2.0.

The VAX–11 RSX AME consists of the following four images:

- RSXSHR.EXE

- RSXUSR.EXE

- RSX.EXE

- BACKTRANS.EXE

The shareable images RSXSHR.EXE and RSXUSR.EXE constitute the actual AME, and the executable images RSX.EXE and BACKTRANS.EXE provide shell functions that are invoked by the current command interpreter (CLI) when VAX–11 RSX tasks are run.

On the previous version of VAX–11 RSX a change in the structure of the AME required changes to the RSXUSR.EXE shareable image. If you created your own RSXUSR.EXE image (and you have not already modified it for VAX–11 RSX Version 1.0), you must modify it before attempting to use it with VAX–11 RSX Version 2.0. Changes affecting RSXUSR.EXE are as follows:

- RSXUSR.EXE supports user-defined RSX–11 directives. When the AME encounters an RSX DIC code that it does not support, it dispatches

control to RSXUSR through the transfer vector RSX$UNKNOWN to allow emulation of the directive user-supplied code. The RSXUSR.EXE image supplied with VAX–11 RSX supplies the directive error status IE.SDP (DIC or DPB size invalid). You can substitute a user-written RSXUSR.EXE image for the supplied image by defining the logical name RSXUSR to the full file name of the desired replacement or by replacing the image SYS$SHARE:RSXUSR.EXE.

- Prior to VAX–11 RSX Version 1.0, all compatibility mode images were linked against base addresses. The base address for the RSXUSR.EXE shareable image was 21000 (16). This address placed size constraints on the RSXSHR.EXE shareable image. With the new software features added, the location of the RSXUSR.EXE image had to be modified. For Version 1.0 of VAX–11 RSX, RSXSHR.EXE and RSXUSR.EXE were converted to **unbased shareable images**, which means that the images are placed in the process virtual address space at image activation time (by VAX/VMS), regardless of how large either image may become in the future.

- To use your own RSXUSR.EXE shareable image with the VAX–11 RSX AME, you must modify the source code and the data to be Position-Independent Code (PIC). You must then link the object files as an unbased shareable image (by removing the "BASE=n" statement in the option file for the DCL command LINK).

- When converting your source code and data to PIC, be sure that your RSXUSR.EXE adheres to the following rules:

  — Any data references to addresses must use the MACRO-32 .ADDRESS directive rather than the .LONG directive. Address references using .LONG cannot be modified at run time by the image activator and, therefore, can prevent the shareable image from being relocated properly.

  — Any code references to addresses outside the RSXUSR.EXE shareable image must use the G^ (general) addressing mode. Doing so allows the image activator to modify the target address at run time. References to addresses within the RSXUSR.EXE image must use either G^ or relative (L^, W^, or B^) addressing mode. General addressing mode allows the image activator to modify the target addresses at run time. Relative addressing mode within a shareable image is PIC and works correctly wherever the image is located.

    In particular, be careful to make sure that the RSXUSR.EXE transfer vector uses relative addressing for the "jump" to the DIC Dispatch routine.

  — Do not use labels in the RSXUSR.EXE image as displacements, because the image activator will not be able to relocate the references properly. Use of labels is common when an offset into a static data structure must be calculated. However, the offset could be calculated in a register, so the data can be referenced using the data structure label as a displacement to the register. Change such code to use index mode rather than displacement mode.

  — Do not use immediate mode when referencing labels.

Note: For more detailed information, refer to the *VAX/VMS Utilities Reference Volume*.

3.3.6 Restrictions on Passing Device Names to Other Tasks

There are some applications that may pass information on devices (for example, SY or TI) to another task. For example, the MACRO–11 and TKB tasks pass a file specification to the CRF task when the /CR switch is supplied.

When a device name is passed to another task, that device name must be a name that is usable by the second task. In particular, the device name must be either a physical device name, or a logical name that is shared by both tasks. If both tasks are in the same application, the device name could be a jobwide logical name.

In VAX–11 RSX Version 2.0, RSX–11 tasks infrequently use physical device names. Because most VAX/VMS device names cannot be represented accurately in RSX–11 format, VAX–11 RSX returns a logical name if at all possible.

The logical names LB and SP are defined systemwide by the VAX–11 RSX startup command procedure and are accessible to all tasks on the system. If either name matches the VAX/VMS device name being used, VAX–11 RSX returns it to the RSX–11 task.

However, if the device name does not match either the LB or SP logical names, there is no guarantee of finding a representation that can be used by another task. Under these circumstances, VAX–11 RSX uses an algorithm that grants this control to the system manager and to the user. VAX–11 RSX will translate the $$n logical names to find one that matches the device. A $$n name may be defined in any logical name table that is included in the logical name search list "LNM$FILE_DEV" and, therefore, may be systemwide, groupwide, jobwide, or process-local.

If no $$n name matches the device, VAX–11 RSX may use internal logical names, such as $I or SY. These are not actual VAX/VMS logical names; they are understood only by the AME, which assigns them. Therefore, these names should not be passed to a different task. If passed to a native mode task, the names will simply be meaningless. If passed to another RSX–11 task, they will be meaningful, but probably will not mean the same thing that they mean to the task that passed them. For example, each task has a TI. However, each task will take TI to mean its own TI, even though the actual terminals may be different.

When the AME assigns a logical unit number (LUN) to SY, it will attempt to find a VAX/VMS logical name (such as LB or a $$n name) that refers to the same device. If the AME finds the VAX/VMS logical name, then the RSX–11 pseudodevice name SY will have the RSX–11 physical device name LB, $$0, or whichever name was found. If the process default disk is not LB or a $$n name, the physical device name for SY is SY0.

If a $$n name is not defined for the process terminal, the physical name for all terminal pseudodevices (TI, CL, CO) will be $I0. If a $$n name is defined, it can be passed to another task, provided that the $$n name is defined in a table to which both tasks have access. If SYS$INPUT, SYS$COMMAND, SYS$OUTPUT, or SYS$ERROR are actually files rather than a terminal device, $$n names cannot be used. (They cannot be shared with another task in any case.)

3.3.7 Restrictions for Indirect Using MCR on VAX–11 RSX

For Version 2.0, the Indirect Command Processor (Indirect) spawns a command line to MCR and that command line is executed by a subprocess with the MCR CLI. Command exit status and process-context information changed by the command are moved to the main process when the command completes in the subprocess. The following sections describe the restrictions imposed by this implementation.

3.3.7.1 Logical Name Handling

The following list describes restrictions to logical name handling:

- If you redefine SYS$OUTPUT in a command procedure, the new definition remains in effect only through the execution of the command procedure. When the command procedure exits, SYS$OUTPUT reverts to the original definition specified for your process.

- If you create process-local logical name tables in a command procedure, the tables are created in the MCR subprocess but do not exist in your main process when the command procedure exits.

- If, during a command procedure, you create logical names in a process-local table that you created, for example, a process-local table other than LNM$PROCESS, the logical names are not available after the command procedure completes. However, if you create logical names in the LNM$PROCESS table or any table that is not process-local, the logical names are available after the command procedure exits.

3.3.7.2 Process Context

The following list describes restrictions to the process context:

- Because MCR commands included in command procedures are actually executed in a subprocess, the output of a SHOW PROCESS command in a command procedure gives information on the subprocess. The process name, Process Identification Number (PID), and the deductible quotas are different from those of the main process in which Indirect is running. The remaining information looks the same.

- If you include the SET TERMINAL command in a command procedure, you must specify the terminal name, even if the terminal used is your own.

- In a command procedure, the SET PROCESS/NAME command changes only the name of the subprocess being used to execute the MCR commands within the command procedure. When the command procedure exits, your main process will have the same process name it had when you invoked the command procedure.

- If you include the following VAX/VMS-specific MCR commands from an indirect command procedure, changes in context are lost when the command procedure exits:

  — SET COMMAND

  — SET MESSAGE

  — SET RMS_DEFAULT

  — SET WORKING_SET

All MCR commands included in an indirect command procedure are executed within a subprocess. Most of the commands that alter process-context information move that information to the main process to make it available after the subprocess exits. The four commands shown do not do this.

NOTE: If you allocate a device during a command procedure and do not mount or deallocate the device during the procedure, the device is automatically deallocated when the command procedure exits.

3.3.8 Differences Between VAX–11 RSX Indirect and RSX–11M–PLUS Indirect

The VAX–11 RSX Indirect Command Processor (Indirect) does not support the following directives included in the RSX–11M–PLUS Version 3.0 Indirect Command Processor:

- .FORM

- .XQT

- .WAIT

The .FORM directive is not included because the FMS/RSX–11 driver is not supported under VAX–11 RSX. Complexities in correctly handling the asynchronous .XQT and .WAIT directives prevent their implementation in VAX–11 RSX Indirect for Version 2.0.

A few Indirect directives have minimal effect in the VAX/VMS environment and, as a result, they have been set to standard values that allow most command files to operate as expected. These directives include the following:

- .IFACT/.IFNACT

- .IFINS/.IFNINS

- .IFLOA/.IFNLOA

- .TESTPARTITION

The .TESTPARTITION directive can return information on RSX–11 regions, but partitions do not exist in the VAX/VMS environment.

The .TESTDEVICE directive returns the device-independent characteristics in the RSX–11 format, but the device-dependent characteristics it returns in words 2 and 3 of the device characteristics words are a VAX/VMS device-dependent longword.

A few of the special symbols are also slightly different due to the constraints and differences in the VAX/VMS environment. The following symbols, which can have varying values for RSX–11M–PLUS, have constant values for VAX–11 RSX:

- <BASLIN>

- <LOCAL>

- <PRIVIL>

- <FILER2>

- <MEMSIZ>
- <SYSTEM>
- <SYUNIT>
- <ACCOUN>
- <CLI>
- <FMASK>
- <SYSDEV>
- <SYDISK>

Refer to Section 5 of the *VAX–11 RSX Compatibility Mode Reference Manual* for more details on indirect command file processing.

3.4 Technical Changes to the VAX–11 RSX Documentation

This section describes technical changes affecting the VAX–11 RSX documentation.

In the previous version of the *VAX–11 RSX Installation Guide and Release Notes*, Section 3.2.11 displayed the following command lines inaccurately:

MCR:

```
> SET TERMINAL/NOLINE_EDITING
```

DCL:

```
$ SET TERMINAL/NOLINE_EDITING
```

4 RMS–11 Version 2.0 Release Notes

Sections 4.1 and 4.2 describe new or modified software features for and restrictions to RMS–11 Version 2.0.

4.1 Building RMS–11 Applications on VAX–11 RSX

The RMSLIB object library provided on VAX–11 RSX contains four versions of the R0EXEC programming module. The default version (R0EXEC) of this module is sufficient for most situations, however, you may wish to use another version depending on your requirements. A list of the modules and their features follows.

Module	Function
R0EXEC	Asynchronous/synchronous executive with RMS–11 software that allows execution of a task when the task is transported to an RSX–11 system.
R0EXSY	Synchronous executive with RMS–11 software that allows execution of a task when the task is transported to an RSX–11 system.
R0EXEV	VAX–11 RSX version only; Asynchronous/synchronous executive optimized for size (no RMS–11 software is included). If the task is transported to an RSX–11 system, R0EXEV will report ER$ENV (Feature not in selected RMS–11 environment) error messages on all RMS–11 operations except $WAIT.
R0EXSV	VAX–11 RSX version only; synchronous executive optimized for size (no RMS–11 software included). If the task is transported to an RSX–11 system, R0EXSV will report ER$ENV (Feature not in selected RMS–11 environment) error messages on all RMS–11 operations except $WAIT.

4.2 New Utilities Features

Enhancements have been made to the following RMS–11 utilities:

- File Definition Utility (RMSCNV)
- File Back-Up Utility (RMSBCK)
- File Restoration Utility (RMSDSP)
- Indexed File Load Utility (RMSRST)

The following two subsections describe the enhancements made to these utilities.

4.2.1   RMSCNV

The RMSCNV Utility supports the RMS–11 /ER switch. This switch allows RMSCNV to continue processing a file after an exception record is encountered. This switch uses the following format:

/ER:filename

If the /ER switch is not specified, RMSCNV will terminate immediately after encountering the first exception record. An error message indicating the type of exception record will be returned.

If the /ER:filename switch is specified, RMSCNV will continue processing the file. If exception records are encountered, the exception records will be placed in that file.

4.2.2   RMSBCK, RMSDSP, and RMSRST

The RMSBCK, RMSDSP, and RMSRST Utilities include enhancements to increase performance and decrease task size. The following switches have been added to the RMSBCK Utility:

RMSBCK Switch	Function
/NOQU	Disables query mode
/NV	Creates a new version of the output file
/CD:date:A	Interprets the creation date as after (:A) the specified date, in normal creation/revision date format
/CD:date:B	Interprets the creation date as before (:B) the specified date, in normal creation/revision date format
/RD:date:A	Interprets the revision date as after (:A) the specified date, in normal creation/revision date format
/RD:date:B	Interprets the revision date as before (:B) the specified date, in normal creation/revision date format

The following switches have been added to the RMSDSP Utility:

RMSDSP Switch	Function
/BR	Briefly displays file attributes
/SU	Supersedes existing output file

The following switches have been added to the RMSRST Utility:

RMSRST Switch	Function
/NOCV	Disables version radix conversion
/NV	Creates new version of output file

These utilities process file specifications that include the two wildcard characters, the asterisk (*) and percent sign (%). These two wildcard characters can be used in any position within the directory, file name, file type, and file version number fields.

4.3   RMS–11 Software Restrictions

This section describes software restrictions to RMS–11 Version 2.0. Suggestions for preventing or correcting problems are also included.

4.3.1   RMSDES

RMSDES has the following restrictions:

*   When the record size is greater than 1024 bytes, RMSDES selects area bucket sizes that are not large enough. Consequently, the file is successfully created, but is unusable. To bypass this restriction, specify the bucket size with the correct value before creating the file.

*   When the number of duplicates exceeds the available bucket space, RMSDES allocates more space than necessary. (See Chapter 6 of the *RSX–11M/M–PLUS RMS–11 User's Guide* for more information on calculating this number.) This can be corrected in either of two ways:

    *   Use the RMSDES SET command to override this allocated space.

    *   Reduce the number of duplicates to allow RMSDES to allocate space more conservatively.

*   RMSDES never prompts for the length of a packed decimal key field. A subsequent Create operation returns the error message "Invalid Key Field Specified for File." Therefore, if you select a packed decimal key you must explicitly set the segment length.

*   RMSDES uses the system device as the default device on a GET FILE DAT command. This default may be inconvenient if your login device is not the system device, or if you have set your default device to something other than the system device by issuing the DCL or MCR command SET DEF. To bypass this restriction, supply the device name in the file specification.

4.3.2   RMSIFL

RMSIFL has the following restrictions:

*   RMSIFL aborts with a memory management violation when more than one file is processed in the same RMSIFL session and the /DE switch is used. A temporary solution to this problem is to invoke RMSIFL for each file.

*   RMSIFL uses a sort algorithm that does not, when called upon to sort on a key, preserve the first in, first out (FIFO) ordering of duplicates. For alternate keys, this problem can be avoided by using RMSCNV. If your input file is an indexed file or a file sorted on primary key, you may use RMS IFL /NOSO provided that you wish only to order duplicates of the primary key.

4.3.3 RMSCNV and RMSIFL

RMSCNV and RMSIFL have the following restrictions:

- RMSCNV and RMSIFL ignore user-provided area extension quantities when loading a file. Instead, they use values that are large enough to reduce the number of file extensions.

4.4 Reporting RMS–11 Problems

Software Performance Reports (SPRs) allow you to report any software problems directly to DIGITAL.

The following additional information should be submitted with SPRs for RMS–11 software:

- Include the version number and patch level of the RMS–11 that you are using.

- Indicate whether you are using a programming language to process the file(s) and, if so, include the version number and patch level of the language.

- Include a Postmortem Dump and a map of the task involved, if RMS–11 aborts (or if some other task crashes and RMS–11 appears to be the cause).

- Include a copy or copies of the file(s) involved, in RMSBCK format.

- Include copies of the files that can cause the error, if the errors can be reproduced. If you cannot reproduce the problem, include a copy of the corrupt file, if possible.

- Include a description of the command line(s) or interactive session that led to the error (for RMS–11 utilities).

- Include a listing of the actual error, if possible.

A The Digital Equipment Computer Users Society (DECUS)

DECUS, the Digital Equipment Computer Users Society, is one of the largest and most active user groups in the computer industry. It is a nonprofit association, supported and administered by DIGITAL, but actively controlled by members. DECUS headquarters, located in Marlborough, Massachusetts, administers all international policies and activities. Members include individuals who have purchased, leased, ordered, or used a DIGITAL computer, or anyone who has an interest in DECUS. Membership is free and voluntary.

A.1 DECUS Goals

The primary goals of DECUS are as follow:

- To advance the art of computation through mutual education and exchange of ideas and information

- To establish standards and provide channels to facilitate the exchange of computer programs

- To provide feedback to DIGITAL regarding hardware and software customer needs

- To advance the effective use of DIGITAL computers, peripherals, and software by promoting the interchange of information

To further these goals, DECUS serves its members by holding symposia and maintaining a program library. An association newsletter, technical newsletters, and books are also published. DECUS includes support for a number of Special Interest Groups (SIGs).

A.2 DECUS Activities

Local, regional, and national DECUS organizations give members the opportunity to meet other DIGITAL customers and employees in an informal setting. From the monthly local meetings to the national symposia, members can discuss their ideas, learn what others are doing, and give DIGITAL valuable feedback for future product development.

Often, the national meetings in the various countries also provide the stage for major new product announcements as well as a showplace for developments in both hardware and software technology. At any meeting, members might describe ideas and programs they have implemented or fine-tuning that has been achieved for a particular application. Members present papers, participate in panel discussions, lead workshops, and conduct demonstrations for the benefit of other members.

Many of the technical papers and presentations from each symposium are published as a book, the *DECUS Proceedings*. Copies of the *DECUS Proceedings* are supplied to symposia attendees and can be purchased by DECUS members. DECUS also publishes newsletters focusing on special interests and a society newsletter.

DECUS has a Program Library, which contains over 1700 software packages that are written and submitted by users. A wide range of software is offered, including languages, editors, numerical functions, utilities, display routines, games, and other types of application software. Library catalogs are available that contain program descriptions and ordering information. The programs are available for a nominal service charge that covers the cost of reproduction and media.

Many DECUS members derive additional benefits from joining a DECUS Special Interest Group (SIG). SIGs often meet as subsets of regional and national meetings. They may also meet on their own to discuss their special field, such as operating systems, languages, processors, and applications. There are at least 25 SIGs in the United States alone. Many of the SIGs print newsletters and disseminate valuable technical information to members.

To obtain a membership form for DECUS, contact a DIGITAL sales representative or the nearest DECUS chapter office.

Index

Index

P

Passing device names • 3-15
PDP-11 Instruction Set Emulator • 3-1

R

Record Management Services (RMS-11) • 4-1 to
   4-4
   utilities
      BCK • 4-2
      CNV • 4-2, 4-4
      DES • 4-3
      DSP • 4-2
      IFL • 4-3
      RST • 4-2
Region
   attaching • 3-11
   checkpointing • 3-11
   creating • 3-10
   deleting • 3-10
   displaying information • 3-11
RSX-11 images • 3-7
RSXUSR.EXE images • 3-13 to 3-14

S

Save sets
   RSX020.A • 1-1
   RSX020.B • 1-1
   RSX020.C • 1-1
SYS$LOGIN • 3-10
SYSCOMMON system disks • 3-6

T

Terminal driver • 3-7

V

VAX/VMS utilities
   Authorize Utility • 1-2
   Backup Utility • 1-5
   Install Utility • 1-3
   Network Control Program (NCP) • 1-6
   System Generation Utility • 1-3

READER'S COMMENTS

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

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

_____
_____
_____
_____
_____
_____
_____
_____
_____
_____

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

_____
_____
_____
_____
_____
_____
_____
_____
_____
_____
_____

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

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

Name _____ Date _____

Organization _____

Street _____

City _____ State _____ Zip Code_____
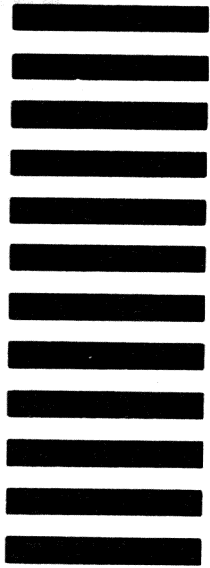   or Country