VAX-11 RSX Version 2.1 Release Notes

February 1986

First Printing, February 1986

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–3072

HOW TO ORDER ADDITIONAL DOCUMENTATION
DIRECT MAIL ORDERS

Contents

Contents

TABLES

Preface

Manual Objectives

The *VAX–11 RSX Version 2.1 Release Notes* provides information specific to the Version 2.1 release of VAX–11 RSX.

Intended Audience

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

Structure of This Manual

This manual contains three sections.

Section 1 describes new VAX–11 RSX software features.

Section 2 describes modifications to existing VAX–11 RSX software features.

Section 3 describes VAX–11 RSX Version 2.1 restrictions.

Associated Manuals

The following manuals provide more information on VAX–11 RSX installation and system operation procedures:

- *VAX–11 RSX Installation Guide and Release Notes*
- *VAX/VMS System Manager's Reference Manual*
- *VAX–11 RSX Compatibility Mode Reference Manual*

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.

Preface

Convention	Meaning
;	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 VAX/VMS DIGITAL Command Language (DCL) prompting character.
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   New Features for VAX–11 RSX

This section provides a brief description of all new features for VAX–11 RSX Version 2.1. These features include the following:

- VAX–11 RSX terminal driver enhancements
- RMS–11 memory-resident library support
- RMS–11 application building
- Privileged image continuation
- Data Terminal Emulator (DTE) suppport
- MCR mode messages

The *VAX–11 RSX Installation Guide and Release Notes* contains the step-by-step procedure for installing VAX–11 RSX on a VAX/VMS or MicroVMS system. Refer to that manual when you install and verify the VAX–11 RSX software on your VAX/VMS system. [1]

1.1   Terminal Driver Enhancements

The following subsections describe enhancements made to the VAX–11 RSX terminal driver, that allow the driver to operate like the RSX–11 terminal driver.

1.1.1   Terminal Characteristics

The VAX–11 RSX terminal driver has added support for the following terminal characteristics (listed in Table 1–1):

Table 1–1   Terminal Characteristics for SF.GMC and SF.SMC Requests

RSX–11 Bit Name	VAX/VMS Code	Meaning
TC.8BC	TT$M_EIGHTBIT	Pass eight bit on input
TC.HSY	TT$M_HOSTSYNC	Host to terminal synchronization
TC.NBR	TT$M_NOBRDCST	Disable broadcast
TC.PTH	TT2$M_PASTHRU	Pass through enable
TC.TSY	TT$M_TTSYNC	Output flow control

---

[1] The VAX–11 RSX Version 2.1 distribution kit includes the DTE and RMSRES task images, and the RMSRES symbol table file.

1.1.2 IO.ATA Function

The following is a list of extended capabilities added to the IO.ATA function:

- The IO.ATA function now supports the parameter2 value in the following format:

  ```
  QIO$C   IO.ATA   [!TF.NOT],....,<[ast], [parameter2], [ast2]>
  ```

  The value of the parameter2 is passed into the high byte of the stack at each unsolicited character AST. This feature can be used to identify terminals in a multiterminal environment.

- The IO.ATA function now supports the TF.NOT subfunction for the notification of unsolicited input. Unsolicited input causes an AST and entry into the AST service routine in the task. When the terminal driver receives unsolicited terminal input (except CTRL/C) and you used the TF.NOT subfunction with IO.ATA, the resulting AST serves only as notification of unsolicited terminal input; the terminal driver does not pass the character to the task. Upon entry to the AST service routine, the high byte of the first word on the stack identifies the terminal causing the AST (parameter2 in the IO.ATA function).

  Using the TF.NOT subfunction allows a task to monitor more than one terminal for unsolicited input without the need to read each terminal continuously for possible unsolicited input. Note that the TF.NOT subfunction cannot be used with the CTRL/C AST (ast2 in IO.ATA); an unsolicited CTRL/C character flushes the type-ahead buffer.

  If TF.NOT is specified, after the AST has been affected, the AST becomes "disarmed" until a read request is issued by the task. A read with timeout that specifies a timeout count of zero (TF.TMO and tmo=0) will move all unsolicited characters in the user's buffer. If TF.NOT is not specified with IO.ATA, every unsolicited character causes an AST.

  See the *RSX–11M/M–PLUS I/O Driver's Reference Manual* for more information on the TF.NOT subfunction.

1.2 RMS–11 Memory-Resident Library Support

VAX–11 RSX Version 2.1 supports RMS–11 memory-resident libraries. The RMS–11 memory-resident library configuration on VAX–11 RSX consists of the following two files:

- LB:[1,1]RMSRES.EXE

- LB:[1,1]RMSRES.STB

These two files are distributed with VAX–11 RSX Version 2.1 and they are the same files distributed with RSX–11M–PLUS Version 3.0.

On VAX–11 RSX, the RMS–11 memory-resident library files function only at task initialization time, to attach the required regions. They have no further role because all the RMS–11 functions during the task's execution are performed by the VAX–11 RSX RMS–11 to RMS–32 translator.

Tasks built on RSX–11 will not correctly transport to VAX–11 RSX unless they are built with the R0EXSY or R0EXEC modules. VAX–11 RSX supplies these modules in RMSLIB.OLB. They check whether the task is executing in the VAX–11 RSX environment and can interface with the system.

Follow these steps to replace the R0EXSY or R0EXEC modules in your RSX–11 RMSLIB.OLB:

1 Extract the R0EXEC and R0EXSY modules on VAX–11 RSX as follows:

```
>LBR ROEXECAME.OBJ=LB:[1,1]RMSLIB/EX:ROEXEC
>LBR ROEXSYAME.OBJ=LB:[1,1]RMSLIB/EX:ROEXSY
```

2 Replace the R0EXEC and R0EXSY modules on RSX–11 as follows:

```
>LBR LB:[1,1]RMSLIB/RP=[UIC]ROEXECAME
>LBR LB:[1,1]RMSLIB/RP/-EP=[UIC]ROEXSYAME
```

You must perform this procedure after each RSX–11 update.

You should also compare the module identifiers to determine if changes to the RSX module have been incorporated into the current release of VAX–11 RSX. Check your release notes for RSX–11 and VAX–11 RSX to make sure that the same modifications have been made.

To ensure that the RMS–11 memory-resident library files on VAX–11 RSX and RSX–11M–PLUS are identical, RMSRES is not built differently for compatibility mode. Therefore, to allow VAX–11 RSX to map the library, the RMSRES.EXE image must be installed as a shared, read-only image. The VAX–11 RSX startup command file, SYSMANAGER:VAX11RSX.COM, automatically installs the RMSRES.EXE image.

1.3 Building RMS–11 Applications

If you are building RMS–11 applications to run only on VAX–11 RSX, you can replace the RMS–11 overlay specification in the .ODL file(s) of your tasks with one of the following modules:

• R0EXEV—for asynchronous/synchronous RMS–11 operations

• R0EXSV—for synchronous RMS–11 operations

Usually, R0EXEV or R0EXSV can replace the entire RMS–11 overlay structure because most RMS–11 functions are performed in the VAX–11 RSX RMS–11 to RMS–32 translator. The following example illustrates how your RMS–11 overlay could be defined.

```
;
;     VAXRSXRMS.ODL VAX-11 RSX Version 2.1
;
;     ODL file to build an RMS-11 task to run only on VAX-11 RSX
;
      .NAME   RMS11
;
; For asynchronous/synchronous support
;
RMSROT: .FCTR  LB:[1,1]RMSLIB/LB:RMSSYM:ROEXEV
;
; For synchronous support
;
;RMSROT:.FCTR  LB:[1,1]RMSLIB/LB:RMSSYM:ROEXSV
;
RMSALL: .FCTR  RMS11
        .END
```

The following example shows an ODL file using an RMS–11 overlay defined by LB:[1,1]RMS12S.ODL.

```
;
; VAX-11 RSX Version 2.1
; ODL to build an RMS-11 task, using the RMS-11 overlay defined by
; LB:[1,1]RMS12S.ODL
;
        .NAME   NUME
        .ROOT   RAD
RAD:    .FCTR   NUME-RMSROT-F1-F2-F3,RMSALL
F1:     .FCTR   RMSQIOCL
F2:     .FCTR   GSA
F3:     .FCTR   LB:[1,1]VMLIB/LB:INIDM:EXTSK
;
;       Define the RMS-11 overlay
;
@LB:[1,1]RMS12S.ODL
        .END
```

The following example shows how to replace the RMS–11 overlay defined by LB:[1,1]RMS12S.ODL, with the R0EXEV module.

```
;
; VAX-11 RSX Version 2.1
; ODL to build an RMS-11 task, that runs only on VAX-11 RSX.
; The RMS-11 overlay defined by LB:[1,1]RMS12S.ODL is replaced
; with the ROEXEV module.
;
        .NAME   NUME
        .ROOT   RAD
RAD:    .FCTR   NUME-RMSROT-F1-F2-F3,RMSALL
F1:     .FCTR   RMSQIOCL
F2:     .FCTR   GSA
F3:     .FCTR   LB:[1,1]VMLIB/LB:INIDM:EXTSK
;
;       Define the RMS-11 overlay
;
        .NAME   RMS11
RMSROT: .FCTR   LB:[1,1]RMSLIB/LB:RMSSYM:ROEXEV
RMSALL: .FCTR   RMS11
        .END
```

Note: If you use the R0EXEV or R0EXSV module to replace the RMS–11 overlay in your task, you will not be able to transport your task to an RSX–11 system. However, using these modules will reduce the amount of memory the task requires by 6 to 12 Kb, depending on the size of the RMS–11 overlay.

1.4 Privileged Images

Because the AME is a privileged image, pressing CTRL/Y in previous versions of VAX–11 RSX terminated the execution of the compatibility mode task. Privileged images, running in a process that has MCR as the command interpreter, can now continue if you issue the CONTINUE or DEBUG command after a CTRL/Y is pressed.

1.5 Data Terminal Emulator (DTE) Support

VAX–11 RSX Version 2.1 includes support for RSX–11 data terminal emulation (DTE).

VAX–11 RSX DTE allows communication from your VAX/VMS system to a RSX–11M–PLUS or Micro/RSX Version 3.0 system. Using a terminal line connected from a system that will run DTE (the server system) to another system (the host system), a user can log in to the host system. You must have the SYSPRV privilege to invoke DTE, as follows:

```
>DTE device[/DIAL[="number"]]
```

where

DTE

invokes the DTE utility.

device

specifies the terminal line on the server system that is connected to a host system or a modem. If it is necessary to define a $$n logical name for the device, refer to the *VAX–11 RSX Compatibility Mode Reference Manual* for more information on $$n logical names.

/DIAL

allows the specification of a dial command string when the line is connected to a DF03, DF112, or DF224 modem. If you specify /DIAL without a number, DTE prompts for the number to be dialed. The string may consist of any of the digits 0 through 9 and the characters P, T, *, #, –, and a space. (Refer to the user guide for your modem for an explanation of these characters.) DTE specifies the terminating character.

DTE sets most of the necessary terminal attributes. However, before running DTE you can set attributes using the following command:

```
>SET TERMINAL/PERMANENT
```

The LOG_IO privilege is required to change attributes.

To terminate emulation, type the LOGOUT command from the host system and then, press CTRL/P. The following message is displayed:

```
%DTE-S-EMUEXIT, Emulation exiting... Please wait
```

You are then returned to command level execution on the server system. See the *RSX–11M/M–PLUS System Management Guide* for more information on using DTE and MFT.

Note: **When you use DTE on VAX–11 RSX, there are several differences not noted in the *RSX–11M/M–PLUS System Management Guide*. These include the following:**

- **DCL command syntax (as described in the *RSX–11M/M–PLUS System Management Guide*) is not applicable when using DTE on VAX–11 RSX.**

- **There is no recommended baud rate for running DTE on VAX–11 RSX.**

- **The DTE qualifier /SLAVE is not supported.**

1.6   MCR Mode Messages

When entering MCR mode, VAX–11 RSX Version 2.1 displays an informational message on your terminal. This message contains the name of the subprocess (spawned from DCL) that will be executing MCR commands. For example:

```
$ MCR
%RSX-S-MCRMODENTER, executing MCR commands in process "process name"
>
```

The subprocess name can be used to abort the execution of any current MCR command and delete the subprocess in which the MCR commands are executed. For example:

```
> CTRL/Y  INTERRUPT
> STOP "process name"
```

When you exit MCR mode, VAX–11 RSX Version 2.1 displays another informational message on your terminal. This message contains the name of the main process now in control of your terminal. For example:

```
> CTRL/Z
%RSX-S-MCRMODEXIT, control returned to process "process name"
$
```

2 Modifications to Existing VAX–11 RSX Features

This section describes modifications and corrections to existing features for VAX–11 RSX Version 2.1. These modified features concern the following:

- MCR
- System directives
- Device name handling
- File name handling
- VAX–11 RSX PDP–11 Emulator
- Indirect Command Processor
- RMS–11
- Task Builder
- Backup and Restore Utility (BRU)
- Undefined symbols in SYSLIB

2.1 MCR

The following subsections describe changes and corrections under MCR for VAX–11 RSX Version 2.1.

2.1.1 MCR Command Qualifiers

The /LOG—FILE qualifier for the SUBMIT command allows (but no longer requires) a file to be specified.

2.1.2 Command Execution

The following is a list of MCR mode problems corrected for VAX–11 RSX Version 2.1:

- On previous versions of VAX–11 RSX, an error during an MCR mode command execution (MCR mode was entered by typing MCR under DCL) occasionally aborted the AME, returning the following message:

  Improperly Handled Condition

  A brief stack and registers dump followed the message.

- When executed from Indirect or an MCR mode subprocess, the ATTACH command occasionally suspended terminal activity.

  On VAX–11 RSX Version 2.1, if you attempt to attach to the parent process of an Indirect or MCR mode subprocess, the terminal now displays the following error message:

  %MCR-E-REFUSED, you cannot attach to this process

- Previously, the SPAWN command could suspend terminal activity, when executed from Indirect or an MCR mode subprocess.

  On VAX–11 RSX Version 2.1, if you do not specify the /INPUT qualifier when executing the MCR SPAWN command from Indirect or MCR mode, the terminal displays the following error message:

  ```
  %MCR-E-SPWNINV, SPAWN not allowed from ICM process or special MCR mode
  ```

2.2   System Directives

The following subsections describe system directive operation problems that have been corrected in VAX–11 RSX Version 2.1.

2.2.1   GPRT$

Previously, the Get Partition Parameters (GPRT$) directive may have returned an incorrect value in the Directive Status Word (1 instead of 0) for regions other than the task region.

This problem has been corrected.

2.2.2   SPWN$

Previously, if a running task spawned another task using the SPWN$ directive that specified MCR and a command line (for example "RUN TSK2") and the spawned task tried to read from the terminal, the subprocess would suspend terminal activity.

2.2.3   ALUN$ or ACHN$

The following is a list of ALUN$ or ACHN$ directive problems corrected for VAX–11 RSX Version 2.1:

- An Assign LUN (ALUN$) directive or Assign Channel to Logical Name (ACHN$) directive failed and returned an error code (IE.LNL) when you assigned a device that had been used previously for RMS–11 operations.

- In a compatibility mode task, if two or more logical unit numbers (LUNs) were assigned to the same device (except SY and LB), reassigning the first LUN to another device caused the I/O operation directed to the other LUNs to fail. This correction removes the restriction noted in the customer letter for VAX–11 RSX Version 2.0.

- A LUN could not be reused for RMS–11 file operations after an RMS–11 $CLOSE because VAX–11 RSX returned an error code, even though the file had been successfully closed.

2.3   Device Name Handling

The following subsections describe changes in device name handling for VAX–11 RSX Version 2.1.

2.3.1   Reassigning LUNs

Previously, changes made to an RSX–11 device name while a LUN was assigned to that device may not have been recognized by the AME, even after reassigning the LUN. In particular, Indirect, when processing an indirect command file that modified the name of an RSX–11 device that had a LUN assigned, ignored the new name definitions and retained the oldest device name despite the LUN's reassignment.

This problem has been corrected, thus removing this restriction noted in the customer letter for VAX–11 RSX Version 2.0

2.3.2   Device Mnemonics and Physical Device Correspondence

Refer to Table 2–1 for changes made to RSX–11 device mnemonics and physical device correspondence. These are the mnemonics returned by the Get Device Information (GDVI$) directive for these devices.

Table 2–1   RSX–11 Device Mnemonics and Physical Device Correspondence

Media	RSX–11 Mnemonic	Physical Device
Tape	MU	TK50[1]
	MS	TS11, TU80[2]

[1]New device
[2]Changed devices

2.4   File Name Handling

The following subsections describe problems with file name handling corrected for VAX–11 RSX Version 2.1.

2.4.1   FCS .PARSE Routine Failure

A file name parse operation performed by the File Control Services (FCS) .PARSE routine may have failed and returned the IE.BDI error code, if both of the following conditions existed:

• The device was a branch of a search list.

• The directory string was not specified in the FCS dataset description.

This error occurred regardless of a valid directory identification specified in the FCS default file name block. This problem has been corrected.

2.4.2 FCS Create and Open Operations

A problem occurred with tasks built under VAX–11 RSX Version 2.0 that specified file names for FCS Create and Open operations without one of the following:

- A directory name string in the FCS dataset descriptor block

- A valid directory identification in the FCS default file name block

When the task was copied to an RSX–11 system, and then run, the files were created or opened in the directory [0,0] instead of in the user's default directory. This problem has been corrected.

2.4.3 FCS Created Files

Previously, any compatibility mode task that used FCS to create a new file received a version number of zero (0) in the file header. This problem has been corrected: files created by FCS now specify the correct version number in the file header.

2.5 VAX–11 RSX PDP–11 Emulator

When running on VAX processors that do not have compatibility mode hardware, an invalid parameter specification in the Directive Parameter Block (DPB) of a directive caused an access violation. The AME aborted with the following message:

```
Improperly Handled Condition
```

A brief stack and registers dump followed the message. This problem has been corrected.

2.6 Indirect Command Processor

Previously, the Indirect directive .PAUSE could only be used from the Indirect Command Processor (Indirect) while running in the main process. This meant that the following sequence would fail:

```
$ MCR
> @TI:
IND>.PAUSE
```

This problem has been corrected.

2.7   RMS–11

The following subsections describe corrections to RMS–11 for VAX–11 RSX Version 2.1.

2.7.1   RMS–11 Access Methods

The following list describes corrections to RMS–11 access methods:

- A problem has been corrected in the asynchronous record operation code. Previously, it was possible to have a record operation fail with the ER$RAB error code. This could occur in cases where an asynchronous operation was completing on one stream while another RMS operation was just beginning to be executed on another stream. The correction has been made in the R0EXEC and R0EXEV modules in RMSLIB.OLB. Therefore, if you perform asynchronous record operations with RMS–11, you must rebuild your application to incorporate this correction.

- During an RMS–11 $OPEN operation (or any similar operation), the device name was not returned in the O$DVI field of the RMS–11 NAM block when the LUN specified in the RMS–11 FAB block was used for a previous RMS–11 $OPEN operation.

2.7.2   RMS–11 Utilities

The following list describes problems with RMS–11 utilities that have been corrected in VAX–11 RSX Version 2.1.

- RMS–11 Indexed File Load Utility (RMSIFL)

  — The switch /DE failed to handle logicals correctly.

  — If an exception record failure was reported in the first command and the switch /NOER was specified, the next command failed with the RMS error code, ER$ISI. This occurred because SORT did not terminate correctly.

  — RMSIFL has been modified to check the file acess block (FAB) default extension quantity (DEQ) for an extension quantity if no explicit non-zero AREA XAB DEQ value has been provided. Thus, the user can set an explicit extension quantity without having to design areas. As before, if the user does not provide an extension quantity, RMSIFL uses approximately 50 blocks (aligned with bucket size).

  — A large number of alternate keys resulted in the RMS error code, ER$DME (dynamic memory exhausted).

  — RMSIFL would occasionally corrupt the alternate key root bucket. This occurred when the output file allowed duplicates on an alternate key, and the duplicate count field for a record straddled a block boundary. Therefore, once the file was loaded, any applications attempting alternate key access would fail with ER$CHK (check byte error in bucket). This problem has been corrected.

— When processing an indexed file with a large number of records, RMSIFL requires the use of temporary files during the sort phase. In some circumstances, RMSIFL will abort (while attempting to create the temporary files) and display the following error message:

```
?IFL -- SORTS error code in octal: 4
?IFL -- Fatal RMS error - STS = - 832, STV=0
```

Until RMSIFL upgrades to SORT–11 (V3.0), you should use the CONVERT Utility, which provides matching functionality and increased performance.

Note: **Make sure you create your output file with prologue 1 or 2 if the file is to be used on a PDP–11 system.**

- RMS–11 File Restoration Utility (RMSRST)

  — RMSRST occasionally failed with a privilege violation when the /FR switch was used to restore a magnetic tape container file. If the user did not have the privilege to create the output file with the protection and ownership of the original account from which the file was backed up, RMSRST would report a failure. In reality, the file was restored, but not with the correct protection codes.

  — The /SE: switch incorrectly required a complete file specification (including version number). If the file was not specified, RMSRST reported that the file was not found.

  — When the /SE: switch was specified, RMSRST did not terminate immediately when all files had been found.

  — RMSRST did not correctly handle magnetic tape files with decimal version numbers if the files were being restored to a system that supported decimal version numbers. The version numbers were erroneously converted to octal in the newly created output file.

  — In certain cases, RMSRST did not list the file version numbers as part of a summary listing.

- RMS–11 File Conversion Utility (RMSCNV)

  — RMSCNV would fail when an indirect command file specified /ER:filespec.

  — The switch /EO worked incorrectly. RMSCNV now null fills all blocks from logical EOF to physical EOF when the switch /EO is specified.

2.8   Task Builder

Prior to VAX–11 RSX Version 2.0, the Get Partition Parameters directive (GPRT$) always returned a successful response, consisting of a partition named GEN with a base address of 40000. If you specified a partition that didn't exist in the system at task-build, the base address and the length of the partition were not required.

Changes made to the GPRT$ directive on VAX–11 RSX Version 2.0 make it possible to get information about regions existing in the system. Therefore when you build a task and specify a partition, the Task Builder under VAX–11 RSX Version 2.0 and 2.1 performs as it does under RSX–11. That means if you do not specify the base address and the length, and the partition is not a known (existent) region, the Task Builder returns the following error message:

```
TKB -- *DIAG* - Invalid partition/common specified.
```

On VAX–11 RSX Version 2.0 and 2.1, a known partition or region is a valid image file of a resident library or a resident common. The name of that partition or region must be defined as a logical name that translates to the full image file specification, if the image file resides in a user directory. (The logical name is not required if the image file resides in LB:[1,1].) See the *VAX–11 RSX Compatibility Mode Reference Manual*, Version 2.0 for more details on defining RSX–11 regions.

2.9 Backup and Restore Utility (BRU)

The RSX–11 Backup and Restore Utility (BRU) on VAX–11 RSX can now create User File Directories (UFDs) on a mounted Files–11 Structure Level 1 output volume, when the /UFD qualifier is specified. This feature eliminates the need to manually create the necessary UFDs when copying files to a mounted output volume. For more information on using the /UFD qualifier, refer to the section of the *RSX–11M/M–PLUS Utilities Manual* that describes BRU.

Note: **The /UFD qualifier should NOT be used to copy files to a Files–11 Structure Level 2 output volume. If the /UFD qualifier is used, BRU will create UFDs with an incorrect format and will therefore be unable to copy files to these UFDs. If you use /UFD by mistake, you should delete the incorrectly formatted UFDs from the Master File Directory (MFD).**

2.10 Undefined symbols in SYSLIB

Under VAX–11 RSX Version 2.1 the A.JUMP and A.MODE symbols are defined in the following:

* FCSGBL in LB:[1,1]SYSLIB.OLB

* FSROF$ in LB:[1,1]RSXMAC.SML

3   VAX–11 RSX Version 2.1 Restrictions

This section describes VAX–11 RSX Version 2.1 restrictions. Areas that include restrictions are as follows:

- DCL
- Undocumented error codes
- Device name handling
- Using QIOs on RMS–11 assigned LUNs
- VMR
- PIP
- LOGIN.CMD
- RSX–11 Sysgens and Netgens
- PDP–11 DATATRIEVE/VAX

3.1   DCL

The following subsections describe DCL restrictions for VAX–11 RSX Version 2.1.

3.1.1   Restriction for HELP and EDIT

A native mode image that runs on and reads data from SYS$INPUT may not execute correctly if you enter MCR mode as follows:

```
$ MCR
>
```

Typing HELP or EDT in MCR mode causes the terminal to suspend activity. This behavior occurs because commands issued in MCR mode are passed to the executing subprocess through a mailbox (SYS$INPUT is a mailbox).

If you are using HELP, you can avoid this restriction. Invoke HELP with the /NOPROMPT qualifier by issuing the command as follows:

```
> HELP/NOPROMPT  subject
```

3.1.2 Aborting Indirect Command Procedures

Pressing CTRL/Y cannot stop the execution of a CLI command when the command is executed in a subprocess whose SYS$INPUT is a mailbox. This occurs in VAX–11 RSX when you perform either of the following:

- Run an indirect command procedure

- Enter a command in MCR mode

In either case, pressing CTRL/Y at the terminal will not stop the execution of the current command. It will, however, leave the subprocess in an indeterminate state and return control to the main process.

If you need to abort an indirect command procedure, it is recommended that you also delete the subprocess in which the commands are running. The name of the subprocess, MCR or MCR.n, will be displayed in an informational message after you enter MCR mode. To delete the subprocess, use the STOP command as follows:

```
$ MCR
%RSX-S-MCRMODENTER, executing MCR commands in process "MCR.1"
> CTRL/Y  INTERRUPT
> STOP MCR.1
```

3.1.3 Using SET TERMINAL and SHOW TERMINAL in an Indirect Command File

The MCR commands SET TERMINAL and SHOW TERMINAL default to SYS$COMMAND if a device name is not specified and return an error message if the device is not a terminal. Under VAX–11 RSX, in an indirect command file, SYS$COMMAND is a mailbox, therefore the commands SET TERMINAL or SHOW TERMINAL fail if a terminal device name is not specified.

If you did not override the default of SYS$OUTPUT by assigning it to a device or file name, you can use SYS$OUTPUT to define your terminal device name with the SET TERMINAL or SHOW TERMINAL command. The following example shows how you can use the SET TERMINAL command in a LOGIN.CMD command file.

```
.;
.;      LOGIN.CMD file
.;
SET TERMINAL/INQUIRE SYS$OUTPUT:
.;
.;      End of LOGIN.CMD
.;
```

3.2 Undocumented Error Codes

When executing an RSX–11 directive, such as SPWN$, some error codes returned by VAX/VMS system services to the AME cannot be translated to relevant RSX–11 error codes. Those error codes, such as LIB$_NOCLI, are simply translated to the RSX I/O error code, IE.VER (-4).

The Spawn directive may also return the RSX I/O error code, IE.NOD (-23), which is the translation of the VAX/VMS error code, SS$_EXQUOTA.

3.3 Device Name Handling

On VAX–11 RSX Version 2.1, logical names that are used as RSX–11 device names are restricted to logical names that translate to device names or device names and root directories. For example:

- Logical names that are valid RSX–11 device names:
  - $$0, which translates to 1DRA1:
  - $$0, which translates to DRA1:[ROOT.]

- Logical names that are not valid RSX–11 device names:
  - $$1, which translates to DRA1:[DIRECTORY]
  - $$1, which translates to DRA1:[DIRECTORY]FILE.NAME

As a consequence of this restriction, under VAX–11 RSX Version 2.1, the default definition of WK has changed from SYS$SCRATCH to SYS$DISK.

3.4 Using QIOs on RMS–11 Assigned LUNs

In the handling of LUNs, VAX–11 RSX does not allow QIOs to be issued to devices that are assigned for RMS–11 operations. VAX–11 RSX performs a checking procedure similar to that performed by the ALUN$ directive; if there is a file accessed by RMS–11 on that device, the directive aborts and returns the error code IE.LNL (similar to the error code returned by the ALUN$ directive performed on a LUN in use for a file operation).

Some of the solutions that you can apply are as follows:

- Do not use QIOs for the same LUNs that are assigned to RMS–11 accessed devices.

- Use only RMS–11 functions on LUNs that are assigned to RMS–11 devices.

The RSX–11 TI device is not subject to this restriction.

3.5 Invoking VMR

The VAX–11 RSX distribution kit does not include the Virtual Monitor Console Routine (VMR). However, you can build the VMR task on your system by using the VMR library from an RSX–11 system and the SYSLIB library from VAX–11 RSX. To build VMR on your system, follow these steps:

1 Copy [1,24]VMR.OLB, [1,24]VMRBLD.CMD, [1,24]VMRBLD.ODL, and [1,1]EXELIB.OLB from an RSX–11M or RSX–11M–PLUS system.

2 Edit the .ODL and .CMD files so that device, directory, and file names are specified correctly.

3 Delete (or comment out) the line "PAR=GEN:0:41100" in VMRBLD.CMD.

4 Link as follows:

```
$ MCR TKB @VMRBLD
```

Note: A VMR task that is only copied from an RSX–11 system may not correctly handle device and directory names.

3.6   Specifying Wildcard Characters with PIP

Under VAX–11 RSX, PIP allows you to execute commands that specify wildcard characters for directories on a Files–11 Structure Level 1 disk. If you specify wildcard characters for directories on a Files–11 Structure Level 2 disk, PIP displays an error message similar to the following:

```
$ MCR PIP [*,*]/LI
PIP -- Cannot find directory file
SYO:[*,*]
```

You can use native mode utilities for file operations that require specifying wildcard characters for directories on a Files–11 Structure Level 2 disk.

3.7   LOGIN.CMD

The maximum nesting level of indirect command files for VAX–11 RSX is four. Since LOGIN.CMD is invoked at login time from a temporary indirect command file created by MCR, the nesting level in LOGIN.CMD cannot exceed three.

Note: **A login command file is invoked after your right to access the system has been validated. Therefore, a failure in the execution of a login command file does not prevent you from continuing your terminal session.**

3.8   RSX–11 Sysgens and Netgens

RSX–11M/S and RSX–11M–PLUS system and network generations are only supported on Files–11 Structure Level 1 target or baseline disk volumes.

3.9   PDP–11 DATATRIEVE/VAX

The installation procedure for PDP–11 DATATRIEVE/VAX Version 3.1 is not compatible with VAX–11 RSX Version 2.1, although PDP–11 DATATRIEVE /VAX Version 3.1 will continue to run after VAX–11 RSX Version 2.1 is installed. However, if you attempt to reinstall PDP–11 DATATRIEVE/VAX Version 3.1, that installation will fail.

If you need to reinstall PDP–11 DATATRIEVE/VAX Version 3.1 with VAX–11 RSX Version 2.1, you should contact the Customer Support Center or your local office for information on how to perform the installation successfully.

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

digital

BUSINESS REPLY MAIL
FIRST CLASS PERMIT NO.33 MAYNARD MASS.

POSTAGE WILL BE PAID BY ADDRESSEE

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