RSX–11M–PLUS

Release Notes

AA–H427J–TC

February 1993

This revised manual contains technical changes made to the RSX–11M–PLUS operating system since the previous version, corrections to the documentation since Version 4.1, a summary of layered products supported on the operating system, and additional information on RMS–11.

Revision/Update Information:	This revised manual supersedes the *RSX–11M–PLUS Version 4.3 Release Notes* AA–H427H–TC
Operating System:	RSX–11M–PLUS Version 4.4

Digital Equipment Corporation
Maynard, Massachusetts

1993

The postpaid Reader's Comments form at the end of this document requests your critical evaluation to assist in preparing future documentation.

This document was prepared using VAX DOCUMENT, Version 2.1.

Contents

2 Corrections to Documentation

3 RMS–11 Version 2.0 Software Modifications, Restrictions, and Supplementary Information

Preface

Manual Objectives

The *RSX–11M–PLUS Release Notes* contains important information for using the RSX–11M–PLUS Version 4.4 operating system. Read this document before generating and using your system.

This manual contains the following categories of information:

- Descriptions of new software features, as well as changes to existing software features
- Descriptions of software and hardware restrictions
- Supplementary information, which appeared in previous release notes but is still applicable
- Corrections to documentation errors and omissions
- Software modifications, restrictions, and supplementary information for RMS–11 Version 2.0
- Correction files and restrictions for layered products

Intended Audience

This manual is intended for all users of the RSX–11M–PLUS Version 4.4 operating system.

Document Structure

Chapter 1 describes software features, software and hardware restrictions, and supplementary information for the RSX–11M–PLUS Version 4.4 operating system.

Chapter 2 corrects errors and omissions in the RSX–11M–PLUS documentation set.

Chapter 3 describes software modifications, restrictions, and supplementary information for RMS–11 Version 2.0.

Chapter 4 contains correction files and restrictions for layered products.

Appendix A describes the procedure for submitting a Software Performance Report (SPR).

Appendix B describes the procedure for applying corrections to source files.

Conventions

The following conventions are used in this manual:

Convention	Meaning
>	A right angle bracket is the default prompt for the Monitor Console Routine (MCR), which is one of the command interfaces used on RSX–11M–PLUS systems. All systems include MCR.
$	A dollar sign followed by a space is the default prompt of the Digital Command Language (DCL), which is one of the command interfaces used on RSX–11M–PLUS and Micro/RSX systems. Many systems include DCL.
xxx>	Three characters followed by a right angle bracket indicate the explicit prompt for a task, utility, or program on the system.
UPPERCASE	Uppercase letters in a command line indicate letters that must be entered as they are shown. For example, utility switches must always be entered as they are shown in format specifications.
command abbreviations	Where short forms of commands are allowed, the shortest form acceptable is represented by uppercase letters. The following example shows the minimum abbreviation allowed for the DCL command DIRECTORY: `$ DIR`
lowercase	Any command in lowercase must be substituted for. Usually the lowercase word identifies the kind of substitution expected, such as a filespec, which indicates that you should fill in a file specification. For example: filename.filetype;version This command indicates the values that comprise a file specification; values are substituted for each of these variables as appropriate.
/keyword, /qualifier, or /switch	A command element preceded by a slash (/) is an MCR keyword; a DCL qualifier; or a task, utility, or program switch. Keywords, qualifiers, and switches alter the action of the command they follow.
parameter	Required command fields are generally called parameters. The most common parameters are file specifications.
[option]	Brackets indicate optional entries in a command line or a file specification. If the brackets include syntactical elements, such as periods (.) or slashes (/), those elements are required for the field. If the field appears in lowercase, you are to substitute a valid command element if you include the field. Note that when an option is entered, the brackets are not included in the command line.

Convention	Meaning
[, . . .]	Brackets around a comma and horizontal ellipsis points indicate that you can use a series of optional elements separated by commas. For example, (argument[, . . .]) means that you can specify a series of optional arguments by enclosing the arguments in parentheses and by separating them with commas.
{ }	Braces indicate a choice among required options. You must choose one of the options listed.
:argument	Some parameters and qualifiers can be altered by the inclusion of arguments preceded by a colon. An argument can be either numerical (COPIES:3) or alphabetical (NAME:QIX). In DCL, the equal sign (=) can be substituted for the colon to introduce arguments. COPIES=3 and COPIES:3 are the same.
( )	Parentheses are used to enclose more than one argument in a command line. For example: `SET PROT = (S:RWED,O:RWED)`
,	Commas are used as separators for command line parameters and to indicate positional entries on a command line. Positional entries are those elements that must be in a certain place within the command line. Although you might omit elements that come before the desired element, the commas that separate them must still be included.
.	Periods immediately following a number indicate a decimal number. For example, 600. would mean 600_{10}.
[g,m] [directory]	The convention [g,m] signifies a User Identification Code (UIC). The g is a group number and the m is a member number. The UIC identifies a user and is used mainly for controlling access to files and privileged system functions. This may also signify a User File Directory (UFD), commonly called a directory. A directory is the location of files. Other notations for directories are: [ggg,mmm], [gggmmm], [ufd], [name], and [directory]. The convention [directory] signifies a directory. Most directories have 1- to 9-character names, but some are in the same [g,m] form as the UIC. Where a UIC, UFD, or directory is required, only one set of brackets is shown (for example, [g,m]). Where the UIC, UFD, or directory is optional, two sets of brackets are shown (for example, [[g,m]]).
. . .	Vertical ellipsis points show where elements of command input or statements in an example or figure have been omitted because they are irrelevant to the point being discussed.

Convention	Meaning
filespec	A full file specification includes device, directory, file name, file type, and version number, as shown in the following example: `DL2:[46,63]INDIRECT.TXT;3` Full file specifications are rarely needed. If you do not provide a version number, the operating system uses the highest numbered version. If you do not provide a directory, the system uses the default directory. Some system functions default to particular file types. Many commands accept a wildcard character (*) in place of the file name, file type, or version number. Some commands accept a file specification with a DECnet node name.
.	A period in a file specification separates the file name and file type. When the file type is not specified, the period may be omitted from the file specification.
;	A semicolon in a file specification separates the file type from the file version. If the version is not specified, the semicolon may be omitted from the file specification.
@	The at sign invokes an indirect command file. The at sign immediately precedes the file specification for the indirect command file, as follows: `@filename[.filetype;version]`
. . .	Horizontal ellipsis points indicate the following: • Additional, optional arguments in a statement have been omitted. • The preceding item or items can be repeated one or more times. • Additional parameters, values, or other information can be entered.
KEYNAME	This typeface denotes one of the keys on the terminal keyboard; for example, the Return key.
print and enter	As these words are used in the text, the system prints and the user enters.
Ctrl/x	The symbol Ctrl/x means that you are to press the key marked Ctrl while pressing another key. Thus, Ctrl/Z indicates that you are to press the Ctrl key and the Z key together in this fashion. Ctrl/Z is echoed on some terminals as ^Z. However, not all control characters echo.

1

New Features, Restrictions, and Supplementary Information

This chapter lists the new software features, describes new hardware support, describes the new software features, lists software problems corrected in this release, lists software and hardware restrictions, and provides supplementary information for the RSX–11M–PLUS Version 4.4 operating system.

1.1 Summary of New Features

The RSX–11M–PLUS Version 4.4 operating system supports the following new features:

- An additional version of the Task Builder (TKB), known as Hybrid Fast Task Builder (HFT). HFT uses a fast-mapped VSECT region for virtual memory table storage. This version of TKB will only be available on I&D systems.
- Support for the year 2000 and beyond.
- Support for the RA71 and RA92 devices.
- Support for the RQZX1 SCSI adaptor.
- KDJ11–E Processor (used in Micro PDP–11/93 and PDP–11/94).
- Additional documentation for the GI.DVJ and GI.VEC subfunctions of the GIN$ system directive.
- New high-level language calls added to the GIN directives.
- Support for QMG after 1992.
- Enhancements to the PRINT and SUBMIT commands.
- Two new Send Data and UNstop directives.
- Support for TDX line continuation.
- Support for Universal Receiver Task (URT).
- An enhancement to loadable XDT.

1.2 New Hardware Support in Version 4.4

The following new devices are supported:

- RA71
- RA72
- RA92
- RQZX1 SCSI adaptor:
  - RZ23L

- RZ24L
- TZ30

Caution

ANSI tape processing for information interchange is not supported on the TZ30 cartridge tape drive.

- KDJ11–E Processor (used in MicroPDP–11/93 and PDP–11/94)

1.2.1 Support for the KDJ11–E Processor

Support for the KDJ11–E processor module has been added to Version 4.4. The following features are supported:

- The module ID
- The TOY clock (Time of Year clock)
- On-board serial lines (as terminal driver option)

Support has been added in SYSGEN, ACF, Error Logging, SAVE, the Executive, and the MCR SET TIME command for this processor.

1.2.2 RA71, RA72, and RA92 Disk Support

Error logging support has been added for the RA71, RA72, and RA92 disk devices.

1.2.3 Supplementary Hardware Information

For a complete list of hardware supported by Version 4.4, refer to the *Software Product Description* contained in your media kit.

1.3 Software Enhancements for Version 4.4

This section describes software enhancements added to Version 4.4.

1.3.1 Support for QMG after 1992

Previous versions of the Queue Manager and spooling tasks (QMGCLI) provided 4 bits for storing the year specification in a SUBMIT/AFTER or PRINT/AFTER job. Also, the data structures passed between the Queue Manager and QMGCLI that were stored in queue file QUEUE.SYS provided 4 bits for storing year specification in SUBMIT/AFTER or PRINT/AFTER jobs. The information has been stored as an increment from 1977.

Now, the year specification is stored as an 8-bit value, as an increment from 1900. Along with this change, the storage of month, day, hour, and minute have been changed.

Prior to starting the new Queue Manager, the system manager should assure that the old QUEUE.SYS file is empty, that is, the results of a SHOW/QUE instruction shows all queues empty of held, pending, stopped, paused, etc. jobs. A new QUEUE.SYS file will be created by the Queue Manager when it starts.

The required input format for the date will be consistent with the standard format for years following the year 2000; for example:

```
Input    Meaning
-----    -------
nn       19nn
19nn     19nn
20nn     20nn
```

Further, inputs of only one digit or of three digits will be rejected as errors. RSX assumes that /AFTER specifications on the submitted command line are 1977 or after; for example:

```
/AFTER:21-JUN-93 refers to 1993
/AFTER:21-JUN-04 refers to 2004
/AFTER 21-JUN-76 refers to 2076
```

Although the spooler tasks included with this version have been modified in accordance with the new formats, application programs may include spoolers that have not been updated. The former syntax for packets from a spooler task to the Queue Manager remains acceptable, although certain assumptions are made.

If a packet is received with function QM.OPJ (former Open Job function) or QM.MDJ (former Modify Job function), the Queue Manager modifies the packet to the new format before storing it in QUEUE.SYS. The former 4-bit Year increment value is assumed to be a modulo-16 delta from 1977, specifying a date subsequent to current system time. This is implemented by adding 16 to the increment until the year is greater than or equal to that of the current system time.

Any tasks that directly access the Job Entry structure in QUEUE.SYS will have to be modified accordingly. Symbolic offset names are maintained so, unless the task accesses the After Time or Job Time Stamp fields, they probably only require you to assemble and link them again. Tasks that do make use of the Time fields should be recoded for consistency with the new format.

1.3.1.1 Queue Manager Changes Affecting DECnet–11M–PLUS and Micro/RSX DECnet

Due to the changes to the Queue Manager, if your system utilizes the DECnet File Transfer Server (FTS), this component will need to be rebuilt during your Network Generation. If your system is not a pregenerated RL02 system or Micro/RSX system, this can also be done by using a "Component mode NETGEN." If you system is either a pregenerated RL02 system or Micro/RSX, you should copy the following files to your system's <NETUIC>:

- [DECNET]FTSRES.TSK
- [DECNET]FTSFSL.TSK.

1.3.2 New Version of Task Builder

This version of RSX–11M–PLUS provides an additional version of TKB, known as Hybrid Fast Task Builder (HFT). HFT uses a fast-mapped VSECT region for virtual memory table storage. This version of TKB will only be available on I&D systems.

Currently, TKB uses a disk workfile for virtual memory table storage of symbols. For tasks with a large number of symbols, link time is greatly increased because of the amount of disk I/O required for swapping virtual memory pages in and out of dynamic memory (the free space between the end of TKB's code and its highest possible virtual address). Using a fast-mapped region will improve performance by eliminating this disk I/O. Performance improvements will vary depending on the task's configuration. As an example of possible improvements, test builds of PDP–11 C and PDP–11 BP2 done on a standalone Micro/RSX system using HFT build 35% to 43% faster than with TKB on the same system. Please note

however, that for smaller tasks that do minimal amounts of paging, there will be little or no benefit.

HFT has been optimized to provide faster linking by using a fast-mapped VSECT region in place of the disk workfile used by TKB, TKBRES, and TKBFSL. Task images produced by HFT are identical to ones produced by TKB, TKBRES, and TKBFSL.

HFT can be installed with a default task name of (. . . HFT). The user may also find it advantageous to install HFT as . . . TKB instead of the standard TKB task image; however, the task image is considerably larger. You can install one, the other, or both. Digital recommends that you not install both tasks on the system, as the HFT requires a great deal of memory.

A task image for HFT built with the standard TKB defaults can be found in LB:[3,54]. If nonstandard defaults are desired, the build command file (HFTRESBLD.CMD) and .ODL file (HFTRESBLD.ODL) can be found in LB:[1,24], and HFT can be rebuilt using the section of SYSGEN which permits rebuilding of supplied system tasks (Building Nonprivileged Tasks). However, please note that during this section of sysgen, HFT will not automatically be rebuilt if you specify ALL, ALLRES, or ALLFSL due to the way HFT is configured. You must specify HFTRES when sysgen prompts for the task to be rebuilt in sysgen question BN020 in order to rebuild HFT.

All references to and use of the disk workfile have been removed in HFT. Initialization has been changed to immediately extend dynamic memory to D-APR 7 (instead of being extended 3100 bytes at a time on demand as TKB executes). The new HFT has been built with I&D, with D-APRs 0-2 mapped to D-space used by the task code and D-APRs 3-6 mapped to dynamic memory. D-APR 7 is used to fast map to the VSECT region. (Note that TKB, TKBRES, and TKBFSL also provide only 4 APRs worth of dynamic memory, so there is no reduction in dynamic memory with this implementation.)

The fast-mapped VSECT region will be the same size as the maximum disk workfile (64KW) and pages will be located in the VSECT region in a fashion similar to that used with the disk workfile. Virtual memory pages will be allocated sequentially, with each virtual memory page 256 words long. A page will be located by means of its page block number (the high byte of its address). When a page is read in or out of virtual memory, its page block number will be used to determine the following:

- Which APR in the VSECT region needs to be fast mapped
- The correct offset into the APR to locate the page

After fast mapping the appropriate APR, a subroutine copies the 256 words of the page into (or out of) the VSECT region to (or from) the designated location in virtual memory.

As a further optimization, FCSFSL is used as a resident library to avoid the overhead from context switching.

Modified versions of the appropriate TKB modules to support the new HFT have been added to [31,10] and carry an *X* suffix (for example, there will now be an INIDMX used solely by the new HFT, in addition to the original INIDM used by TKB). The TKB assembly command file(s) have been updated.

1.3.3 Support for Year 2000 in Date Format

The RSX operating systems have been changed to support dates beyond the year 2000. Generally, RSX treats the current year as a two-digit field. The following areas of the RSX system have been changed:

- Input—All areas of the system that allow input of a date will now accept two different forms of date input. First, the traditional dates of the dd-mmm-yy format (or mm/dd/yy for TIM) will still be accepted. These dates will always be considered as occurring between the years 1900 and 1999. In addition, the date may now be specified as dd-mmm-yyyy or (mm/dd/yyyy for TIM only), to allow the system to recognize dates occurring after 1999.
- Output—In order to preserve compatibility in applications that parse the output of system commands, the format of date output will not be changed. All places that output the date as 2 digits will continue to do so, representing 2000 as 00, and so forth. Places that currently output the date as 4 digits will still output the date as 4 digits.
- Internal representations—As documented for the GTIM$ directive, the year field of time vectors is the number of years since 1900. This is still accurate. This field will reach 100(10) in the year 2000.

  For example, Table 1–1 shows how dates between years 1900 and 2209 will be stored. Dates will be displayed as the lower two digits of the real year or as a four-digit year.

  However, ODS–I only provides a 7-byte ASCII field for creation and other dates. In order to accommodate this restriction in a way that is consistent with RSX date handling, the 2-bytes of year in the date attribute will be used to represent the year since 1900.

  The two bytes will be encoded as follows:

  - Low byte—Low order decimal digit of year, in ASCII.
  - High byte—Quotient of years since 1900 divided by 10, plus 60_8. This represents the ASCII high digit of the year from 1900 – 1999. In 2000, however, this will result in ":" being stored for the high digit of the year. For years 2010 through 2019, ";" will be stored, and so forth. This change should result in another 200 years or more of file representations.

Table 1–1 Storage of Dates in RSX Systems after 1990

Calendar Year	Year Field Representation
1990 – 1991	90 – 99
2000 – 2009	:0 – :9
2010 – 2019	;0 – ;9
2020 – 2029	<0 – <9
2030 – 2039	=0 – =9
2040 – 2049	>0 – >9
2050 – 2059	?0 – ?9
2060 – 2069	@0 – @9

(continued on next page)

Table 1–1 (Cont.) Storage of Dates in RSX Systems after 1990

Calendar Year	Year Field Representation
2070 – 2079	A0 – A9
2080 – 2089	B0 – B9
2090 – 2099	C0 – C9
2100 – 2109	D0 – D9
2110 – 2119	E0 – E9
2120 – 2129	F0 – F9
2130 – 2139	G0 – G9
2140 – 2149	H0 – H9
2150 – 2159	I0 – I9
2160 – 2169	J0 – J9
2170 – 2179	K0 – K9
2180 – 2189	M0 – M9
2190 – 2199	N0 – N9
2200 – 2209	O0 – O9

The following features have been changed to accept or display the new date format:

- BROADCAST utility
- BRU utility
- The following DCL commands:
  - SET TIME
  - DIR/SINCE
  - BACKUP
- F11ACP
- MACRO
- PIP utility
- RPT and the /DATE qualifier
- QMG
- MCR command line interface
- The following MCR commands:
  - BYE command
  - HEL command
  - TIM command

1.3.3.1 Changes to the File Exchange Utility (FLX) for Year 2000 and Beyond

FLX has been modified to correctly record and display the dates of RT–11 files created after the year 2003. A FLX directory listing of RT–11 files will now display the century as part of the year.

The FLX utility correctly displays dates from 1970 to 2001 with the /DO switch set. A *C* is displayed when the date is not in the valid range. Additionally, the dates displayed are invalid. However, the files are intact and correct. FLX will display the date as:

```
ACF.TSK    33.C   15-NOV-98
```

The following example shows how FLX will display a file with invalid date information if the year is greater than 2001 (The <233> is the protection code.):

```
ACF.TSK         33.C    19-70 <233>
```

1.3.4 Additional Documentation for the GIN$ Subfunctions

Please add the following descriptions of the GI.DVJ and the GI.VEC subfunctions of the GIN$ system directive to the *RSX–11M–PLUS and Micro/RSX Executive Reference Manual.*

Note

This description supersedes the description of the GI.DVJ in the Version 4.3 Release Notes. The macro expansion shown in Section 2.8 was incorrect in those release notes.

1.3.4.1 GI.DVJ – Son of Get Device Information

The Son of Get Device Information function returns information about the mount status and accessibility of a particular mountable mass storage device. The device on which information is returned is determined by first performing a logical assignment (if required) and then following any redirects. Device assignments are checked if the high bit in the flags byte is clear; otherwise, no check of device assignments is made.

Macro Call

```
GIN$    GI.DVJ,buf,siz,dev,unt
```

Parameters

GI.DVJ
GIN$ function code (18.)

buf
Address of the buffer to receive the unit information.

siz
Buffer size in words.

dev
Device name. (If zero, use task's TI.)

unt
Device unit number. (If high bit clear, follow assignments.)

Table 1–2 GI.DVJ – Son of Get Device Information Subfunction Buffer Format

Word 0	Device Mount Status
	If word 0 equals – 1, then the device is either not mountable or not a mass storage device. If word 0 equals – 2, then the device is not accessible to the issuing task. If word 0 is greater than zero, then the individual bits have the following meanings: Bit 0 – The issuer is not allowed read access to the device. Bit 1 – The issuer is not allowed write access to the device. Bit 2 – The issuer is not allowed extend access to the device. Bit 3 – The issuer is not allowed delete access to the device. Bit 4 – The device is public. Bit 5 – The device is private (allocated). Bit 6 – The device is mounted foreign. Bit 7 – The foreign mounted device has an ACP.

Macro Expansion

```
GIN$    GI.DVJ, DVBUF, DVSIZ, "DU, 0
.BYTE   169.,18.
.WORD   GI.DVJ
.WORD   DVBUF
.WORD   DVSIZ
.WORD   "DU
.WORD   0
```

Table 1–3 GI.DVJ – Son of Get Device Information Subfunction DSW Return Codes and Their Meaning

Return Code	Meaning
IS.SUC	Successful completion.
IE.ADP	Part of DPB is out of task's address space.
IE.IDU	The specified device does not exist, or device is a virtual terminal and issuing task is not parent or offspring.
IE.SDP	Invalid subfunction code or the DPB size is invalid.

Note

If the task has the *slave* attribute, logical assignments are not checked, regardless of the setting of the high bit in the fourth parameter word (the device unit number parameter).

1.3.4.2 GI.VEC – Translate Executive Entry Point Vector

The translate executive entry point vector subfunction may be used by privileged tasks to provide independence from a particular system's symbol table file.

Macro Call

```
GIN$ GI.VEC,buf,siz
```

Parameters

GI.VEC

GIN$ function code (15.)

buf

Address of the buffer that contains the offset codes and will contain the returned entry point addresses.

siz

Buffer size in words.

Table 1–4 GI.VEC – Translate Executive Entry Point Vector Subfunction Buffer Format

Word 0	Flag word If word 0 equals 0, the translation is performed. If word 0 is greater than zero, then the directive completes with status IS.SUC (success), but no translation is performed. This prevents successive translates on the same buffer from resulting in system faults. Word 0 is set to 1 when the translation is performed.
Word 1 – n Offsets	Each word beyond word 0 in the buffer is an offset into a table. Each of these offsets represents a symbol in the executive. The file LB:[3,54]RSXVEC.STB contains these symbols. The names of these symbols are identical to the represented entry point. For example, RSXVEC.STB might define $TSKHD to a value of 1236. The real value of $TSKHD for that system is at offset 1236 in the executive's vectored entry point table. For example, a value of 20646. The GI.VEC subfunction will replace the 1236 in the buffer with the corresponding real symbol value, 20646. If the word in the buffer does not represent a valid offset, or if it represents a symbol that is not included due to SYSGEN options on the target system, the value 160001 is returned for that value. No error is returned in this case. Applications that employ vectoring should test for the existence of optional features before using any entry points associated with those features. For more information on vectoring, see the description of using vectored executive entry points in the *RSX–11M–PLUS and Micro/RSX Guide to Writing an I/O Driver*.

Macro Expansion

```
GIN$    GI.VEC, EXEBUF, EXESIZ
.BYTE   169.,18.
.WORD   GI.VEC
.WORD   EXEBUF
.WORD   EXESIZ
```

Table 1–5 GI.VEC – Translate Executive Entry Point Vector Subfunction DSW Return Codes and Their Meaning

Return Code	Meaning
IS.SUC	Successful completion.
IE.ADP	Part of DPB is out of task's address space.
IE.SDP	Invalid subfunction code or the DPB size is invalid.

1.3.5 High-Level Language for the GIN Directive

The following new high-level language calls to the GIN directives may be utilized:

- CALL GINABO
- CALL GINAPR
- CALL GINDEF
- CALL GINDEV
- CALL GINDVJ
- CALL GINFMK
- CALL GINGAS
- CALL GINQMC
- CALL GINSPR
- CALL GINTSK
- CALL GINUAB
- CALL GINUIC
- CALL GINUPD
- CALL GINVEC

See Section 2.9 for more details on these calls.

1.3.6 Enhancements to the PRINT and SUBMIT Commands

The PRINT and SUBMIT commands have been enhanced to have a mnemonic queue name translated to a processor name without requiring definition and assignment for the mnemonic queue name.

Previously, the Queue Manager restricted the system to a total of 16 print queues. By providing the capability to translate the PRINT command mnemonic queue name to a processor name, you can avoid the need to define and assign the second queue for each printer, thereby effectively doubling the capacity of the Queue Manager. The device-specific queue must still be created. As a side benefit, you also can define a logical name (for instance, in your Session Table) and avoid the need to remember which /FORM=n to specify when printing to a terminal-like

device. Previously, you might have issued a command similar to the following to print 80 columns, letter quality on an LN03:

```
PRINT/FORM=4 resume.txt
```

Now, you can define a queue similar to the following and then issue the next command:

```
DEF QMG$QUE_NICE "TT20/FORM=4"
PRINT/QUE=NICE resume.txt
```

Rather than requiring the system manager define a mnemonic queue and assign it to the processor-related queue, the ASSIGN/QUE command will now define a logical name in the System Table to describe the relationship, if the queue being assigned has not previously been created. Subsequent use of the PRINT /QUE=name command attempts to translate the queue name to a processor name, that is then substituted for the original queue name. The translation repeats until a failure occurs, at which time the last translated name is submitted to the Queue Manager.

1.3.6.1 ASSIGN/QUEUE (QUE /AS: in MCR) Command

DCL FORMAT: $ ASSIGN/QUEUE que_name proc_name[/qualifiers]

Description All DCL command qualifiers valid for the PRINT command may be used.

MCR FORMAT: > QUE proc_name:[/qualifiers]/AS:que_name

Description All MCR Job Switches valid for the PRINT command may be used.

Note

If your system does not have logical name support installed, QMG cannot use logical names.

If que_name has previously been defined to the Queue Manager, the command exhibits prior behavior; for example, a path is established from the queue to the processor. In this case, any qualifiers are ignored and a warning message is issued to the user. Note that queue names PRINT and BATCH are always defined in the default system.

If que_name has not been defined to the Queue Manager, a special logical name is defined in the System Table to represent the translation from que_name to proc_name, including qualifiers. This is equivalent to issuing the following DCL command:

```
DEF QMG$QUE_que_name "proc_name[/qualifiers]"
```

1.3.6.2 PRINT (PRI in MCR) Command

DCL FORMAT: $ PRINT/QUEUE:que_name [/qualifiers] filespec[/filequalifiers]

Description All DCL command qualifiers valid for the PRINT command may be used.

MCR FORMAT: > PRI que_name:[jobname][/jobswitch]=filespec[/filequalifiers]

Description All MCR Job Switches valid for the PRINT command may be used.

Note

Note, however, that any qualifiers that were included in the ASSIGN /QUEUE (or QUE /AS:) command must not be repeated here. The resulting multiple qualifiers after que_name translation will cause an error return from the Queue Manager.

Execution of the PRINT command first iteratively translates que_name. At each step, the original que_name is replaced by the translation, including qualifiers (if any), and the new que_name is extracted and submitted for further translation. When there are no more translations, the last constructed string is submitted to the Queue Manager for execution.

1.3.6.3 DEASSIGN/QUEUE (QUE /DEA in MCR) Command

DCL FORMAT: $ DEASSIGN/QUEUE que_name proc_name

MCR FORMAT: > QUE proc_name:/DEA:que_name

If que_name has previously been defined to the Queue Manager, the command exhibits prior behavior; for example, the path from the queue to the processor is severed.

If que_name has not been defined to the Queue Manager, the special logical name representing the que_name to proc_name translation is deleted from the System Table. This is equivalent to issuing the following DCL command:

```
DEASS/SYSTEM QMG$QUE_que_name
```

The following commands are unchanged from their prior behavior:

INITIALIZE/QUEUE	(QUE /CR)
DELETE/QUEUE	(QUE /DEL)
STOP/QUEUE	(QUE /STO)
START/QUEUE	(QUE /STA)

Note

Note, however, that they will affect only queues that *exist* in the Queue Manager because they were created by the INITIALIZE/QUEUE command (or by default in the case of PRINT and BATCH queues); they will not affect mnemonic queues that are mapped to processors by means of logical names.

1.3.7 New Directives

Two new directives have been added to the system. These are Send Data and UNstop (SDUN$), and Variable Send and UNstop (VSUN$). Additionally, two new high-level language interfaces (SDUN and VSUN) have been added. These directives, and their functional descriptions, are identical to SDAT/SEND and VSRC, with two exceptions:

- The DIC for this directive is 179.

- The directives may return a status of IS.CLR(0) or IS.SET (+2). A status of IS.SET indicates the send was successful but the target task was not active. A status of IS.CLR indicates the send was successful and the target task was active but not stopped.

1.3.8 Support for TDX Line Continuation

The TDX utility has been enhanced to accept continuation lines and to pass command lines greater than 80 characters to spawned utilities. TDX will now perform multiple GMCR directives if the first command line ends with a continuation mark ("-"). All segments of the command are assembled before the parsing routine is called. Additionally, TDX will now spawn external tasks with command lines that exceed 80 characters. Commands that exceed 132 characters and are processed by command files will now receive the "AT.tnn – Syntax Error" error message instead of the "MCR – Line too long" message. Commands that greatly exceed 132 characters from DCL terminals will receive the "MCR – Iform buffer full" error message.

1.3.9 Support for Universal Receiver Task

A Universal Receiver Task has been added to RSX. The Universal Receiver Task receives all SDAT type packets that received IE.INS in previous versions of RSX–11M–PLUS. This task then routes the packets to the appropriate tasks and/or systems.

If a task is installed with the name URT . . . , it is considered to be a Universal Receiver Task for all variants of the Send Data (SDAT) executive directive. SDAT$, VSDA$, SDRC$, SDRP$, and SVSRC$ will not return the status IE.INS (specified task not installed), but instead the message will be routed to URT . . .

If the application already uses a task named URT . . . and the user creates a new task named URT . . . , there may be a compatibility problem. If so, clearing the TCB pointer with an OPEN command will point the application to use the earlier task.

1.3.10 Enhancement to Loadable XDT

Loadable XDT may now be configured so that at each entry to XDT (by means of SST, breakpoint, or single step), the next instruction to be executed is automatically displayed. This is achieved by setting the 10th bit in the /FLAGS=n qualifier when the LOA /EXP=XDT command is issued.

Additionally, it may be useful to be able to disable this feature at run time, as well as to enable/disable some of the other XDT features that are controlled by the /FLAGS= keyword. A new internal register, $X, may be opened to control the /FLAGS= options. For example, the following sequence will enable the instruction decode feature:

```
XDT>$X/
$X 000003 000013 Return
```

1.4 Software Problems Corrected in Version 4.4

This section describes software problems corrected in Version 4.4.

1.4.1 Data Caching

The following corrections have been made to data caching.

1.4.1.1 Data Caching May Have Deadlocked the System

Previously, data caching did not take any special consideration of any I/O requests issued by the Replacement Control Task (RCT . . .) This caused some I/O requests issued by RCT . . . to have the potential for interacting with data caching including the potential for system deadlock during a replacement operation. The changes incorporated in data caching ensure that data caching will not interact in any way with any I/O requests issued by RCT . . . so system deadlocks should not occur.

This problem has been fixed.

1.4.1.2 Problems with Write-check or Read-check

Previously, issuing a write with write-check (IO.WLC) QIO request and then issuing a read (IO.RLC) QIO request sometimes resulted in a loss of data due to interaction with data caching. This problem has been fixed.

1.4.1.3 File Corruption

Under extremely rare circumstances, a combination of data caching read-ahead and write operations occurring at the same time may have caused file corruption, due to certain interactions with the hardware. This problem has only been seen with RMS indexed files, although it may have been possible with other file structures. This problem has been fixed.

1.4.1.4 Enhancements to the SET /CACHE Display Command

The SET /CACHE display has been enhanced to be able to display the current cache settings on a device basis without the need to use RMD to see the current cache settings.

1.4.1.5 Correction to CDA

Previously, CDA gave an error message and incremented the error count if a data caching partition was not in the system even if the system was not supposed to have data caching partition. This problem may have affected command files that used the exit status from CDA.

Now, CDA no longer includes /CACHE when the /ALL qualifier is issued. The user can reliably determine whether or not the system should return an error if the system fails to find a cache region and can then design their command files to rely on the exit status of CDA.

1.4.2 Corrections to SYSGEN

This section describes corrections to SYSGEN.

1.4.2.1 SYSGEN Entered a Loop if More than 4 Units Existed on a DU-Type Controller

When AUTOCONFIGURE was run on a test system that had more than 4 units attached to the same DU controller, all the devices were found. However, when the information was passed to SYSGEN, SYSGEN did not handle the extra devices properly. SYSGEN assumed there was a maximum of 4 units per controller. SYSGEN displayed the following error message:

```
SGN - Controller A already has 4 units.
        You cannot put DU4 on it.
```

SYSGEN then prompted for a physical unit number for DU4:. If a unit number was given, SYSGEN repeated the above error message, and reprompted for the unit number. SYSGEN stayed inside this loop. The only way to exit the loop was to issue a Ctrl/Z out of SYSGEN.

If there are greater than 4 units per controller, SYSGEN will warn the user they are creating a non-standard configuration and will update the system rather than looping without giving an option to change. The problems of overriding Autoconfigure, lowering the number of controllers, and having the number of units out of range have also been corrected.

1.4.2.2 Problems with Con Display

A problem occurred after performing a SYSGEN and then running AUTOCONFIGURE on the host system and then electing to override the results to add devices. During the override, SYSGEN stepped through the DU devices and asked for the physical unit number and to what controller (a or b) the device was attached. The following answers were given: putting the RA60 as DU0 on DUA, and putting the RC25 as DU4/5 on DUB. Logical unit DU7 was set to have a physical unit number of 200. This was accepted.

After the SYSGEN was done and the system VMRed, the SYSVMR.CMD had VMR display some of the information in the system using a CON DISP command. This command listed:

```
DU7:    DUB177600:    OFL   DRIVER
```

The unit number in the second column was incorrect. However, after a boot of the generated system, a CON DISP CONTR ATTR ALL showed the proper unit number.

This problem has been fixed.

1.4.2.3 Invalid Question on Processor-Type after an Autoconfigure

Previously, the following string of questions may have been asked during a SYSGEN operation:

```
>;
>* CE330   Is your system clock programmable (KW11-P)? [Y/N D:N]: n
>;
>* CE350   Is your line frequency 50 Hz? [Y/N D:N]:
>;
>* Is the target processor a PDP-11/70? [Y/N D:N]:
>;
>;
>;
>;============================================================
>;  Choosing Peripheral Configuration      01-AUG-91 at 03:09
>;============================================================
```

Although the system was autoconfigured, the installer had elected not to override the results of autoconfigure and the system asked if this was a PDP–11/70 processor. Also, the system did not have the question in the standard SYSGEN format. For example, as a *CExxx* question.

This question was asked as a result of a system memory size being less than 128.KW. The default range checking has been modified to insure that any default memory size is in range.

1.4.2.4 SYSGEN Terminated Abnormally if a User Chose to Override ACF Results

Previously, the user may have encountered the following series of questions when generating a system if the user chose to override the ACF results:

```
>* CE010   What is your target processor type? [S R:5.-12. D:"11/73"]: 11/93
>* CE020   Do you want the Full-functionality Executive? [Y/N D:Y]: y
>* CE120   Do you want support for communications products
>*         (such as DECnet, PSI, and LAT)? [Y/N D:N]: y
>* CE130   What is the system name? [S R:0-6 D:"RSXMPL"]: RSXMP1
>* CE264   Do you want IP11 industrial I/O subsystem support? [Y/N D:N]: n
>* CE270   Do you want to include XDT? [Y/N D:N]: y
>* CE280   Enter the crash notification device CSR
>*         address [O R:160000-177700 D:177564]:
>; Note:  Enter both the device and the logical unit number.
>;        For example, MM0: or DL1:.
>; Note:  Enter XX: if you desire loadable DU:, DL:, MU:, MS: or MM:
>;        crash driver support.
>* CE290   On what device and unit do you want crash dumps
>*         to be written? [S R:2-6]: xx:

AT.T0  -- Bad range or default specification
.ASKN ['TEMPN%D.':2044.:DFLTN] $S$YSZ CE310   Enter memory size (in K words)
.ASKN [256.:2044.:DFLTN] $S$YSZ CE310   Enter memory size (in K words)
```

This error occurred because the string "DFLTN" was not enclosed in quotes which allowed substitution. This only occurred if the user elected to override ACF results. If the user answered N to that question, the SYSGEN proceeded normally.

This problem was caused by SYSGEN and ACF incorrectly identifying the CPU-type. SYSGEN and ACF were setting the system memory size at 124.K which was less than the low range for the question. SYSGEN now has better default checking. Autoconfigure has also been corrected to identify the PDP–11/93 as a 22-bit processor.

1.4.3 Corrections to ACF

This section describes corrections made to ACF.

1.4.3.1 ACF Did Not Correctly Determine the Processor Type and Memory Size on a PDP–11/93 Processor

Previously, ACF did not correctly determine the processor type or the memory size. The highest memory size reported by the ACF task was 1920_{10} KiloWords, even for systems that supported a maximum memory size of 2048_{10} KiloWords.

Autoconfigure has now been corrected to now size memory up to 2044. KW on the following processors:

- PDP–11/23+
- PDP–11/53
- PDP–11/53+
- PDP–11/73
- PDP–11/83
- PDP–11/84
- PDP–11/93
- PDP–11/94

1.4.3.2 ACF Did Not Identify a PDP–11/93 Correctly During an Installation of Micro/RSX Using the RX50 Kit

After installing the Micro/RSX floppy disk kit (with all options installed except the reference disk) and examining ACFPAR.DAT to determine the results of ACF, the user found the CPU was incorrectly identified. The following is the contents of the ACFPAR.DAT file that was generated:

```
CPU=11/00
LIN=60
CON=DLA,CSR=174400,VEC=160
CON=DUA,CSR=172150,VEC=154,UNI=(3,0)
CON=MUA,CSR=174500,VEC=260
CON=YLA,CSR=177560,VEC=60,UNI=(1,0)
CON=YLB,CSR=176500,VEC=300,UNI=(1,0)
CON=YLC,CSR=176510,VEC=310,UNI=(1,0)
CON=YLD,CSR=176520,VEC=320,UNI=(1,0)
CON=YLE,CSR=176530,VEC=330,UNI=(1,0)
CON=YLF,CSR=176540,VEC=340,UNI=(1,0)
CON=YLG,CSR=176550,VEC=350,UNI=(1,0)
CON=YLH,CSR=176560,VEC=360,UNI=(1,0)
CON=MUB,CSR=160404,VEC=370,UNI=(1,0)
```

Corrections in ACF caused additional bits to be cleared, including the bits that designated a system as a PDP–11/93 or PDP–11/94 processor. This should not affect the user or the system, except in the case of a CPU-based error log packet, which are extremely rare.

1.4.4 Corrections to Shadow Recording

The following section describes corrections that have been made to shadow recording.

1.4.4.1 SHA Terminated with a Memory Protect Violation

Previously, shadow recording would terminate with a "Memory protect violation" if there was no bad block data found on the two disks. Shadow recording will now report the error as:

```
SHA -- No bad block data found on input volume.
```

1.4.4.2 SHAERR Reported Invalid LBNs on RA-Type Devices

When an error occurred on a shadow set, the logical block number printed by SHE . . . was incorrect. This occurred because DUDRV did not convert the LBN to track/sector/cylinder parameters prior to using the information. This problem has been fixed.

1.4.4.3 User Routine Enhancements for SHAERR

In order to increase the usefulness of the user action routine in the shadow recording error task (SHE . . .), the following changes have been included:

- The internal variable LOST, has been changed to the global symbol $LOST, to allow access to the number of packets lost due to saturation.
- The entry point, $SHERR, has been made a local symbol so the variable in SYSCM can be accessed without conflicting with the local task entry point label.

1.4.5 Backup and Restore Utility (BRU) Corrections

The following corrections have been made to the BRU utility.

1.4.5.1 Error with /EXCLUDE Using Wildcards

Previously, issuing a command line similar to the following caused all files except the first one in the [named] directory to be copied, even though they should have been excluded. The file copying also occurred if a numbered directory was specified.

```
>BRU/EXC/BAC:name/MOU LB:[*,*]*.LOG,[named] MT1:
```

The file copying problem has been fixed.

1.4.5.2 BRU Dismount Problem

Dismounting a drive after a multivolume tape backup caused the following error message:

```
DMO -- I/O error on MU0:
```

This problem occurred because a change was made to have BRU dismount the various tapes of a BRU backup in a previous release. This was done to prevent users from appending a backup set to any tape other than the first tape of a multivolume set, as BRU was not able to restore a backup set if it did not start on the first tape of the set. To ensure the tape was not used for appending, BRU issued a command that rewound and unloaded the tape. This had the effect of taking the tape unit off line. When DISMOUNT sent its command to the tape drive, the command was rejected because the drive was not on line.

This problem has been corrected.

1.4.5.3 Invalid Date or Time Error with /REVISED

When using the /REVISED switch, BRU issued the following error message on files that contained no revision date:

```
BRU -- *WARNING* -- Invalid date or time (filename)
```

This problem has been fixed.

1.4.5.4 Problems with Creating the Master File Directory

When BRU/INI and INI created the Master File Directory (000000.DIR), the directory was not marked as contiguous even though the file was only one block long. This did not cause a problem for any RSX system. However, it would cause a boot failure on VAX 8xxx systems because the boot disks are copied using BRU. This problem has been corrected.

1.4.5.5 BRU Could Not Abort When Used With a TK50

Previously, BRU could not be aborted when used with a TK50. Issuing a Ctrl/C key sequence followed by an ABO BRUT0 command resulted in BRU being marked for abort. However, BRU did not abort. You had to reboot the system to stop the BRU operation. This problem has been fixed.

1.4.5.6 Help File for BRUSYS Updated

Previously, Help BRU STANDALONE instructed the user to use VMRM42. This was an outdated version of VMR. Also, both the VMRM46 and VMRM47 version of VMS were on the kit in UIC [6,54]. The BRU help text now indicates that VMRM47 is the correct version of VMR to use. Additionally, VMR47 is the only version of VMR in UIC [6,54].

1.4.5.7 BRU Error Message Contains Extraneous Text

While running a BRU backup from one disk to (192,000 blocks) to another (13,000), you sometimes received the following warning:

```
This disk will not contain a hardware bootable system 002300.DIR;1
```

The last part of this warning indicating a file directory name was misleading. This problem has been fixed.

1.4.5.8 BRU Aborted Following a Tape Error

Tape errors during the disk-to-tape backup operations occasionally caused BRU to terminate with an error message of "odd address trap."

This problem has been fixed.

1.4.6 Correction to SHUTUP

SHUTUP.TSK did not complete properly if the CO: device was logged in. SHUTUP.CMD would not execute and the CPU would not halt. This problem has been fixed by changing SHUTUP to log out all users before turning off console logging.

1.4.7 Corrections to Peripheral Interchange Program (PIP)

The following corrections have been made to the PIP utility.

1.4.7.1 Problem with Checking the Directory Specification

PIP was not checking the directory syntax when the /DF switch was used. It was accepting a directory name greater than 9 characters, and even embedded hyphens, underscores, and dollar signs. The error was reported only when issuing a subsequent command without a directory specification. This problem has been corrected.

1.4.7.2 Specifying Files Using a Wildcard

PIP was ignoring input wildcard file specifications when reading files from an ANSI tape. PIP was selecting all files instead of just those that matched the wildcard file specification supplied by the user. This worked correctly prior to Version 4.3. This problem has been fixed.

1.4.7.3 Problem with Using Wildcards and Creating a Directory

Previously, while doing a file copy, PIP set the filename and type blank if the command line contained explicit wildcards in the destination field and PIP had to create the directory for the file. This problem has been fixed.

1.4.7.4 Problem with Issuing the PIP/LI Command

In the previous release, issuing the PIP [1,]/LI command caused PIP to abort whether it was invoked by MCR or DCL. This problem has been corrected.

1.4.7.5 PIP Did Not Give Correct Error for Command [34,477]/LI

Previously, issuing a command such as PIP [34,477]/LI resulted in the following incorrect error message, "Cannot find directory file."

Now, issuing this command will result in the following correct error message, "Bad directory syntax".

1.4.8 Problems with Submitting a Batch Job with the /NOPRINT Switch

On RSX–11M–PLUS Version 4.2, submitting a batch job with a logical device name in the batch file specification and with the /NOPRINT switch caused a "log file open error" from the batch processor on systems generated without extended logical name support. For example:

```
>ASN DB1:=CC25:/GBL
>SUB /NOPRINT=CC25:[240,1]NORA.BAT

BPR-*FATAL*-Log file open error (BAP0  )
```

If the real device name was used or if the /NOPRINT switch was omitted, the SUBMIT command worked. This problem has been fixed.

1.4.9 Using a CLI Build with the /CL Switch Correction

Using the install-run-remove option to run a task which had been built with the /CL (CLI) switch caused a system crash. Install printed the "INS – Task may not be run" error message and the system crashed at some point after the message was printed. This problem has been corrected.

1.4.10 IO.EIO QIO Request

The IO.EIO QIO request to an unattached remote terminal line on a non-I&D system caused a system crash. This problem has been fixed.

1.4.11 SHOW LOGICALS

The DCL command SHOW LOGICALS did not accept a qualifier and a parameter on the same command. For example, the SHOW LOGICALS xxx /SYSTEM command failed with the following error:

```
SHOW  --  Extraneous input - Check for proper command structure.
```

This problem has been corrected.

1.4.12 KW11–P Clock

When trying to use the KW11–P clock by means of a CINT$ routine, when the clock was not being used as the system clock, the following DSW error was returned:

```
IE.RSU (Vector in Use)
```

This problem has been corrected.

1.4.13 FAL

On Version 4.3, FAL would hang if a remote node issued a directory command with a blank file specification. For example:

```
DIR REMNOD"usernam password"::LB:[1,2]
```

This problem has been corrected.

1.4.14 Help Files

Previously, HELP INDIRECT ENABLE for the Indirect Command Processor stated that the .ENABLE CLI and .ENABLE MCR operating modes were local whereas actual operation showed them to be global, with their effect being passed when your program chained to a new command file or when a new command file is invoked. The help files were incorrect in stating that the .ENABLE CLI and .ENABLE MCR operating modes are local. They are global; their effect is retained when invoking or chaining to a subsequent command file. The

documentation of the /CLI and /MCR switches and of the .ENABLE command in the *RSX–11M–PLUS Indirect Command Processor Manual* is correct. Section 2.5 states the switches that are used in initiating indirect processing are used as defaults when executing subsequent @filespec or .CHAIN commands. Further, the CLI and MCR operating modes are specified in Section 2.6.12 as being *retained* when Indirect passes to a lower-level command file. The help file has been corrected in RSX–11M–PLUS Version 4.4 to correctly state that the CLI and MCR modes are global.

1.4.15 Support for RA82 Errors

Previously, RPT trapped on RA82 SDI errors. The MSCP8x error log control module has been updated to include support for RA82 SDI errors. This problem has been fixed.

1.4.16 Corrections to Error Logging

This section describes corrections to error logging.

1.4.16.1 ERRLOG Opened Wrong File for Append and/or for Random Access

Previously, ERRLOG would open the wrong file for append (instead of ERRORS.LOG) and/or for random access (instead of ERRORS.ACC).

This problem has been corrected.

1.4.16.2 Error Logging of Volume Not Mounted Packets for DU Disks

The disk controller error logging has been changed to ignore "Volume not mounted" packets. These are not errors and can rapidly clog up error log files.

1.4.16.3 Error Logging Support for PDP–11/84

Previously, the system would not recognize a PDP–11/84 as a valid device type when that device was entered in the error logging tune command file, as shown in the following example:

```
UTRPSU$ * Enter CPU type [S]: 11/84
UTRPSU$ ;
UTRPSU$ ;       Unrecognized CPU type 11/84 - Please enter again
UTRPSU$ ;
UTRPSU$ * Enter CPU type [S]: 11/83
UTRPSU$ ;
UTRPSU$ ;       Now  enter  the  devices in your configuration separated by
UTRPSU$ ;       commas
```

The TUNE.CMD file has been updated to include the PDP–11/84 as a valid CPU-type.

1.4.17 Using the .LIMIT Feature with Tasks

The MACRO .LIMIT feature did not work correctly with a task built with the /EX feature (extend tasks with memory resident overlays). This problem has been fixed.

1.4.18 Parity Problems with Terminal Multiplexers

Previously, unsynchronized access to terminal multiplexer CSRs in terminal driver module TTYHV caused parity to be set on incorrect terminal lines. This problem has been fixed.

1.4.19 Printing to a Printer Connected to a LAT Port

Previously, printing to a shared printer connected to a LAT application port occasionally produced the console error message:

```
LAT -- CONNECT FAILURE #-3              (Device not ready)
```

A LAT connection QIO timing problem exists that may cause an application to hang or not behave as expected on a heavily loaded Ethernet. This timing problem exhibits itself most often when an application does a disconnect followed by a connection in quick succession, as could happen in the print despooler. Although the despooler does not detach until all relevant jobs have been processed and cleared from the queue, there can be occasions when it would immediately attempt to connect again. The print despooler task includes a correction for this problem. If the "Device Not Ready" status is returned from a LAT printer connect, the despooler will wait for one second and retry the connect. After five failures, the despooler will give up and report the error as shown above.

1.4.20 UFD Created Directory Files with Embedded Blanks

Previously, UFD did not correctly handle the directory specification when the default directory was specified. For example:

```
$ set def/named lb:[61,40]
$ cre/dir []
$ dir DIR -- Cannot find directory file
DR:[61,40]
```

This set of commands resulted in UFD creating "61 40.DIR" rather than "061040.dir." When the command UFD ddn:[ ] was entered, the directory specification should have defaulted to the default directory string (assuming there was one). In the case where this string existed and was in UIC format (i.e. [g,m]), UFD created the numbered directory string with leading zeros suppressed, resulting in the creation of directory files with embedded blanks (e.g. "7 44.dir"). This problem of suppressing leading zeros has been fixed.

1.4.21 Incompatibility Between VMS DIFFERENCES Command and RSX DCL DIFFERENCES Command

The DCL DIFFERENCES command did not accept DIFF A.TXT, B.TXT but would accept DIFF A.TXT B.TXT.

The description of the DIFFERENCES command in the *RSX–11M–PLUS Command Language Manual* shows only the blank space as a valid file delimiter. A comma (,) can now be used as a valid file delimiter.

1.4.22 Enhancement to the DCL SORT Command

Previously, DCL did not allow use of the /INDEXED_SEQUENTIAL qualifier to the input and the output file for the DCL SORT command. DCL has been modified to allow use of the /INDEXED_SEQUENTIAL qualifier to both the SORT input and output files.

1.4.23 Correction to FLX

Previously, FLX did not display the maximum number of blocks on a tape if the number of blocks was too large. This problem has been fixed.

1.4.24 Task Builder (TKB)

Support has been added for using the MACRO .LIMIT directive in conjunction with the EXTM$ system directive.

1.4.25 Undefined Symbol Error in RPT

Previously, RPT encountered an "undefined symbol referenced in expression eval" error when attempting to process a packet in a DSP4P1 module.

This problem has been corrected.

1.4.26 Changing Crash Device from BOO Request Did Not Pass Information to Display

Previously, the system was modified to display three digits for the crash unit. However, this change was not propagated back to INITL. This problem has been fixed.

1.4.27 DSA Timeout Processing Led to Hung Tasks

DSA drivers should have had a timeout code to prevent the drive/controller from being hung for more than limited period of time. If a attempt to mount a drive hung, and the mount was then aborted, the drive was unusable. The only was to regain use of the drive was to reboot the system.

Timeout processing context should have been kept for the oldest outstanding command to the DU controller. However, the driver was improperly updating the context. This problem has been fixed.

1.4.28 Support for Inputting Lower-Case Characters in the TUNE.CMD and TUNERL02.CMD Files

Previously, the TUNE.CMD and TUNERL02.CMD command files did not allow the user to enter the device names using lower-case characters.

Now, the user can enter characters from terminals supporting both upper-case and lower-case characters.

1.4.29 Submitting a File That Did Not Exist Resulted in an Error

Previously, if you tried to submit a file that did did not exist to a BATCH job, you would receive the following error:

```
>chu 1,1
>subm /noprin=do.cmd
22:16:53  Task "SUBT0 " terminated
          Odd address or other trap four
          R0=022135
          R1=000030
          R2=001062
          R3=006022
          R4=006060
          R5=000000
          SP=000346
          PC=020002
          PS=170000
>
```

This problem has been fixed.

1.4.30 Support for PDP–11/94 in the SAVSIZ List of 2044KW CPUs

The PDP–11/94 has been added to the SAVSIZ routine, which sizes memory, as an Unibus CPU that can have a memory size greater than 1920KW.

1.4.31 Support for the Angle Bracket (< >) Characters for Directory Specifications

Support has been added to the SET /DEF and the SET /UIC commands to allow the user to use the angle bracket characters (< >) for a directory specification. This is especially useful on the German keyboard which requires the use of the compose key to produce a square bracket.

1.4.32 Support for J–11 UNIBUS Systems in the Crash Driver

Previously, the crash driver for the TU80 would only write the first 3840Kb of the memory on PDP–11/94 systems (or other J–11 based UNIBUS systems) that had 4Mb of memory. This created problems when these systems crashed, especially if the crash was related to DECnet–11M–PLUS or P.S.I./11M–PLUS, as that software was usually loaded in the top part of memory (above the 3840Kb border). This problem has been fixed.

1.4.33 Enhanced Support for RX02 Devices

Previously, if you generated a system for a Q–bus PDP–11 system with 22-bit support and you included RX02 devices, the system might crash when you tried to use one of the RX02s.

The problem occurred because DYCOM was one of the last RSX driver pieces to be installed in the system during a Virtual Monitor Console Routine (VMR) operation. DYCOM went into GEN instead of DRVPAR. This sometimes caused DYCOM to reside outside of the first 124 KW of memory. There was no comment in SYSVMR.CMD that it should be loaded below 124 KW. Additionally, there was an error in DYDRV where DYDRV tested for the base address of DYCOM.

These problems have been corrected. Other problems in online processing with DYDRV and DYCOM have also been corrected.

1.4.34 SET TERMINAL/INQUIRE Command Did Not Work on a VT420 Terminal

Previously, the SET TERMINAL/INQUIRE did not work properly on a VT420 terminal. The command appeared to work as no error was returned but issuing the command had no influence on the setup of the terminal characteristics. Now, the terminal will work the same as a VT3xx terminal.

1.4.35 Problems with ODT in an I&D System

Previously, the I&D version of ODT had a problem that appeared if you issued a sequence of commands similar to the following:

```
 .mcall exit$s
 .psect aacode,i
 .blkw 340.
 .psect zzcode,i
start: .rept 128.
 NOP
 .endr
 exit$s
 .end start

$ODTDT:(RW,D,LCL,REL,CON)     001304 001020 00528.
```

```
UTRPSU$ run odttst
ODT:VT3
_i
_1554/000240
_d
_1554/005020
_s
8B:001556
_$7/001556
_s
8B:$1E
_d
_1554/005020
_i
_1554/005020
_x
```

The problems included the following:

- After a register reference ($x/), when an address was referenced using ODT that was within the range of the $ODTDT p-sect of ODTID.OBJ, then ODT did not work properly.
- The I&D commands no longer worked and all memory references were from the D-space.
- The addresses printed were wrong when entering breakpoints or when you single-stepped. They appeared as *8B:$T*.

These problems occurred because the address not only has to be within $ODTDT but it also has to be between labels INTBEG and INTEND. This problem has been fixed.

1.4.36 Intermittent Problem When Getting Data Late Errors on RK06 or RK07 Drives

Previously, there was an intermittent problem that resulted in getting data late errors.

Due to potential inaccuracies in the way the RK06 reports word count when a data late is declared, the retry logic in the driver would restart the read or write operation in the wrong block.

This problem has been fixed.

1.5 Software Problems and Restrictions in Version 4.4

This section describes software problems and restrictions that have not been corrected for Version 4.4.

1.5.1 Restrictions in SYSGEN

This section describes restrictions in SYSGEN.

1.5.1.1 Selecting FCPMDL in SYSGEN

If you perform a SYSGEN and select both the nonstandard EXECUTIVE and FCPMDL, the system being generated will fail because FCPMDL is not present on the distribution and it is not rebuilt automatically. This problem occurs because SYSGEN was modified to eliminate rebuilding of all vectored privileged tasks (except SAV) and FCPMDL is the only vectored privileged task not shipped on the baseline system. You can rebuild FCPMDL by answering Y to the *BN010* question in the section on Building Nonprivileged Tasks and entering "FCPMDL" in response to the following question:

```
* BN020  Enter task name(s) [S]: FCPMDL
```

1.5.1.2 Incorrect Displays for Multiple MM Devices in SYSGEN

The RSX–11M–PLUS SYSGEN RH configuration display lists only one MM device when there are actually two MM devices. ACF finds two TU77s (and a TM03 formatter) but SYSGEN prompts for the number of tapes drives (question CP1608) and formatters (CP1612). SYSGEN does configure correctly for two MM devices.

1.5.1.3 Specifying a Nonexisting Device as the Crash Device During SYSGEN

If a nonexistent device is specified as the crash device during a SYSGEN operation, then you cannot change the crash device after the initial boot of the system. If the system reports an error, you must SYSGEN the system again and specify an existing device as the crash device.

1.5.2 Restrictions on Year 2000 Support

The following sections describe restrictions on the Year 2000 Support.

1.5.2.1 Special Error Log Display Commands Do Not Use the 4-Digit Year Field

The special error log commands will still display time stamp information in the following "dd-mmm-yy" format:

```
ELI /SH/NE
ELI /SH/CU
ELI /SH/HI
```

These problems will not be fixed.

1.5.3 Error Logging Restrictions

This section describes restrictions to Error Logging.

1.5.3.1 Support for Error Logging History Summaries

Error logging history summaries are no longer supported. See Section 2.8.

1.5.3.2 Incorrect Displays of Register Fields for DU and MU Devices

The errorlog driver internal timeout packet SA register fields for both DU and MU devices are not being displayed properly.

1.5.4 BRU Restrictions

This section describes restrictions to the Backup and Restore Utility (BRU).

1.5.4.1 Use of Wildcards in BRU

BRU supports the asterisk (*) character only as a replacement of an entire field in a file specification. BRU will issue the "Invalid value or name" fatal syntax error when an asterisk is used to replace part of a file specification field.

1.5.4.2 Problem with the /NEW_VERSION Qualifier

A problem with the /NEW_VERSION qualifier occurred when used in a BRU command line during a disk-to-disk backup. From the input device, BRU backed up multiple versions of a file in the order that they appeared in a directory, instead of in ascending order by version number. As a consequence, when the files were restored to the output device, new version numbers were assigned when each file was restored, and the original order of the files was not preserved. Thus, the last version of a file on the output device may not have been the last version created on the input device.

The following workarounds are suggested to avoid this problem:

- Sorting files on the input device.
- Purging files prior to backing up your files.

1.5.4.3 BRU Does Not Give Correct Error Messages When Backing Up a Named Directory That Does Not Exist

BRU does not give a correct error message when backing up a named directory that does not exist. It does not give any error messages during the copy pass. Also, a number of EOT marker errors occurred before the verify pass began. BRU also asked for another tape even though there was nothing for BRU to write. The following example shows the results of backing up a named directory that does not exist:

```
>bru /rew/mou/ver:double/bac:test du1:[jjmrel] mu:
BRU - Starting Tape 1 on MU0:
BRU - End of Tape 1 on MU0:
BRU -- *WARNING* -- EOT marker error
I/O error code -4
BRU -- *WARNING* -- EOT marker error
I/O error code -4
BRU -- *WARNING* -- EOT marker error
I/O error code -4
BRU -- *WARNING* -- EOT marker error
I/O error code -4
BRU -- *WARNING* -- EOT marker error
I/O error code -4
BRU - Mount another tape
```

1.5.4.4 Restriction on Incremental Backups

If BRU performs an incremental backup with the following command and there are no files after the specified date, then BRU does not properly terminate the tape:

```
BRU   /MOU/REW/TAP:label/BAC:name/REV:AFT:date DU1:   MU0:
```

If you do not specify the /VER switch, BRU gives the "No files found" and then appears to terminate normally. If you then issue a "BRU /REW/DIR/BAC:name MUO:", BRU responds with the following error messages:

```
*FATAL* -- Tape read error
I/O error code -10
```

If you specify the /VER switch, BRU returns the following messages after the end of the writing pass:

```
*WARNING* -- EOT marker error
I/O error code -4
Mount another tape
Mount Tape 1 on MUO:
```

BRU repeatedly issues the "Mount Tape 1 on MUO:" message.

1.5.5 DCL Command ABORT Taskname/TAS Does Not Translate Correctly

When an ABORT taskname/TAS command is entered, it translates to ABORT taskname/TERM=TAS, which is incorrect. It should be ABORT/TAS taskname.

The correct syntax for the DCL ABORT command requires that the /TASK qualifier follow immediately after the ABORT command. Furthermore, the MCR ABO command does not allow any of the following (you get an "invalid keyword" error message):

```
ABO/TAS task
ABO /TAS task
ABO task/TAS
ABO task /TAS
```

ABO task does work in MCR, which is identical to the action of the DCL ABORT command.

1.5.6 Tasks Hang When Attaching to Another Terminal Already Attached to a Task

Tasks would occasionally hang when they attempted to attach a terminal that was already attached to another task. The attach QIO request never returned and the task had to be manually aborted.

As a workaround, the Marktime directive (MRKT$) can be used to implement a timeout function within the application code. If the attach function is issued as a QIOW, you can precede it with a MRKT$ directive, specifying the same event flag number using the I/O status block to distinguish between timeouts (IS.PND) and I/O completion (other values). Alternatively, the MRKT$ directive can specify an AST routine to be entered at the expiration of the timer with a CMKT immediately following the QIOW. There are other workarounds such as using WTLO or WTSE with QIO and MRKT$ directives. In these workarounds, the expiration of the time interval without I/O completion would signal the need for an I/O kill to be issued.

1.5.7 ELI/SH Reports No Errors if ELI/NOLIM Has Been Set

If error log limiting has been disabled with the ELI /NOLIM switch, then subsequent hard errors are not displayed by the ELI /SH switch even though a full report generated with RPT shows that device hard errors have been logged.

This restriction still exists.

1.5.8 Problem with Unused Terminal Ports

The system can crash if you have an unused terminal port with a cable connected to the port. This problem can be eliminated by either removing the unused cables or setting the unused ports to slaved ports.

1.5.9 Task Builder (TKB) Restriction

To obtain the long version of the memory allocation file (map) when building an I&D task, you should specify only the /–SH switch (the DCL /LONG qualifier) after the map file in the command line. The /MA switch (the DCL /MAP qualifier) should not be specified with the /–SH switch for I&D tasks.

1.5.10 RMD Restriction

The following restriction applies to RMD:

If the RMD C (cache) page is being displayed interactively and the caching configuration is changed, RMD must be exited and a new RMD C command issued to display the new caching configuration correctly. RMD cannot refresh the display correctly for a caching configuration changed during an existing RMD display.

1.5.11 PIP Restriction

When requesting PIP to create a file to be contiguous with a file size of 32767_{10} blocks (77777_8) or less, PIP does not create the file and issues the following error message when there is not enough contiguous space:

```
PIP -- Allocation failure - no contiguous space
```

However, when the requested file size is 32768_{10} blocks (100000_8) or larger, PIP creates the file with zero blocks but still issues the same error message when there is not enough contiguous space.

1.5.12 Indirect Preprocessor Restriction

The Indirect Preprocessor (IPP) does not handle .END statements in an ENABLE data block correctly.

1.5.13 Queue Manager Restriction

If a printer on an asynchronous terminal line is not functioning due to a fault or power failure after the queue manager submitted a job to the printer, the queue manager can go into a deadlock state. The print job is shown as active although nothing is being printed and the job cannot be aborted, killed or deleted from the queue.

1.6 Supplementary Software Information

This section provides additional information on the following topics:

- Changes in the utility used to save and restore data on media
- Changes to File Control Services
- Restrictions and additional information on the Taskbuilder utility

1.6.1 Disk Save and Compress Utility

RSX–11M–PLUS Version 4.3 was the last release to have the Disk Save and Compress utility (DSC) included on the software kit. Any data currently saved on media by the DSC utility will have to be restored using the DSC utility and then resaved using BRU. Beginning with RSX–11M–PLUS Version 4.4, customers will have to use the save and compression features of the Backup and Restore Utility (BRU).

1.6.2 File Control Services Changes Supplementary Information

File Control Services (FCS) has been modified to support VMS ancillary control process (ACP) functions that are needed for compatibility with future RSX products and versions of VAX–11 RSX. These modifications increase the size of the FCS code that is included in the task image. Because the increase in FCS code size may affect the building of some large tasks, FCS routine versions that do not have VMS ACP support are included in the object library [1,1]NOVACPLIB.OLB and in the concatenated object module [1,1]FCSNOVACP.OBS.

Some large tasks that have a complicated Overlay Description Language (ODL) may be affected by the internal reorganization of specific FCS modules. This situation is indicated by "multiply defined symbol" errors occurring in ODL structures that were previously valid. You must correct the situation by revising the ODL structure.

The object library [1,1]NOVACPLIB.OLB can be used to build individual tasks that do not have VMS ACP support. The default routines in the system library remain unchanged. If you replace the routines in the system library (SYSLIB) with the routines from the concatenated object module [1,1]FCSNOVACP.OBS, you will affect all the tasks that are built using the system library, and you will be unable to build tasks with VMS ACP support.

If this support is not desired, it can be removed from the system library. The concatenated object module [1,1]FCSNOVACP.OBS is included on the kits and contains the affected modules with the support removed. To remove this support from your system, replace [1,1]FCSNOVACP.OBS in the system library by using the following Librarian Utility Program (LBR) command:

```
LBR>[1,1]SYSLIB/RP=[1,1]FCSNOVACP.OBS [Return]
```

If you also want to remove extended logical name support from the system library by replacing [1,1]FCSNOLOG.OBS in the system library, you must first replace FCSNOVACP.OBS. Some of the same modules are affected by both extended logical name support and VMS ACP support, and the modules are included in both of the concatenated object modules. If FCSNOLOG.OBS is replaced first, some modules in the system library will contain extended logical name support when FCSNOVACP is replaced because the modules in FCSNOVACP.OBS contain extended logical name support.

1.6.3 TKB Restrictions and Supplementary Information

The following section describes TKB restrictions and supplementary information.

1.6.3.1 OTS Fast Map Restriction

The Object Time System (OTS) Fast Map routine uses the RSX–11M–PLUS fast-mapping facility, which means that the task must not use the IOT instruction for any purpose except fast mapping. (For more information on the fast-mapping facility, see the *RSX–11M–PLUS and Micro/RSX Executive Reference Manual.*)

Caution

Be sure to use both TKB and the system library supplied for your system to ensure that the correct OTS modules are incorporated in your task. If you use the Version 4.1 TKB with an incompatible (older) version of the system library to build an autoloaded overlaid task, the following new fatal error message is issued:

```
TKB -- *FATAL* -- Incompatible OTS module
```

Conversely, if you use the Version 4.1 system library with an older version of TKB, the OTS module FSTMAP is automatically included in any task built, but the module will not be accessible. The FSTMAP module will not affect the execution or performance of your task.

2
Corrections to Documentation

This chapter corrects errors and omissions in the RSX–11M–PLUS and RMS–11 documentation sets. The section titles in this chapter are the titles of manuals that require correction.

2.1 RSX–11M–PLUS Batch and Queue Manual

In Section 2.6.1, page 2–15 of the *RSX–11M–PLUS Batch and Queue Operations Manual*, please add the following description of the /ALL qualifier to the SHOW QUEUE command:

/ALL
Displays information on all entries in all queues.

2.2 RSX–11M–PLUS Command Language Manual

Please add the following information to the *RSX–11M–PLUS Command Language Manual*:

- In Section 3.8.1, page 3–41, please note that the DCL command SET TERMINAL/INQUIRE supports the VT300-series terminals in VT200 mode only.
- In Section 3.8.1, page 3–42, please note that the DCL command SET TERMINAL/*model* supports the VT300-series terminals in VT200 mode only. You cannot specify VT3xx_SERIES as a terminal type.
- The following listed changes should be made to the /HEADERS:n and /MAXIMUM_FILES:n qualifiers in Section 5.11, which is the description of the DCL INITIALIZE command. Please make the following changes:
  - In place of all the text that follows the table in the explanation of the /HEADERS:n qualifier, please add the following text and table:

    The value used for /HEADERS:n is derived from the maximum number of file headers specified by /MAXIMUM_FILES:n, with consideration for pre-extending the index file if the maximum number of file headers will require a multiheader index file (see the description of /MAXIMUM_FILES:n for more information on multiheader index files). You can use Table 2–1 to determine an appropriate value to use for the /HEADERS:n parameter.

Table 2–1 Number of Index File Headers

Maximum Number of Files	Disk Size	Number of File Headers
Maximum files fewer than 25588	Disks fewer than 209 Mb	Headers=maximum/2
Maximum files greater than 25588 and fewer than 51176	Disks between 210Mb and 419Mb	Headers=25593
Maximum files greater than 51176 and fewer than 51693	Disks between 420Mb and 423Mb	Headers=25846
Maximum files greater than 51693 and fewer than 65500	Disks greater than 423Mb	Headers=51693

– Please replace the second paragraph and formula after the /MAXIMUM_FILES:n qualifier with the following paragraph, table, paragraph, and formula:

The maximum number of files varies according to disk size. You may want to use the values in Table 2–2 as the default approximations:

Table 2–2 Default Number of Files – Approximation

Disk Size	Approximate Number of Files
Up to 64Mb	Calculate the maximum number of files using the formula following the table
From 64Mb to 532Mb	Specify the number of blocks divided by 16 as the maximum number of files
Greater than 532Mb	Specify 65500 as the maximum number of files

The default used for the maximum number of files is the theoretical maximum possible number of one-block files, divided by eight. The following formula calculates the theoretical maximum number of files:

```
MAXIMUM FILES = ((N-(((N+4095.)/4096.)+9.))*127.)/258.
```

- Please add the following information to Section 5.11.1, page 5–80 at the end of the first paragraph:

  You must specify the /WRITE:n qualifier when formatting volumes on DL-type devices.

- In Section 5.12.5.1, Section 5.12.5.2, and Chapter 9, pages 5–94, 5–96, and 9–6, the format of the BACKUP command qualifiers /CREATED and /MODIFIED is incorrect. The correct formats are:

  Formats

  BACKUP/CREATED/arg

  BACKUP/MODIFIED/arg

  Where *arg* can be specified as either BEFORE:(dd-mmm-yy hh:mm[:ss]) or AFTER:(dd-mmm-yy hh:mm[:ss]).

- In Section 5.12.6, page 5–104, the second example is incorrect. The correct example is shown next.

```
$ BACKUP/MODIFIED/AFTER:(14-MAY-87 17:00)/VERIFY DU0: MS0: Return
```

This example backs up all files on the fixed disk that were modified after 5:00 P.M. on May 14, 1987. After all the files have been copied onto the tape, BACKUP verifies the tape. If files on the tape do not verify, BACKUP returns an error message.

- Please add the following text to Section 5.13.1 of the manual:

  The DCL commands SHOW ASSIGNMENTS and SHOW LOGICALS have a new functionality and a new display format. The commands can now display at your terminal all logical assignments for a specific logical name.

  Formats

  $ SHOW ASSIGNMENTS [logicalname][/qualifier]

  $ SHOW LOGICALS [logicalname][/qualifier]

  Parameter

  logicalname

  Specifies the logical name. You can also specify a portion of a logical name and either the generic wildcard character asterisk (*) or the specific wildcard character percent sign (%). The asterisk can represent any number of characters. The percent sign can represent only one character.

  If you specify quotation marks around the logical name, the SHOW ASSIGNMENTS command interprets the quotation marks literally, while the SHOW LOGICALS command removes the quotation marks.

  Qualifiers

  /ALL
  /GLOBAL
  /GROUP[:n]
  /LOCAL
  /LOGIN
  /SYSTEM
  /TERMINAL[:ttnn:]

  Refer to the *RSX–11M–PLUS Command Language Manual* for information on qualifiers.

  The display format of logical assignments has been modified to be more readable. The display specifies the logical name table and then lists the logical name assignments for the issuing terminal.

  Examples

```
$ SHOW ASSIGNMENTS  Return

(Session Login Logical Table for VT2:)

     "DCM"         =   "DU:[CACHE]"
     "ER"          =   "EDT/RO "
     "GO"          =   "SET /DEF="
     "HOME"        =   "SYS$LOGIN"
     "MAP"         =   "LB:[1,54]RSX11M.MAP"
     "R"           =   "RECALL"
     "REVIEW"      =   "DU1:[REVIEWS]"
     "SYS$LOGIN"   =   "DU:[USER]" [Final]
     "SYS$PROMPT"  =   "Jim>"

(Session Local Logical Table for VT2:)

     "SYS$CLI"     =   "DCL"

$
```

Displays all the login and local assignments for the issuing terminal.

```
$ SHOW ASSIGNMENTS S* Return

(Session Login Logical Table for VT2:)

     "SYS$LOGIN"  =   "DU:[USER]" [Final]
     "SYS$PROMPT" =   "Jim>"

(Session Local Logical Table for VT2:)

     "SYS$CLI"    =   "DCL"

$
```

Displays the login and local assignments that begin with the letter S for the issuing terminal.

```
$ SHOW LOGICALS "SYS$CLI"  Return

(Session Local Logical Table for VT2:)

     "SYS$CLI"    =   "DCL"

$
```

Displays the login and local assignments that match the string SYS$CLI for the issuing terminal.

```
$ SHOW ASSIGNMENTS "SYS$CLI"  Return

$
```

Displays no matching login and local assignments for the string "SYS$CLI" for the issuing terminal.

```
$ SHOW ASSIGNMENTS /ALL  Return

(System Logical Table)

     "LI"         =   "LB:" [Final]
     "SS"         =   "LB:" [Final]
     "WK"         =   "LB:" [Final]

(Group 7 Logical Table)

     "GROUP"      =   "DR5:[7,43]"

(Session Login Logical Table for VT2:)

     "DCM"        =   "DU:[CACHE]"
     "ER"         =   "EDT/RO "
     "GO"         =   "SET /DEF="
     "HOME"       =   "SYS$LOGIN"
     "MAP"        =   "LB:[1,54]RSX11M.MAP"
     "R"          =   "RECALL"
     "REVIEW"     =   "DU1:[REVIEWS]"
     "SYS$LOGIN"  =   "DU:[USER]" [Final]
     "SYS$PROMPT" =   "Jim>"

(Session Local Logical Table for VT2:)

     "SYS$CLI"    =   "DCL"

$
```

Displays all the system, group, login, and local assignments for the issuing terminal.

```
$ SHOW LOGICALS S*/ALL  Return
(System Logical Table)
     "SS"          =   "LB:" [Final]
(Session Login Logical Table for VT2:)
     "SYS$LOGIN"  =   "DU:[USER]" [Final]
     "SYS$PROMPT" =   "Jim>"
(Session Local Logical Table for VT2:)
     "SYS$CLI"    =   "DCL"
$
```

Displays the system, group, login, and local assignments that begin with the letter S for the issuing terminal.

- Section 8.1.3, page 8–5, states that the DCL command SET SYSTEM /CRASH_DEVICE is valid only for pregenerated operating systems. This is no longer true; the command is now valid for all RSX–11M–PLUS operating systems.
- The following section describes an enhancement and lists restrictions for the Backup and Restore Utility (BRU).

  The behavior of the BRU qualifier /ERROR with restore operations is unchanged. However, the qualifier can be used with double-buffered compare and verify operations on data backed up from a disk to an MU-type device. The default number of errors is 25.

  The BRU Utility has the following restrictions:

  - The /LENGTH qualifier does not apply to cartridge tape devices, such as the TK25 and TK50, and it does not function properly if used on those devices.
  - The /NEW_VERSION and /VERIFY qualifiers are mutually exclusive. They cannot be specified together in the same command line.
  - You must specify /REWIND/APPEND when placing the first backup set on a magnetic tape that contains a bootable system image.
  - The /TAPE_LABEL qualifier is ignored during a restore operation if there is a bootable system image at the beginning of the tape

2.3 RSX–11M–PLUS Indirect Command Processor Manual

Please make the following changes to Chapter 2 of the *RSX–11M–PLUS Indirect Command Processor Manual*:

- Change the final sentence in Section 2.4.2 to read as follows:

  See Section 2.4.6.1 for more information.
- In Section 2.6.16, page 2–61 of the manual, remove the second sentence of the third paragraph. In its place, please add the following text:

  The .FORM directive in the Indirect Command Processor (Indirect) includes support for VT200-series terminals in VT200 mode.
- Delete the following text in Section 2.6.22:

  Task not installed in system (.XQT, .WAIT)
- Replace the example in Section 2.6.26 with the following:

The following example is from an interactive terminal session:

```
>@ti: Return
AT.>.enable substitution Return
AT.>.sets a "1,2,"  Return
AT.>.parse a "," b c d Return
AT.>;'b' Return
>;1
AT.>;'c' Return
>;2
AT.>;'d' Return
>;                              (null substring)
AT.>.parse a "," b c Return
AT.>;'b' Return
>;1
AT.>;'c' Return
>;2,
AT.>Ctrl/Z
>@ <EOF>
>
```

- An initialization test has been added in the Indirect Command Processor. This test assures that the allocation of buffers and file structures does not extend into the internal string storage region. A failure is reported as

  ```
  AT. -- Initialization error, code 6.
  ```

  Change the explanation of this error code, in Appendix A.2 by replacing the last sentence. The resulting description reads as:

  6. Unable to allocate enough space for command and data I/O buffers.

     For privileged Indirect tasks, Indirect was not installed with a large enough increment value. The system manager should remove and reinstall Indirect with a larger increment or in a larger partition.

     For the nonprivileged Indirect task, the Executive directive Extend Task failed to return sufficient space for Indirect to allocate the buffers or Indirect was installed with an excessively large increment value. If it was installed with an increment value, the system manager should remove and reinstall Indirect with a smaller increment. Note that it is unnecessary to install the nonprivileged Indirect with an increment value, unless checkpointing is disabled or checkpoint space has not been allocated, because it will extend itself as required to the extent that address space is available.

2.4 RSX–11M–PLUS MCR Operations Manual

Please make the following corrections to the *RSX–11M–PLUS MCR Operations Manual*:

- In Section 3.6, page 3–12, the descriptions of the ALT keywords /RPRI and /TERM are incorrect. Delete the second paragraph, which is in parentheses, of the /RPRI keyword description. Also, the first sentence of the /TERM keyword description should read as follows:

  "Alters the priority of a task not initiated from the issuing terminal."

- In Section 3.7.7 of the manual, on page 3–22 immediately before the heading examples, please add the following text:

  The MCR command ASSIGN (ASN) can display at your terminal all logical assignments for a specific logical name.

Format

>ASN [logicalname][/keyword]

Parameter

logicalname

Specifies the logical name. You can also specify a portion of a logical name and either the generic wildcard character asterisk (*) or the specific wildcard character percent sign (%). The asterisk can represent any number of characters. The percent sign represents only one character.

If you specify quotation marks around the logical name, the ASN command interprets the quotation marks literally, while the DFL command removes the quotation marks.

Keywords

/ALL
/GBL
/GR[=n]
/SYSTEM
/TERM[=ddnn:]

Refer to the *RSX–11M–PLUS MCR Operations Manual* for information on keywords.

The display format of logical assignments has been modified to be more readable. The display specifies the logical name table and then lists the logical name assignments for the issuing terminal.

Examples

1. >ASN Return

```
(Session Login Logical Table for VT2:)

     "DCM"         =   "DU:[CACHE]"
     "ER"          =   "EDT/RO "
     "GO"          =   "SET /DEF="
     "HOME"        =   "SYS$LOGIN"
     "MAP"         =   "LB:[1,54]RSX11M.MAP"
     "R"           =   "RECALL"
     "REVIEW"      =   "DU1:[REVIEWS]"
     "SYS$LOGIN"   =   "DU:[USER]" [Final]
     "SYS$PROMPT"  =   "Jim>"

(Session Local Logical Table for VT2:)

     "SYS$CLI"     =   "DCL"

>
```

This example displays all the login and local assignments for the issuing terminal.

2. >ASN S* Return

```
(Session Login Logical Table for VT2:)

     "SYS$LOGIN"   =   "DU:[USER]" [Final]
     "SYS$PROMPT"  =   "Jim>"

(Session Local Logical Table for VT2:)

     "SYS$CLI"     =   "DCL"
```

```
>
```

This example displays the login and local assignments that begin with letter S for the issuing terminal.

3. `>ASN "SYS$CLI"` Return

```
>
```

This example displays no matching login and local assignments that match the string "SYS$CLI" for the issuing terminal.

4. `>ASN /ALL` Return

```
(System Logical Table)
     "LI"          =   "LB:" [Final]
     "SS"          =   "LB:" [Final]
     "WK"          =   "LB:" [Final]
(Group 7 Logical Table)
     "GROUP"       =   "DR5:[7,43]"
(Session Login Logical Table for VT2:)
     "DCM"         =   "DU:[CACHE]"
     "ER"          =   "EDT/RO "
     "GO"          =   "SET /DEF="
     "HOME"        =   "SYS$LOGIN"
     "MAP"         =   "LB:[1,54]RSX11M.MAP"
     "R"           =   "RECALL"
     "REVIEW"      =   "DU1:[REVIEWS]"
     "SYS$LOGIN"   =   "DU:[USER]" [Final]
     "SYS$PROMPT"  =   "Jim>"
(Session Local Logical Table for VT2:)
     "SYS$CLI"     =   "DCL"
>
```

This example displays all the system, group, login, and local assignments for the issuing terminal.

- In Section 3.22.2 of the manual, on page 3–57 immediately before the heading Examples, please add the following text:

  The MCR command DEFINE LOGICALS (DFL) can display at your terminal all logical assignments for a specific logical name.

  Format

  >DFL [logicalname][/keyword]

  Parameter

  logicalname

  Specifies the logical name. You can also specify a portion of a logical name and either the generic wildcard character asterisk (*) or the specific wildcard character percent sign (%). The asterisk can represent any number of characters. The percent sign represents only one character.

  If you specify quotation marks around the logical name, the ASN command interprets the quotation marks literally, while the DFL command removes the quotation marks.

Keywords

/ALL
/GBL
/GR[=n]
/SYSTEM
/TERM[=ddnn:]

Refer to the *RSX–11M–PLUS MCR Operations Manual* for information on keywords.

The display format of logical assignments has been modified to be more readable. The display specifies the logical name table and then lists the logical name assignments for the issuing terminal.

Examples

1. `>DFL "SYS$CLI"` Return

```
(Session Local Logical Table for VT2:)
     "SYS$CLI"    =   "DCL"
>
```

This example displays the login and local assignments that match the string SYS$CLI for the issuing terminal.

2. `>DFL S*/ALL` Return

```
(System Logical Table)
     "SS"         =   "LB:" [Final]
(Session Login Logical Table for VT2:)
     "SYS$LOGIN"  =   "DU:[USER]" [Final]
     "SYS$PROMPT" =   "Jim>"
(Session Local Logical Table for VT2:)
     "SYS$CLI"    =   "DCL"
>
```

This example displays the system, group, login, and local assignments that begin with the letter S for the issuing terminal.

- Please add the following information to Section 3.26 on page 3–67:

  The HELLO task displays login attempts on the console. If the account name or UIC is found in the user account file but the failure was due to an invalid password, the failure will be reported as:

```
hh:mm:ss  Login failure USERNAME        [ggg,mmm] TTnn:
```

  If the failure results from an invalid account name or UIC, the report will be:

```
hh:mm:ss  Login failure INVALID USER              TTnn:
```

  The overlay structure of the HELLO tasks (HEL.TSK and HELRES.TSK) has been modified to improve maintainability. Formerly, there was insufficient task address space to build either version to include the On-Line Debugging Tool (ODT). HELLO now includes two overlay segments plus a third segment that contains common data and routines, acting as a *root* for these two. Data and routines needed only by HELLO have been moved from the task root (HELROT) to one of the HELLO segments.

If you have added *user-provided* routines to your HELLO task, you will need to rebuild using the new overlay structure defined in HELBLD.ODL or HELRESBLD.ODL.

- Please add the following information to Section 3.27:

  The overlay structure of the HELLO tasks (HEL.TSK and HELRES.TSK) has been modified to improve maintainability. Formerly, there was insufficient task address space to build either version to include the On-Line Debugging Tool (ODT). HELLO now includes two overlay segments plus a third segment that contains common data and routines, acting as a *root* for these two. Data and routines needed only by HELLO have been moved from the task root (HELROT) to one of the HELLO segments. The HELP overlay segment is not affected by this modification.

- The following listed changes should be made to the /INF and /MXF keywords in Section 3.29, which is the description of INITIALIZE VOLUME command (INI). Please make the following changes:

  - In place of the text that explains the /INF keyword, please add the following in place of the first three paragraphs, table, and fourth paragraph:

    Specifies the number of file headers to allocate initially in the index file. The five system files (INDEXF.SYS, BITMAP.SYS, BADBLK.SYS, CORIMG.SYS, and 000000.DIR) are not included in the value for INF.

    The value used for /INF is derived from the maximum number of file headers specified by /MXF, with consideration for pre-extending the index file if the maximum number of file headers will require a multiheader index file (see the description of /MXF for more information on multiheader index files). You can use Table 2–3 to determine an appropriate value to use for the /INF parameter.

Table 2–3 Number of Index File Headers

Maximum Number of Files	Disk Size	Number of File Headers
Maximum files less than 25588	Disks less than 209 Mb	Headers=maximum/2
Maximum files greater than 25588 and less than 51176	Disks between 210Mb and 419Mb	Headers=25593
Maximum files greater than 51176 and less than 51693	Disks between 420Mb and 423Mb	Headers=25846
Maximum files greater than 51693 and less than 65500	Disks greater than 423Mb	Headers=51693

  - Please replace the second paragraph and formula after the /MXF keyword with the following paragraph, table, and formula:

    The maximum number of files varies according to disk size. You may want to use the values in Table 2–4 as the default approximations:

Table 2–4 Default Number of Files—Approximation

Disk Size	Approximate Number of Files
Up to 64Mb	Calculate the maximum number of files using the formula following the table
From 64Mb to 532Mb	Specify the number of blocks divided by 16 as the maximum number of files
Greater than 532Mb	Specify 65500 as the maximum number of files

The default used for the maximum number of files is the theoretical maximum possible number of one-block files, divided by eight. The following formula calculates the theoretical maximum number of files:

```
MXF = ((N-(((N+4095.)/4096.)+9.))*127.)/258.
```

- In Section 3.31, page 3–103, the description and format of the MCR command LOAD is incomplete. Please add the following sentence to the end of the second paragraph:

  "The Load command also loads the extended Executive partitions into memory."

  The correct formats of the Load command are shown next.

  Formats

  LOA[D] dd:[/keyword(s)]

  LOA[D] /EXP=expname[/keyword(s)]

- In Section 3.34, page 3–128, the formats of the OPE keywords /TASKD and /TASKI are incorrect. The correct formats are shown next.

  Formats

  /TASKD=taskname

  /TASKI=taskname

- In Section 3.34, the second sentence in the first paragraph on page 3–129 is incorrect; it should read as follows:

  "You are limited to the specified memory region (TASK or REG)."

  Also, the second, third, and fourth examples on page 3–129 are incorrect. The correct examples are shown next.

```
>OPE 0/REG=TSTREG [Return]
00000000/50712
```

This command opens the region TSTREG at location 0 and displays in octal the current value at the location.

```
>OPE 0/REG=TSTREG [Return]
00000000/50712 % [ESC] MCR
```

This command opens location 0 of TSTREG, displays the current value in octal, and then displays the Radix–50 value in ASCII format.

```
>OPE 0/REG=TSTREG [Return]
00000000/50712 % [ESC] MCR %DCL [Return]
00000002/00000 [Return]
00000004/14604
```

This command sequence displays the current value at location 0 in octal and then Radix–50 format, and then changes the value. The new value is also in Radix–50 format. Pressing the Return key enters the new value into location 0 and displays it in octal, then opens the next location in memory and displays its contents in octal.

- In Section 3.40, pages 3–138, 3–140, and 3–141, the format of the RUN keyword /UIC is incorrect in Format 3 and Format 4. The correct format is shown next.

  Format

  /UIC=[g,m]

- In Section 3.40, page 3–139, the definition of the keyword /UIC is incorrect. The correct definition is shown next.

  /UIC
  (Privileged keyword.) The User Identification Code (UIC) under which the task will be requested to run. This UIC also determines which files the task can access.

  The UIC has the format [g,m], where the variables *g* and *m* specify octal numbers between 1 and 377 that represent the group and member numbers, respectively. The square brackets are required syntax.

  When you specify /UIC, the UIC that you specify becomes the default and protection UICs for the task.

  When you do not specify /UIC, the terminal's UIC becomes the default and protection UICs for the task. This is the default.

- In Section 3.40, page 3–139, the definitions of the parameters *taskname* and *dtime* are incorrect. The correct definitions are shown next.

  taskname
  Specifies a 1- to 6-character name of an installed task.

  dtime
  Specifies a delta time. Delta time is an increment from the current time.

- In Section 3.40, pages 3–140 and 3–141, the definition of the parameter *taskname* is incorrect. The correct definition is shown next.

  taskname
  Specifies a 1- to 6-character name of an installed task.

- In Section 3.40, page 3–142, the definition of the parameter *$* is incorrect. The correct definition is shown next.

  $
  When you specify a dollar sign ($) in the command line, the parameter *dev* defaults to LB:, and [g,m] defaults to the current library UIC (usually, [3,54]). If the task cannot be found in the library UIC, INSTALL searches the system UIC on device LB: (usually, [1,54]).

  If you do not specify a dollar sign ($), RUN searches for the task image file in the UIC (on device SY) to which the terminal requesting the task is set.

Note

If LB has been reassigned to another device and the system UIC for that device contains privileged tasks built for another system, your system will fail.

- In Section 3.40, pages 3–144 and 3–145, the descriptions of the RUN command keywords /PRI=number and /UIC are incorrect. The correct description is shown next.

  /PRI=number
  Specifies the priority of the task. The value range is 1 to 250_{10}, where 250 is the highest priority. Standard number conventions apply: octal by default, decimal if followed by a period.

  If a nonprivileged user specifies a priority that is greater than 50_{10}, the priority is set to 50_{10}.

  The default is /PRI=50.

  /UIC=[g,m]
  (Privileged keyword.) Specifies the User Identification Code (UIC) under which the task will be requested to run. The square brackets are required syntax.

  If you specify a UIC, the UIC becomes the default UIC and protection UIC for the task. If you do not specify a UIC, your default UIC and protection UIC become the default UIC and protection UIC for the task.

  The default is the UIC of the terminal that issues the RUN command.

- Please add the following information to Table 3–4 in Section 3–42:

Table 2–5 SET Command Keyword Summary

Keyword	Description
Ensuring System Protection	
/INTRUSION:yes	When enabled, invalid login transaction blocks include the user's responses to the Account: and Password: prompts. The system manager can use this information to determine which patterns are being used in an intrusion attempt.

Resource Accounting has been expanded to include a new qualifier to the SET command. The qualifier is:

```
/INTRUSION:yes/no
```

When disabled, invalid login transaction blocks include the user's name and UIC, as found in the user account file, if the login failure results from a password discrepancy. If the user name or UIC is not found in the account file, the invalid login transaction block will simply list INVALID USER as the user name and the UIC field will be blank.

When enabled, invalid login transaction blocks include the user's responses to the Account: and Password: prompts. The system manager can use this information to determine which patterns are being used in an intrusion attempt.

The default is /INTRUSION:no

- In Section 3.42, page 3–154 and page 3–169, the format of the MCR command SET /INQUIRE is incorrect. The correct format is shown next.

  Format

  /INQUIRE=term

- In Section 3.42, page 3–161, the format of the MCR command SET /CRASHDEV is incorrect. The correct format is shown next.

  Formats

  SET /CRASHDEV=ddnn:[CSRaddr]

  SET /CRASH_DEVICE=ddnn:[CSRaddr]

  Also, the section states that the MCR commands SET /CRASHDEV and SET /CRASH_DEVICE are valid only for pregenerated operating systems. This is no longer true; the commands are now valid for all RSX–11M–PLUS operating systems.

- Please add the following information to Section 3.42, page 3–162:

  Format

  SET /CTRLC=ttnn:

  This qualifier establishes whether a CTRL/C causes an abort (/CTRLC=ttnn:) or causes an explicit MCR> prompt (/NOCTRLC=ttnn:).

- In Section 3.42, page 3–163 and page 3–164, the description of the MCR command SET /DPRO is incorrect. The text incorrectly states that the parameter *protection-spec* has two formats. The parameter *protection-spec* can only be specified in the format shown next.

  Format

  [RWED,RWED,RWED,RWED]

  In Section 3.42, page 3–164, the first paragraph under the bulleted items is incorrect. The paragraph should read as follows:

  The /DPRO keyword establishes a default file protection for your current session at the terminal. When you start a new terminal session, it resets your file protection to the system default protection (or to the default protection in your account file if one was specified when your account was created). To establish a default protection code for all of your future sessions at the terminal, specify the code as a command in your login command file (LOGIN.COM).

- In Section 3.42, page 3–169, add the following information:

  The MCR command SET/INQUIRE supports the VT300-series terminals in VT200 mode only.

- In Section 3.42, page 3–185, add the following information:

  The MCR command SET/TERM supports the VT300-series terminals in VT200 mode only. You cannot specify VT3xx as a terminal type.

- In Section 3.42, page 3–197, the description of the sixth example is incorrect. The correct example is shown next.

```
>SET /SECPOL Return
SECPOL=285.:640.:44%
```

Displays the amount of available secondary pool.

- Immediately before Section 12.1.4.5 on page 12–11 of the manual, please enter the following text:

  The Peripheral Interchange Program (PIP) qualifier /DD must be used with the ampersand (&) character.

2.5 RSX–11M–PLUS System Generation Guide and Installation Guide

Please add the following information to the text in Section 3.2.2, Choosing Executive Options (CE), on page 3–26. This text should be added after the question * *CE200 Which FCP do you want? [S R:1-6 D:"FCPLRG"]:* and just before the * *CE210: Do you want support for file windows in secondary pool? [Y/N D:N]:* question:

If you entered FCPMDL, then you must answer Y to question BN010 in the section on Creating the System Image File and enter FCPMDL to question *BN020 Enter task name(s) [S]: FCPMDL*. Both of these questions appear on page 3–64 of the *RSX–11M–PLUS System Generation Guide and Installation Guide.*

The paragraph in Section 5.4.7 on page 5–43 is incorrect. Please replace it with the following paragraph:

RMS–11 Version 2.0 is included on the pregenerated kit disk. The RMS–11 segmented library (RMSRES, and RMSLBL through RMSLBM) and all the RMS–11 utilities are already installed in the system image. No further installation is needed, unless you install the DECnet package on your system and you want to use the RMS–11 remote access facilities. See Section 5.4.1 in the *RSX–11M–PLUS System Generation Guide and Installation Guide* for information on installing the RMS–11 remote access package (DAPRES).

2.6 RSX–11M–PLUS and Micro/RSX Crash Dump Analyzer Reference Manual

Please make the following correction to the *RSX–11M–PLUS and Micro/RSX Crash Dump Analyzer Reference Manual*:

- Section 1.1.1 specifies that, for RSX–11M–PLUS operating systems that are not pregenerated, you must select the crash dump driver during system generation. This is no longer true; RSX–11M–PLUS now supports loadable crash dump drivers.

  You can select loadable crash dump support during system generation and choose a loadable crash driver during the system startup procedure. Crash dump devices can be DU-, DL-, MU-, MS-, and MM-type devices. Refer to the *RSX–11M–PLUS System Generation and Installation Guide* for more information on including loadable crash support in your system if it is not pregenerated.

  In addition, for systems that have loadable crash dump support included, you no longer must perform another SYSGEN to change the crash dump device or unit number; you can use either the DCL command SET SYSTEM /CRASH_DEVICE or the MCR command SET /CRASHDEV. The previous commands are now valid for all RSX–11M–PLUS operating systems. Refer to the *RSX–11M–PLUS Command Language Manual* and the *RSX–11M–PLUS MCR Operations Manual* for more information on the commands.

- Please add the following text to Section 1.1.2 of the manual:

  Crash drivers on RSX–11M–PLUS systems with 4Mb of memory have the following functions:

  – The MU and DU crash drivers on 4Mb systems keep track of the amount of memory dumped and terminate the dump when 2044K words of memory have been dumped.

  – The DL crash driver on 4Mb systems dumps 2044K words of memory.

- Section 1.1.2.1 specifies an incorrect format for the MCR command SET /CRASHDEV (or SET /CRASH_DEVICE). The correct format is shown next.

  Formats

  SET /CRASHDEV=ddnn[:CSRaddr]

  SET /CRASH_DEVICE=ddnn[:CSRaddr]

- Section 1.2.1 specifies that, for systems that are not pregenerated, transferring processor control to the crash dump driver depends on whether you built the Executive Debugging Tool (XDT) into your system during system generation. The information in this section is generally incorrect because RSX–11M–PLUS Version 4.0 included loadable XDT support for all RSX–11M–PLUS systems and RSX–11M–PLUS Version 4.1 included loadable crash dump support for all RSX–11M–PLUS operating systems.

  Loadable XDT support enables you to load XDT when you want to use it for debugging. System performance is improved when XDT is not part of the system. Features such as instruction decoding and automated searching of symbol addresses listed in the Executive map are included in the loadable version of XDT. Refer to the *RSX–11M–PLUS and Micro/RSX Crash Dump Analyzer Reference Manual* for more information on loadable XDT support.

2.7 RSX–11M–PLUS and Micro/RSX Debugging Reference Manual

In the *RSX–11M–PLUS and Micro/RSX Debugging Reference Manual*, Section 1.2.3.3, please remove the second paragraph. For information on how to write to supervisor-mode libraries, refer to the *RSX–11M–PLUS Release Notes Version 4.2*, Section 1.17.1.

2.8 RSX–11M–PLUS and Micro/RSX Error Logging Manual

Please make the following corrections to the *RSX–11M–PLUS and Micro/RSX Error Logging Manual*:

- In Table 2–2, page 2–8, the following change must be made to the device entry for the control file module ETSV05:

  TSV05 /TK25 ETSV05

- In Section 3.3.3.4, the /HISTORY qualifier does not always work as described in the following text:

  "RPT generates a summary report sorted by device error history. It displays the hard and soft error count and QIO count for every volume on each device."

  The use of a date/time range may make the history summary more prone to error. The history summary information within the Report Generator (RPT) is valid only when there is a MOUNT (or RESET) followed by device activity followed by a DISMOUNT (or RESET) with no activity outside that interval. It is likely that devices that are mounted before Error Logging is activated

and that remain mounted for the duration, or are left mounted when Error Logging is turned off, will not be properly displayed in the history summary. No record will be created for those devices in the history summary database even if they reported errors during the selection range. This occurs because those devices were not followed by a RESET record within the specified date/time range to force updates of the counts. The QIO counts may also be wrong because they were not updated after the last MOUNT record.

- In Example A–1, page A–4, the following change must be made to the list of acceptable device names:

```
;       TSV05 or TK25
```

2.9 RSX–11M–PLUS and Micro/RSX Executive Reference Manual

Please make the following corrections to the *RSX–11M–PLUS and Micro/RSX Executive Reference Manual*:

- In Section 2.1, page 2–1, the tenth bullet in the list is incorrect. The following is the correct text:

  The execution of the round-robin scheduling algorithm at the end of a round-robin scheduling interval if the eligibility of the current task has changed.

- In Section 3.7.2, page 3–21, the last sentence in the paragraph is incorrect. The following is the correct text:

  R0 is returned as the status (IS.SUC or IE.ALG). If a length change was requested (for example, if the high bit of R0 was set) and the new length was set by default (for example, R2 was set to 0), R2 is returned as the new length of the mapping. In all cases, R2 and R3 are destroyed by the fast-map call.

- Please add the following descriptions of the new high-level language calls to the GIN directives to Chapter 5 of the *RSX–11M–PLUS and Micro/RSX Executive Reference Manual*. The following GIN directives may be called:

  - CALL GINABO
  - CALL GINAPR
  - CALL GINDEF
  - CALL GINDEV
  - CALL GINDVJ
  - CALL GINFMK
  - CALL GINGAS
  - CALL GINQMC
  - CALL GINREN
  - CALL GINSPR
  - CALL GINTSK
  - CALL GINUAB
  - CALL GINUIC
  - CALL GINUPD

- CALL GINVEC

2.9.1 The GINABO Call

CALL GINABO (buf , siz [,ids])

buf
A buffer area containing the name of the device for the aborts.

siz
An integer containing the size of the buffer in words. If the buffer is declared as an INT*2 array, then the size is equal to the number of elements in the array.

ids
An integer to receive the directive status.

See the description of the GI.ABO subfunction of the GIN directive in the *RSX–11M–PLUS and Micro/RSX Executive Reference Manual* for details on the buffer format and use of this directive.

2.9.2 The GINAPR Call

CALL GINAPR (buf , siz [,ids])

buf
A buffer area for the return of the information.

siz
An integer containing the size of the buffer in words. If the buffer is declared as an INT*2 array, then the size is equal to the number of elements in the array.

ids
An integer to receive the directive status.

See the description of the GI.APR subfunction of the GIN directive in the *RSX–11M–PLUS and Micro/RSX Executive Reference Manual* for details on the buffer format and use of this directive.

2.9.3 The GINDEF Call

CALL GINDEF (uic [,ids])

uic
An integer representing an user identification code (UIC).

ids
An integer to receive the directive status.

See the description of the GI.DEF subfunction of the GIN directive in the *RSX–11M–PLUS and Micro/RSX Executive Reference Manual* for details on the buffer format and use of this directive.

2.9.4 The GINDEV Call

CALL GINDEV (buf , siz , devnam , unit [,ids])

buf
A buffer area for the return of the information.

siz
An integer containing the size of the buffer in words. If the buffer is declared as an INT*2 array, then the size is equal to the number of elements in the array.

devnam
A two-character ASCII device mnemonic.

unit
An integer containing the device unit number.

ids
An integer to receive the directive status.

See the description of the GI.DEV subfunction of the GIN directive in the *RSX–11M–PLUS and Micro/RSX Executive Reference Manual* for details on the buffer format and use of this directive.

2.9.5 The GINDVJ Call

```
CALL GINDVJ    ( buf , siz , devnam , unit [,ids] )
```

buf
A buffer area for the return of the information.

siz
An integer containing the size of the buffer in words. If the buffer is declared as an INT*2 array, then the size is equal to the number of elements in the array.

devnam
A two-character ASCII device mnemonic.

unit
An integer containing the device unit number.

ids
An integer to receive the directive status.

See the description of the GI.DVJ subfunction of the GIN directive in Section 1.3.4.1 and also, later in this section in this document, for details on the buffer format and use of this directive.

2.9.6 The GINFMK Call

```
CALL GINFMK    ( buf , siz [,ids] )
```

buf
A buffer area for the return of the information.

siz
An integer containing the size of the buffer in words. If the buffer is declared as an INT*2 array, then the size is equal to the number of elements in the array.

ids
An integer to receive the directive status.

See the description of the GI.FMK subfunction of the GIN directive in the *RSX–11M–PLUS and Micro/RSX Executive Reference Manual* for details on the buffer format and use of this directive.

2.9.7 The GINGAS Call

```
CALL GINGAS  ( buf , siz , devnam , unit , udev , unum [,ids] )
```

buf
A buffer area for the return of the information.

siz
An integer containing the size of the buffer in words. If the buffer is declared as an INT*2 array, then the size is equal to the number of elements in the array.

devnam
A two-character ASCII device mnemonic.

unit
An integer containing the device unit number.

udev
unum
A device name and unit number pair, as described for devnam and unit. This device name is used to identify a terminal for the interpretation of logical names.

ids
An integer to receive the directive status.

See the description of the GI.GAS subfunction of the GIN directive in the *RSX–11M–PLUS and Micro/RSX Executive Reference Manual* for details on the buffer format and use of this directive.

2.9.8 The GINQMC Call

```
CALL GINQMC  ( buf , siz [,ids] )
```

buf
A buffer area containing the command line.

siz
An integer containing the size of the buffer in words. If the buffer is declared as an INT*2 array, then the size is equal to the number of elements in the array.

ids
An integer to receive the directive status.

See the description of the GI.QMC subfunction of the GIN directive in the *RSX–11M–PLUS and Micro/RSX Executive Reference Manual* for details on the buffer format and use of this directive.

2.9.9 The GINREN Call

```
CALL GINREN  ( nam1 , nam2 [,ids] )
```

nam1
nam2
Two integers that represent, respectively, the first and second half of a task name expressed in RAD50.

ids
An integer to receive the directive status.

See the description of the GI.REN subfunction of the GIN directive in the *RSX–11M–PLUS and Micro/RSX Executive Reference Manual* for details on the buffer format and use of this directive.

2.9.10 The GINSPR Call

CALL GINSPR (flag [,ids])

flag
An integer that is interpreted as true if not zero; false if zero.

ids
An integer to receive the directive status.

See the description of the GI.SPR subfunction of the GIN directive in the *RSX–11M–PLUS and Micro/RSX Executive Reference Manual* for details on the buffer format and use of this directive.

2.9.11 The GINTSK Call

CALL GINTSK (buf , siz , nam1 , nam2 [,ids])

buf
A buffer area for the return of the information.

siz
An integer containing the size of the buffer in words. If the buffer is declared as an INT*2 array, then the size is equal to the number of elements in the array.

nam1
nam2
Two integers that represent, respectively, the first and second half of a task name expressed in RAD50.

ids
An integer to receive the directive status.

See the description of the GI.TSK subfunction of the GIN directive in the *RSX–11M–PLUS and Micro/RSX Executive Reference Manual* for details on the buffer format and use of this directive.

2.9.12 The GINUAB Call

CALL GINUAB (buf , siz , devnam , unit [,ids])

buf
A buffer area for the return of the information.

siz
An integer containing the size of the buffer in words. If the buffer is declared as an INT*2 array, then the size is equal to the number of elements in the array.

devnam
A two-character ASCII device mnemonic.

unit
An integer containing the device unit number.

ids
An integer to receive the directive status.

See the description of the GI.UAB subfunction of the GIN directive in the *RSX–11M–PLUS and Micro/RSX Executive Reference Manual* for details on the buffer format and use of this directive.

2.9.13 The GINUIC Call

CALL GINUIC (buf , siz [,ids])

buf
A buffer area for the return of the information.

siz
An integer containing the size of the buffer in words. If the buffer is declared as an INT*2 array, then the size is equal to the number of elements in the array.

ids
An integer to receive the directive status.

See the description of the GI.UIC subfunction of the GIN directive in the *RSX–11M–PLUS and Micro/RSX Executive Reference Manual* for details on the buffer format and use of this directive.

2.9.14 The GINUPD Call

CALL GINUPD (buf , siz [,ids])

buf
A buffer area for the return of the information.

siz
An integer containing the size of the buffer in words. If the buffer is declared as an INT*2 array, then the size is equal to the number of elements in the array.

ids
An integer to receive the directive status.

See the description of the GI.UPD subfunction of the GIN directive in the *RSX–11M–PLUS and Micro/RSX Executive Reference Manual* for details on the buffer format and use of this directive.

2.9.15 The GINVEC Call

CALL GINVEC (buf , siz [,ids])

buf
A buffer area for the return of the information.

siz
An integer containing the size of the buffer in words. If the buffer is declared as an INT*2 array, then the size is equal to the number of elements in the array.

ids
An integer to receive the directive status.

See the description of the GI.VEC subfunction of the GIN directive in Section 1.3.4.2 in this document for details on the buffer format and use of this directive.

2.9.16 Other Corrections to the RSX–11M–PLUS and Micro/RSX Executive Reference Manual

This section contains additional corrections to the *RSX–11M–PLUS and Micro/RSX Executive Reference Manual.*

- In Section 5.13, page 5–40, the macro call for CMKT$ is incorrect. The correct macro call is shown next.

  Macro Call:

  ```
  CMKT$   [[efn],[ast]]
  ```

  where:

efn	=	Event flag number
ast	=	Mark time AST address

  Macro Expansion:

  ```
  CMKT$   52.,MRKAST
  .BYTE   27.,3         ;CMKT$ MACRO DIC, DPB SIZE = 3 WORDS
  .WORD   52.           ;EVENT FLAG NUMBER 52
  .WORD   MRKAST        ;ADDRESS OF MARK TIME REQUEST AST ROUTINE
  ```

  Note

  The above example will cancel only the Mark Time requests that were specified with efn 52 or the AST address MRKAST. If no ast or efn parameters are specified, all Mark Time requests issued by the task are canceled and the DPB size equals 1.

- In Section 5.18, page 5–53, please add the following text to the description of the CRRG$ directive:

  A facility has been added to the Create Regions (CRRG$) directive.

  If RS.RES is set in the region definition block prior to executing the CRRG$ directive, the region will be created only if it can immediately be made resident (without necessitating checkpointing of other resident regions). This includes checking to see that it will fit in the current available holes. If space is not currently available, the directive will fail and return a status of IE.ALG (– 84.)

This allows an application to create a shared region without the complication that the region may be permanently or semi-permanently locked out of memory by other regions, even though its size can be accommodated in the partition requested.

- In Section 5.35, page 5–91, please delete the second bullet in the list. Add the following text to the description of the EXTK$ directive:

  Previously, the EXTK$ and EXTM$ extend task directives would not attempt a task extend if the task had memory resident overlays. This restriction has been removed, as described in the following paragraphs.

  The restriction is overridden if the task is built with the new Task Builder /EX task file switch. Using this switch when performing a task build will permit extension of the task when a valid EXTK$ or EXTM$ directive is issued. Normal physical allocation of memory positions the root first, then any memory resident overlays, followed by any disk resident overlays. When the /EX switch is used, the Task Builder modifies the physical allocation of memory so that the root of the task is positioned after all memory resident overlays, but before any disk resident overlays, thus allowing the task to be extended, since any extension is done to the root. For ID tasks, only the D-space root (along with the D-space stack and header) is shifted, since only the D-space root is extended.

  Virtual memory is also reallocated to facilitate run-time task extensions over APR boundaries as follows: after all allocation of APRs has been carried out (for the task, shared regions, VSECTS, etc.), the task overlays are shifted up as high as possible, to allow for run-time task extension. This shift is done in one contiguous block, and applies to both memory resident and disk overlays. Note that a disk overlay can occur only up-tree from a memory resident overlay, and will share the same window as the memory resident overlay.

  As an example, suppose a task is built that normally uses APR 0 for the root, APRs 1 and 2 for memory resident (and possibly disk resident) overlays, and APR 7 for a library. If the /EX switch is used, APR 0 will continue to be allocated for the root, and APR 7 will be allocated for the library as before. However, the task overlays will now be allocated to APRs 5 and 6, leaving room for task extension into APRs 1 through 4.

- In Section 5.36, page 5–95, additional symbols have been added to the FEAT$ directive. Please add the following symbols to Table 5–1, System Feature Symbols:

Table 5–1: System Feature Symbols

Symbol	Value	Meaning
FE$NCT	63.	System has NCT support
FE$LSD	64.	System has LUT scan in LOADR disabled
FE$PC3	65.	System supports Professional 3xx series personal computers
FE$DFB	66.	System supports deferred binding
FE$RTB	67.	Run time binding
FE$ODB	68.	Deferred binding is overridable
FE$XDJ	69.	XDT is doing I/O via TPRs on KXJ
FE$NSY	70.	No local system disk

Symbol	Value	Meaning
FE$NCO	71.	No local console
FE$RTK	72.	Remote task services
FE$RDR	73.	Remote directory storage
FE$RLG	74.	Remote logical support
FE$LDR	75.	Remote load/overlays
FE$VTL	76.	VT logins disabled
FE$ANT	77.	Automatic network startup by save
FE$NRT	78.	Network remote system
FE$EXE	79.	Task file names default to .EXE
FE$CMO	80.	Resident overlays default to 512. byte alignment

- In Section 5.64, page 5–188, the macro expansion for the RLON$ and RLOG$ directives is incorrect. The correct macro expansion is shown next.

```
RLON$ MOD,TBMSK,STATUS,LNS,LNSSZ,ENS,ENSSZ,RSIZE,RTBMOD
.BYTE     207.,10.      ;RLON$ MACRO DIC, DPB SIZE = 10(10) WORDS
.BYTE     14.           ;SUBFUNCTION VALUE (RLOG$ = 10(10))

.BYTE     MOD           ;LOGICAL NAME MODIFIER
.WORD     TBMSK         ;LOGICAL NAME TABLE INHIBIT MASK

.WORD     LNS           ;LOGICAL NAME STRING ARRAY
.WORD     LNSSZ         ;SIZE (IN BYTES) OF LOGICAL NAME STRING

.WORD     ENS           ;RETURNED EQUIVALENCE NAME ARRAY
.WORD     ENSSZ         ;SIZE (IN BYTES) OF EQUIVALENCE NAME

.WORD     RSIZE         ;LOCATION OF SIZE FOR RETURNED EQUIVALENCE NAME

.WORD     RTBMOD        ;LOCATION OF LOGICAL TABLE NUMBER (LOWER BYTE) AND
                        ;MODIFIER VALUE OF LOCATED LOGICAL NAME (HIGHER BYTE)

.WORD     STATUS        ;LOCATION OF LOGICAL NAME STATUS
```

- In Section 5.98, page 5–281, the reference to the CINT$ section in the second sentence in the first paragraph is incorrect. Replace that sentence with the following:

  For information on mapping the subroutines, see Note 2 for the description of the CINT$ (Connect to Interrupt) directive.

- In Section 5.98, page 5–282, the calculation of the addresses in Notes 3 and 4 is incorrect. The calculation should be as follows:

  n = n + 120000 + (base & 177700)

- In Section 5.99, page 5–284, an additional symbol has been added to the TFEA$ directive. Please add the following symbol to Table 5–2, Task Feature Symbols:

Table 5–2: Task Feature Symbols

Symbol	Value	Meaning
T4$DFB	41.	Task has deferred binding

- In Section 5.100, page 5–288, the macro expansion for the TLON$ and TLOG$ directives is incorrect. The correct macro expansion is shown next.

```
TLONS MOD,TBMSK,STATUS,LNS,LNSSZ,ENS,ENSSZ,RSIZE,RTBMOD
.BYTE     207.,10.      ;TLON$ MACRO DIC, DPB SIZE = 10(10) WORDS
.BYTE     13.           ;SUBFUNCTION VALUE (TLOG$ = 9(10))

.BYTE     MOD           ;LOGICAL NAME MODIFIER
.WORD     TBMSK         ;LOGICAL NAME TABLE INHIBIT MASK

.WORD     LNS           ;LOGICAL NAME STRING ARRAY
.WORD     LNSSZ         ;SIZE (IN BYTES) OF LOGICAL NAME STRING

.WORD     ENS           ;RETURNED EQUIVALENCE NAME ARRAY
.WORD     ENSSZ         ;SIZE (IN BYTES) OF EQUIVALENCE NAME

.WORD     RSIZE         ;LOCATION OF SIZE FOR RETURNED EQUIVALENCE NAME

.WORD     RTBMOD        ;LOCATION OF LOGICAL TABLE NUMBER (LOWER BYTE) AND
                        ;MODIFIER VALUE OF LOCATED LOGICAL NAME (HIGHER BYTE)

.WORD     STATUS        ;LOCATION OF LOGICAL NAME STATUS
```

- In Sections 5.104, 5.105, and 5.106, the definition of the FORTRAN subroutine call parameter bufadr is incorrect. The correct definition is as follows:

  bufadr
  Specifies an array containing data to be sent (must be word aligned (INTEGER*2)).

  Please note that the definition of the macro call parameter bufadr is correct and should not be changed.

- The following sections describe an Executive modification, list a restriction, and provide supplementary information.

  Modification to the IOSUB.MAC Module

  A modification was made to the IOSUB.MAC module to prohibit both UNIBUS Mapping Registers (UMRs) and error logging processing for IO.ATT and IO.DET functions.

  Shared Regions Restriction

  You cannot build a shared region that contains I- and D-space. Shared regions that are built with the /ID switch will not operate correctly. As stated in Section 10.19 of the *RSX–11M–PLUS and Micro/RSX Task Builder Manual*, you cannot use the /-HD switch and /ID switch in the same build.

  Extend Task Directive

  The Extend Task directive with the Active Page Register (APR) protection mask enhancement instructs the system to modify the size of the issuing task by a positive or negative increment of 32-word blocks. If the directive does not specify an increment value or if it specifies an increment value of zero, the Executive makes the issuing task's size equal to its installed size. The issuing task must be running in a system-controlled partition and cannot have any outstanding I/O when it issues the directive. The task must also be checkpointable to increase its size; if necessary, the Executive checkpoints the task and then returns the task to memory with its size modified as directed.

  In a system that supports the memory management directives, the Executive does not change any current mapping assignments if the task has memory-resident overlays unless the task was built with the Task Builder /EX switch. (Please see the additional text for Section 5.35 explaining enhancements to the EXTK$ directive. This text appears earlier in this section.) However, if the task does not have memory-resident overlays, the Executive attempts to modify, by the specified number of 32-word blocks, the mapping of the task to its task region.

If the issuing task is checkpointable but has no preallocated checkpoint space available, a positive increment may require dynamic memory and extra space in a checkpoint file sufficient to contain the task.

The Extend Task directive with the APR protection mask enhancement enables you to specify a mask parameter that will prevent the extension from changing the default mapping of the task's APR mapping. This enables you to extend the data-space window of a task without changing the mapping of APRs that default to overmapping a library in instruction space.

There are several constraints on the size to which a task can extend itself using the Extend Task directive enhancement. These constraints are as follows:

- No task can extend itself beyond the maximum size set by the MCR command SET /MAXEXT or the DCL command SET EXTENSION_LIMIT or the size of the partition in which it is running. (See the *RSX–11M–PLUS MCR Operations Manual* or the *RSX–11M–PLUS Command Language Manual*.)
- A task that does not have memory-resident overlays cannot extend itself beyond 32K minus 32 words.
- A task that has preallocated checkpoint space in its task image file cannot extend itself beyond its installed size.
- A task that has memory-resident overlays cannot reduce its size below the highest window in the task partition.

Format

EXTM$ [inc],mask

Parameters

inc

Specifies a positive or negative number equal to the number of 32-word blocks by which the task size is to be extended or reduced.

mask

Specifies a mask of APRs to be protected. Bit 0 represents APR0 and bit 1 represents APR1. For example, if you specify 340, then APRs 7, 6, and 5 are protected.

Macro Expansion

```
EXTM$   40,340
.BYTE   89.,3           ;EXTK$ MACRO DIC, DPB SIZE = 3 WORDS
.WORD   40              ;EXTEND INCREMENT, 40(8) BLOCKS (1K WORDS)
.WORD   340             ;APR PROTECTION MASK
```

Local Symbol Definition

E.XTIN	Extend increment (2)

DSW Return Codes

IS.SUC	Successful completion.
IE.UPN	Insufficient dynamic memory or insufficient space in a checkpoint file.
IE.ITS	The issuing task is not running in a system-controlled partition.

IE.ALG	The issuing task attempted to reduce its size to less than the size of its task header; the task tried to increase its size beyond 32K words or beyond the maximum set by the MCR command SET /MAXEXT or DCL command SET EXTENSION_LIMIT; the task tried to increase its size to the extent that one virtual address window would overlap another; the task has memory-resident overlays and it attempted to reduce its size below the highest window mapped to the task partition; or the extend would unmap a protected APR.
IE.RSU	Other tasks are attached to this task partition.
IE.IOP	I/O is in progress for this task partition.
IE.CKP	The issuing task is not checkpointable and specified a positive integer.
IE.NSW	The task attempted to extend itself to larger than the installed size (when checkpoint space is allocated in the task).
IE.ADP	Part of the Directive Parameter Block (DPB) is out of the issuing task's address space.
IE.SDP	Directive Identification Code (DIC) or DPB size is invalid.

General Information Directive

The general information directive function Get Device Information Junior (GI.DVJ) returns information about a particular device. The device for which information is returned is determined by first performing a logical assignment (if required) and then following any redirection assignments. Device assignments are checked if the high bit in the flag's byte is clear; otherwise, no check of device assignments is made.

Format

GIN$ GI.DVJ, buf, siz, dev, unt

Parameters

GI.DVJ

Specifies GIN$ function code (18).

buf

Specifies address of buffer to receive the unit information.

siz

Specifies size of buffer. Buffer size is 1 word.

dev

Specifies device name (if blank, use task's TI:).

unt

Specifies device unit number (if high bit clear, follow assignments).

Buffer Format

– 1	Device is not a mass-storage and mountable device.
– 2	Device is not mounted for issuer.
Bit 0	READ access is prohibited.
Bit 1	WRITE access is prohibited.
Bit 2	CREATE access is prohibited.
Bit 3	DELETE access is prohibited.
Bit 4	Device is mounted public.

Bit 5	Device is mounted private (allocated).
Bit 6	Device is mounted foreign.
Bit 7	Foreign device has ACP.

Macro Expansion

```
GIN$    GI.DEV, DVBUF, DVSIZ, "TT, 1
.BYTE   169.,6
.WORD   GI.DEV
.WORD   DVBUF
.WORD   DVSIZ
.WORD   "TT
.WORD   1
```

DSW Return Codes

IS.SUC	Successful completion.
IE.ADP	Part of Directive Parameter Block (DPB) is out of task's address space.
IE.IDU	The specified device does not exist, or device is a virtual terminal and issuing task is not parent or offspring.
IE.SDP	Invalid function code or the DPB size is invalid.

Notes

1. If the task has the "slave" attribute, logical assignments are not checked regardless of the setting of the high bit in the fourth parameter word.
2. If bit 4 and bit 5 are off, the device may be mounted semiprivate; that is, the device may be mounted by the user but not allocated.

Creating Region Names

RSX–11M–PLUS allows a region to create a region with the name GEN. In addition, RSX–11M–PLUS allows regions in region GEN to be created even if a common named GEN is installed in any main partition.

The Get Partition Parameters directive (GPRT$) uses the subroutine $SRNAM, which is used by many parts of the executive to look up region and partition names. GPRT$ looks at the main partition list first; then it looks at the common block directory.

The TKB command shown next performs a GPRT$ directive for the main partition GEN.

```
>TKB task=object
```

Any application that uses GPRT$ to get the parameters of a named common region gets the parameters of a main partition if the name of the common region is not unique with respect to the main partition name. All applications that use $SRNAM must be changed to attach to the region by name and then to use the Get Region Parameters directive (GREG$) to obtain the region parameters.

2.10 RSX–11M–PLUS and Micro/RSX Guide to Writing an I/O Driver

Please make the following corrections to the *RSX–11M–PLUS and Micro/RSX Guide to Writing an I/O Driver:*

- Section 1.5.5, page 1–21, illustrates how to create an executive entry point vector table. It does not indicate that there are any differences between creating a vector table for a driver and creating a vector table for a privileged task. However, the length is calculated differently in each case. Use the

following statement when determining the length of the vector table for a privileged task:

```
EXEVCL=<<.-EXEVEC>/2>
```

Use the following statement when determining the length of the vector table for a driver:

```
EXEVCL=<<<.-EXEVEC>/2>-1>
```

- In Section 1.5.6, page 1–22, the table entitled Callable Routine for Converting Executive References in a Driver is incorrect. The correct callable routine is as follows:

```
MOV     KINAR6,-(SP)     ; SAVE KINAR6
MOV     @#112,R0         ; GET ADDRESS OF TABLE OF ENTRIES
MOV     (R0),R0          ; GET ADDRESS OF APR BIAS -- FIRST WORD IN TABLE
MOV     (R0),KINAR6      ; MAP COMMON THROUGH I-SPACE APR 6
MOV     #EXEVEC,R3       ; POINT TO VECTOR
MOV     #EXEVCL,R2       ; SPECIFY LENGTH OF VECTOR
CALL    @#140004         ; TRANSLATE THE VECTOR
MOV     (SP)+,KINAR6     ; RESTORE KINAR6
```

- Please add the following information to Section 2.4 of the manual:

  Error logging support has been added for RA70 and RA90 disk drives (in SA550 and SA650 arrays only).

- Please add the following information to Section 4.5.12 of the manual:

  The Executive's support for volume valid has been changed to allow the setting of the hardware and software volume valid bit (a QIO with a function code of IO.STC and a second parameter value of VV$SET) on any volume that is mounted foreign without an ACP. Previously, only hardware volume valid could be set on volumes where software volume valid was already set.

  This change is primarily intended to ease the handling of multivolume data sets on foreign mounted devices, by allowing an application (such as the RSX Backup and Restore Utility, BRU) to clear hardware volume valid prior to requesting the change of volumes, and then to reset it after the volume has been changed.

- Please add the following information to Section 7.4.16 of the manual:

  If you build a vectored driver for the system macro call GTPKT$, the following symbol must be defined in the driver code:

```
VC$xx = 0
```

  The parameter xx represents the 2-character device mnemonic.

- Please add the following information to Section 7.4.19 of the manual:

  If you build a vectored driver for the system macro call INTSV$, the following symbol must be defined in the driver code:

```
VC$xx = 0
```

  The parameter xx represents the 2-character device mnemonic.

- Please make the following correction to Section 8.3, pages 8–24 and 8–25:

The example of the sample driver BMDRV.MAC is incorrect because the offset I.PRM+16 must be cleared if you cannot do buffered I/O. Replace the section of code in the example with the following code:

```
;       ************************************************************
;       *                                                          *
;       *       CONVERT TO BUFFERED I/O REQUEST                    *
;       *                                                          *
;       ************************************************************
        MOV     R5,R3           ; COPY I/O PACKET ADDRESS BACK
;       +----------------------------------------------------------+
;       |                                                          |
;       |       THE INPUT PARAMETERS FOR $INIBF ARE:               |
;       |                                                          |
;       |       R3 = ADDRESS OF THE I/O PACKET TO BUFFER           |
;       |                                                          |
;       |       NO OUTPUT PARAMETERS.                              |
;       |                                                          |
;       +----------------------------------------------------------+
        CALL    $INIBF          ; INITIALIZE BUFFERED I/O

        BR      45$             ; SKIP CLEARING OF BUFFER ADDRESS

40$:    CLR     I.PRM+16(R3)    ; INDICATE NO BUFFERED I/O
;       ************************************************************
;       *                                                          *
;       *       QUEUE THE CLOCK BLOCK                              *
;       *                                                          *
;       ************************************************************

45$:    MOV     I.PRM+14(R3),R0 ; GET ADDRESS OF CLOCK BLOCK
```

2.11 RSX–11M–PLUS and Micro/RSX I/O Drivers Reference Manual

Please make the following corrections to the *RSX–11M–PLUS and Micro/RSX I/O Drivers Reference Manual*:

The format shown in Section 2.4.3 on page 2–23 is incorrect. The correct format is:

QIO$C IO.ATA [!TF.ESQ / !TF.NOT / !TF.XCC] ,lun,[efn],[pri],[isb],,<[ast1],[parameter2],[ast2]>

The values listed for the RX33 drive in Table 4–1, page 4–2 are incorrect. The correct characteristics for the RX33 drive are as follows:

Drive	Revolutions Per Minute	Sectors	Tracks	Cylinders	Bytes/Drive	Decimal Blocks
RX33	3600	15	2	80	1,228,800	2400

Information on message-oriented communication drivers was inadvertently removed from the *RSX–11M–PLUS and Micro/RSX I/O Drivers Reference Manual*. This information is applicable only to RSX–11M–PLUS operating systems. Refer to Appendix B, Section B.3.10 for information on message-oriented I/O function codes.

The information shown next should be added to the manual.

Introduction to Message-Oriented Communication Drivers

Message-oriented communication line interfaces usually link two separate but complementary computer systems. One system must serve as the transmitting device and the other as the receiving device. Message-oriented communication line interfaces are used to transfer large blocks of data.

While character-oriented interfaces can only be accessed indirectly through the terminal driver, the DMC11 and DUP11 synchronous line interfaces allow I/O requests to be queued directly for them. These devices have drivers of their own and can be accessed by means of logical device names. You can use these names in assigning Logical Unit Numbers (LUNs) with the Assign LUN system directive at task build or with the MCR command REASSIGN.

DMC11 Synchronous Line Interface

The DMC11 synchronous line interface provides a direct memory access interface between two PDP–11 computer systems using the DDCMP line protocol, thus delivering high throughput and reliability while simplifying programming. The DMC11 supports nonprocessor request (NPR) data transfers of up to 8K words at rates of 1,000,000 baud for local operation (over coaxial cable) and 19,200 baud for remote operation (using modems). Both full- and half-duplex modes are supported. The DMC11 synchronous line interface also implements remote load detect, allowing it to reinitialize a halted computer system.

DUP11 Synchronous Line Interface

The DUP11 synchronous line interface is a single-line communications device that provides a program-controlled interface between the PDP–11 and a serial synchronous line. The PDP–11 can be interfaced with a high-speed line to perform remote batch processing, remote data collection, and remote concentration applications. Modem control is a standard feature of the DUP11 and allows using the device in a switched or dedicated configuration. The DUP11 transmits data at a maximum rate of 9600 baud; this rate is limited by modem and data set interface level converters.

The DUP11 can be programmed to accept any sync character that you define. The DUP11 incorporates hardware to perform a cyclic redundancy check (CRC).

Get LUN Information Macro

Word 2 of the buffer filled by the Get LUN Information system directive (the first characteristics word) contains the following information for message-oriented communication interfaces. A bit setting of 1 indicates that the described characteristic is true for the interfaces described in this section.

Bit	Setting	Meaning
0	0	Record-oriented device
1	0	Carriage-control device
2	0	Terminal device
3	0	File-structured device
4	0	Single-directory device
5	0	Sequential device
6	0	Mass-storage device
7	0	User-mode diagnostics supported

Bit	Setting	Meaning
8	0	Device supports 22-bit direct addressing
9	0	Unit software write locked
10	0	Input spooled device
11	0	Output spooled device
12	0	Pseudo device
13	1	Device mountable as a communications channel
14	0	Device mountable as a Files–11 volume
15	1	Device mountable

Words 3 and 4 are undefined, and word 5 has a special meaning for the DUP11 interface. Byte 0 of word 5 contains the number of sync characters to be transmitted before a synching message (for example, after line turn-around in a half-duplex operation), and byte 1 is a sync counter.

QIO$ Macro

The following sections summarize the standard and device-specific functions of the QIO$ macro that are valid for the communication interfaces.

Standard QIO$ Functions

The standard functions of the QIO$ macro that are valid for the communication devices are shown next.

Format	Function
QIO$C IO.ATT, . . .	Attach device [1]
QIO$C IO.DET, . . .	Detach device
QIO$C IO.KIL, . . .	Cancel I/O requests
QIO$C IO.RLB, . . . ,<stadd,size>	Read logical block (stripping sync)
QIO$C IO.WLB, . . . ,<stadd,size>	Write logical block (preceded by syncs)

[1]Only unmounted channels may be attached. An attempt to attach a mounted channel results in an IE.PRI status return in the I/O status doubleword.

stadd

The starting address of the data buffer (may be on a byte boundary).

size

The data buffer size in bytes (must be greater than 0).

Device-Specific QIO$ Functions

The specific functions of the QIO$ macro that are valid for the communication line interfaces are shown next.

Format	Function
QIO$C IO.FDX	Set device to full-duplex mode
QIO$C IO.HDX, . . . ,<stat,mode>	Set device to half-duplex mode
QIO$C IO.INL, . . .	Initialize device and set device characteristics

Format	Function
QIO$C IO.RNS, . . . ,<stadd,size>	Read logical block, without stripping sync characters (transparent mode); for DMC11, treated like IO.RLB. Not supported on DUP11.
QIO$C IO.SYN, . . . ,<syn>	Specify sync character; not applicable to DMC11
QIO$C IO.TRM, . . .	Terminate communication, disconnecting from physical channel
QIO$C IO.WNS, . . . ,<stadd,size>	Write logical block without preceding sync characters (transparent mode); for DMC11, treated like IO.WLB

stat
Specifies the station assignment (primary or secondary).

mode
Specifies the transmission mode (normal or maintenance).

stadd
Specifies the starting address of the data buffer (may be on a byte boundary).

size
Specifies the data buffer size in bytes (must be greater than 0).

syn
Specifies the sync character, expressed as an octal value.

The device-specific QIO$ functions are described in the following sections.

IO.FDX Function

The QIO$ function IO.FDX sets the mode on a DUP11 or DMC11 unit to full duplex. The IO.FDX function code can be combined (ORed) with the IO.SYN function code, if desired, to set the operational characteristics of the physical device unit.

IO.HDX Function

The QIO$ function IO.HDX sets the mode on a DUP11 or DMC11 unit to half duplex. The IO.HDX function code can be combined (ORed together) with the IO.SYN function code, if desired, to set the operational characteristics of the physical device unit.

Setting half-duplex mode on the DMC11 also involves setting the station assignment (primary/secondary) and may include selecting maintenance mode (MOP) as opposed to normal mode. The station assignment is included in the optional QIO$ parameter p1. A 0 indicates a primary station and a nonzero indicates a secondary station. The DMC11 works properly if both ends are primary stations or if there is one primary and one secondary station. It does not work if both ends are secondary stations. The optional QIO$ parameter p2 selects the mode. A 0 selects normal mode and a nonzero selects MOP mode. A DMC11 in MOP mode cannot communicate with a DMC11 in normal mode.

IO.INL and IO.TRM Functions

The QIO$ functions IO.INL and IO.TRM have the same function code but different modifier bits.

IO.INL initializes a physical device unit for use as a communications link. It turns the device on line, sets device characteristics, and ensures that the appropriate data terminal is ready.

IO.TRM disconnects the device. If the device has a dial-up interface, it also hangs up the line.

IO.RNS Function

The IO.RNS QIO$ function reads a logical block of data without stripping the sync characters that may precede the data.

IO.RLB is a similar function, which is nontransparent, in that it causes the sync characters that precede the data message to be stripped. Use IO.RLB at the start of a segmented data request, in which the block might have the following layout:

S
Specifies a sync character.

H
Specifies a header character.

CS
Specifies a validity check character.

You must strip sync characters from the beginning of a data block in this way. Stripping only at the beginning of a read operation allows a later character that happens to have the same binary value as a sync character to be read without stripping. Use IO.RLB to read a logical block with leading sync characters stripped; use IO.RNS to read the block without stripping leading sync characters.

IO.SYN Function

This QIO$ function allows the programmer to specify the sync character to be recognized when an IO.RLB or IO.WLB function is performed. IO.SYN can be combined (ORed together) with IO.HDX or with IO.FDX to set the characteristics of the physical device unit.

IO.WNS Function

This QIO$ function causes a logical block to be written with no preceding sync characters. To ensure that the two systems involved in a communication are synchronized, two or more sync characters are transmitted by one system and received by the other before any other message can be sent.

Use IO.WLB to write a block of data preceded by sync characters.

Use IO.WNS to perform a block transfer without sending sync characters first.

Programming Hints

The following sections contain important information about programming the message-oriented communication interfaces.

Transmission Validation

Because there is no way for the transmitting device to verify that the data block has successfully arrived at the receiving device unless the receiver responds, the transmitter assumes that any message that is clocked out on the line (without line or device outage) has been successfully transmitted. As soon as the receiver is able to satisfy a read request, it returns a successful status code (IS.SUC) in the I/O status block. Of course, only the task receiving the message can determine whether the message has actually been transmitted accurately.

The receiving device should be ready to receive data (with a read request) at the time the transmission is sent.

Redundancy Checking

By the nature of message-oriented communications, only the task that receives a communication can determine whether the message was received successfully. The transmitter simply transfers data, without validation of any kind. It is therefore the responsibility of the communicating tasks that use the device to check the accuracy of the transmission. A simple validity check is a checksum-type longitudinal redundancy check. A better approach to validating data is the use of a cyclic redundancy check (CRC). A CRC can be computed in software or with a hardware device, such as the KG–11 communications arithmetic option.

The DUP11 incorporates hardware to compute a CRC.

Half-Duplex and Full-Duplex Considerations

Because there is a single I/O request queue, only one QIO$ request can be performed at a time. It is therefore not possible, through QIO$s, for a device to send and receive data at the same time. Also, because timeouts are not set for receive functions, a receive QIO$ is terminated only by receiving a message from the remote system, or by issuing an IO.KIL QIO$ for the device. Therefore, if no message is transmitted by the remote system, a receive does not terminate, and no further I/O can be performed on that device until the receive is killed by issuing an IO.KIL QIO$.

You can use both half-duplex and full-duplex lines with the DMC11 and DUP11. The mode is settable by using IO.FDX for full-duplex mode and IO.HDX for half-duplex mode. In half-duplex mode, the modem signal RTS (Request To Send) is cleared after each "transmit message." In full-duplex mode, this signal is always left on. Using full-duplex mode eliminates modem delays in transmission, but requires full-duplex hardware and communication links.

The DMC11 Driver maintains both transmit operations and receive operations separately in its own internal queues. Thus, it is a full-duplex driver. There is no limit on the number of outstanding I/O requests that can be active at any given time. The DMC11 hardware, however, allows a maximum of only seven transmit operations and seven receive operations to be active at any time. The driver gives the first seven transmit operations (or receive operations) directly to the DMC11 and queues the eighth and subsequent transmit operations (or receive operations) internally until the DMC11 acknowledges a successful I/O request. When running on a MicroPDP–11/70, the driver gives only two transmit operations (or receive operations) to the DMC11 because each request requires a UNIBUS mapping register (UMR). The DMC11 driver is assigned five UMRs: one for base table(s), two for active transmit operations, and two for active receive operations.

Low-Traffic Sync Character Considerations

If message traffic on a line is low, each message sent from a communications device should be preceded by a sync train. This enables the controller to resynchronize if a message is "broken" (that is, part or all of it is lost in transmission). Correspondingly, every message received by a communications device under low-traffic conditions, when messages are not contiguous (back-to-back), should be read with an IO.RLB (read, strip sync) function. This requires that the first character in the data message itself not have the binary value of the sync character.

Powerfail with DMC11

The DMC11 currently cannot recover after a power failure because the random-access memory (RAM) in its internal microprocessor is erased when power fails. Any I/O requests outstanding at the time of a power failure return the IE.ABO status. These requests must be reissued after initializing the DMC11 (IO.INL).

Importance of IO.INL

After the type of communication line has been determined, and after IO.SYN has specified the sync character, it is extremely important that IO.INL be issued before any transfers occur. This ensures that appropriate parameters are initialized and that the interface is properly conditioned. Note that IO.INL provides the only means of setting device characteristics, such as sync character. For this reason, you should always use IO.INL immediately prior to the first transfer over a newly activated link.

Tasks sending messages to the DMC11 should begin by terminating and reinitializing the device (IO.TRM, IO.INL). Note that this causes the error IE.CNR to be returned on any I/O outstanding on the other end of the line. IO.INL must be issued after each IO.KIL (which effectively kills the DMC11), after power fail, and upon receipt of any error code.

Programming Example

The following example illustrates the initialization, the setting of device parameters, and the transmission of a block of data on a message-oriented communication device.

```
.MCALL  ALUN$S,QIO$S
.       .
.       .
.       .

ALUN$S  1,"XP,0                                 ;USE LUN1 FOR DP11
QIO$S   IO.HDX!IO.SYN,<1,,,,,226>               ;SET DEVICE PARAMETERS
QIO$S   IO.INL,1                                ;PUT DEVICE ON LINE
QIO$S   IO.WLB,1,,,<TXSTS,TXAST,TXBUF,100>      ;SEND A BLOCK
.       .
.       .
.       .

TXAST:  CMPB    IS.SUC&377,@(SP)+               ;WAS DATA CLOCKED OUT
                                                ;SUCCESSFULLY?
                                                ;IF SO, SET UP FOR NEXT
BEQ     10$                                     ;BLOCK
```

Please add the following section to the manual:

Section 4.5.6 DU Driver Enhancements

The DU-type device driver (DUDRV) returns more information in the IO.RSN (Read Serial Number) function. The added information is used by the Bad Block Replacement Control Task (RCT). In addition to the volume serial number, the information returned includes the controller identification number, the hardware and software version numbers of the controller, the unit identifier (device identification number), and the hardware and software version numbers of the device. RCT sends this information to the Error Logger.

A new subfunction, IQ.SEL (Suppress Error Logging), allows you to prevent error messages from being sent to the Error Logger.

In addition, DUDRV does not request RCT unless the drive is set volume valid.

Please add the following sections to the manual:

Section 4.5.7 Modifications to DUDRV and PUCOM for Digital Storage Architecture Requirements

The Digital Storage Architecture (DSA) drivers, DUDRV and PUCOM, were modified as follows:

- The DSA driver DUDRV and the PUCOM partition now implement a controller specific I/O sequence number.

  The sequence number ensures that the DSA command reference number is unique, and it is useful in debugging problems related to DSA drivers.

- Because of the information RCT puts in the system error log file, DUDRV and PUCOM save information about controller identification in their internal data structures. The IO.RSN (Read Serial Number) function returns the controller identification number.

- DUDRV invokes a time delay before it requests the unit status of RA81 devices following a powerfail. Previously, RA81 devices did not always spin up because the driver requested the unit status from the controller too quickly.

  The time delay is defined in the module DSAPRE.MAC.

- DUDRV ensures that error logs are not requested from the DSA controller if error log support was not generated in the system. This behavior reduces the number of response packets that the drivers must process on systems without error log support.

- Information from internal data structures in the DSA driver PUCOM is passed to RCT because of the information logged by RCT in the system error log file. The information is returned by the IO.RSN (Read Volume Serial Number) function code. The format of the information is found in the DSAPRE.MAC module.

- The DSA driver DUDRV correctly determines, on entry into the powerfail recovery routine, if the recovery was for a controller or a unit.

- Offset P.SEQ in the UDADF.MAC module points to a controller-supplied sequence number and is used for RCT processing.

- DUDRV uses a bit definition (UU.SEL) in the UCBDF.MAC module to determine if error packets should be logged during bad block replacement by RCT. In addition, RCT uses the submodifier IQ.SEL in module QIOMAC.MAC to set and reset UU.SEL. The IQ.SEL submodifier only is supported by DUDRV.

Section 6.7 Modifications to MUDRV and PUCOM for Digital Storage Architecture Requirements

The Digital Storage Architecture (DSA) drivers, MUDRV and PUCOM, were modified as follows:

- The DSA driver MUDRV and the PUCOM partition now implement a controller specific I/O sequence number.

  The sequence number ensures that the DSA command reference number is unique, and it is useful in debugging problems related to DSA drivers.

- MUDRV ensures that error logs are not requested from the DSA controller if error log support was not generated in the system. This behavior reduces the number of response packets that the drivers must process on systems without error log support.

- The DSA driver MUDRV correctly determines on entry into the powerfail recovery routine if the recovery was for a controller or a unit.

2.12 RSX–11M–PLUS and Micro/RSX System Library Routines Reference Manual

Please make the following correction to the *RSX–11M–PLUS and Micro/RSX System Library Routines Reference Manual*:

In Table 6–1 on page 6–14, the text explaining file name string conversion using the X Directive is incorrect. Please replace that text with the following:

Directive	Form	Operation
X (file name string conversion)	%X	Convert Radix–50 filename string in ARGBLK to ASCII string in format "name.typ"; convert version number, if non-zero, to ASCII decimal string if decimal version support is selected in your system. Otherwise, the version number is converted to ASCII octal string. If the version number is zero, no version number is put into OUTBLK. Store the results in OUTBLK.
	%nX	Convert next n Radix–50 filename strings in ARGBLK to ASCII strings in format "name.typ"; convert version numbers, if non-zero, to ASCII decimal strings if decimal version support is selected in your system. Otherwise, version numbers are converted to ASCII octal strings. If a version number is zero, no version is put into OUTBLK for that filename string. Store results in OUTBLK and insert tab between strings.

Key

ARGBLK = The argument block containing the binary data to be converted, the addresses of ASCII and extended ASCII characters or the address of a double precision value.

OUTBLK = The output block in which $EDMSG is to store output.

Directive	Form	Operation
	%VX	Use the value in the next word in ARGBLK as a repeat count, convert specified number of Radix–50 filename strings to ASCII strings in format "name.typ"; convert version numbers, if non-zero, to ASCII decimal strings if the decimal version support is selected in your system. Otherwise, version numbers are converted to ASCII octal strings. If a version number is zero, no version is put into OUTBLK for that filename string. Store results in OUTBLK and insert tab between strings.

Key

ARGBLK = The argument block containing the binary data to be converted, the addresses of ASCII and extended ASCII characters or the address of a double precision value.

OUTBLK = The output block in which $EDMSG is to store output.

2.13 RSX–11M–PLUS and Micro/RSX System Management Guide

- Please add the following information to Section 1.3.3:

  The print job attributes "Print adjacent to prior job" and "Print job should be held" are included in [1,20]QMGBLD.BLD as valid options for $JATDF. This global symbol determines the selection of default qualifiers for print jobs that are spooled by the PRINT$ macro or the .PRINT subroutine in an application, or by using the /SP switch in a command such as MAC or TKB.

  Note that if QMG is rebuilt to include these options, they will apply to all such spooled print jobs. The PRINT$ macro, .PRINT subroutine, and /SP switch have no provision to modify the task-built defaults.

- In Section 5.5.16, page 5–74, the third example is incorrect. The correct example is as follows:

```
VMR>SET /SECPOL Return
SECPOL=285.:640.:44%
```

  Displays the amount of available secondary pool.

- Please add the following to Section 7.9 and replace Example 7–6 with the following example:

  The RMD C displays now include percentages on load rates for writes. Also, the format of these displays has been slightly modified to provide easier reading.

```
>rmd c

RSX-11M-PLUS V4.4  BL78      Cache Statistics (General)       17-JAN-93
12:56:54

Cache Region Name: CACHE     Region Size: 47440 (2500. disk blocks)

Device      Total  Hit Fail Load      Total  Hit Fail Load Defer     Total Cache
Name        Reads Rate Rate Rate     Writes Rate Rate Rate Rate    I/O Ops Used

DU0:       304725.  97%   0%   1%    107309.  82%   0%   0%   0%     412034. 95%
DU3:            0.   0%   0%   0%         0.   0%   0%   0%   0%          0.  0%

Total      304725.  97%   0%   1%    107309.  82%   0%   0%   0%     412034. 95%
```

- Please add the following to Section 7.10 and replace Example 7–7 with the following example:

The RMD D displays now include percentages on load rates for writes. Also, the format of these displays have been slightly modified to provide easier reading.

```
>rmd d

RSX-11M-PLUS V4.4  BL78      Cache Statistics (Detailed)      17-JAN-93
12:57:04

Device Name:  SY0:                Region Name: CACHE     Region Size: 47440
Cache Status: Active              Requests Being Cached: Dir,Ovr,Vir,Rdh

             Virtual   Readahead   Directory     Logical     Overlay       Total
Reads        122764.          0.      40793.          0.     141176.     304733.
Read Hit Rate    96%          0%         94%          0%         99%         97%
Read Load Rate    1%          0%          3%          0%          0%          1%

Read Overlap      0%          0%          2%          0%          0%          0%
Extent Too Big    0%          0%          0%          0%          0%          0%
Max Extent Size 127.        127.          3.          1.        127.

Writes        57203.                   49792.        314.                107309.
Write Hit Rate   90%                      72%          1%                    82%
Write Overlap     0%                       0%          0%                     0%

Total I/O    179967.          0.      90585.        314.     141176.     412042.

     Failure Rates (as a % of Total):       Deferred Write Rate   0%
       Primary Pool Allocation   0%         Write Load Rate       0%
       Cache Pool Allocation     0%
       Read Load                 0%
```

- Please add the following information to Section 10.2.1:

  Resource Accounting has been expanded to include a new qualifier to the START command. The qualifier is:

  ```
  /INTRUSION:yes/no
  ```

  When disabled, invalid login transaction blocks include the user's name and UIC, as found in the user account file, if the login failure results from a password discrepancy. If the user name or UIC is not found in the account file, the invalid login transaction block will simply list INVALID USER as the user name and the UIC field will be blank.

  When enabled, invalid login transaction blocks include the user's responses to the Account: and Password: prompts. The system manager can use this information to determine which patterns are being used in an intrusion attempt.

  The default is /INTRUSION:no

- In Section 13.3.4 of the manual, please add the following text:

  The Bad Block Replacement Control Task (RCT) supports the latest Mass Storage Control Protocol (MSCP) disk storage architecture specification. RCT is used with MSCP controllers, such as the UDA–50, which do not perform automatic revectoring of bad blocks. Instead, these controllers rely on the RCT task to perform revectoring for them. Revectoring is the redirection of *reference* from an unreliable block to a reliable one.

  A controller that performs its own revectoring creates a complete error log report on the I/O it handles and sends the report to the device driver. The device driver, in turn, sends a report to the Error Logger. In this way, all messages on bad blocks appear in the error log file generated when you enter an ANALYZE/ERROR_LOG command.

- In Section 13.4 of the manual, please add the following text:

When RCT performs revectoring, it creates the error log report and sends it to the Error Logger. A new error message is issued by the Task Termination Notification program (TKTN), which sends the following message to the console terminal whenever a nonrecoverable hardware error occurs:

```
*** ddnn: -- Replace command failure
```

If you receive this message, Digital recommends that you back up the media and note any errors that are reported during the backup operation. You can use the error information to determine if files are corrupted.

In addition, RCT does not produce error log packets if a device is write-protected and RCT is unable to write to the device.

- The Resource Monitoring Display (RMD) memory display supports the following two setup commands:

TOP=n	Specifies the upper limit of the memory display, where *n* is the limit value in K words.
BOTTOM=n	Specifies the lower limit of the memory display, where *n* is the limit value in K words.

  The commands allow you to examine a specific portion of system memory and will display details of system memory, including tasks and shared regions, that do not appear in a full display of system memory.

  You can enter the commands as part of the MCR command line, or you can enter the commands after you access the setup page. If you specify values for *n* that are invalid, RMD will default to a full display of system memory.

- Please add the following text to Section 15.1.4 of the manual following the DEFER_WRITES option:

  Data caching supports write loading for temporary files.

  I/O requests that can be deferred will result, if necessary, in a write load, which creates a new cache extent. Blocks in a temporary file that have the potential to be deferred will no longer have to be read before a deferred-write operation can occur. This feature will increase the number of blocks that are deferred for a task, and it will increase the performance of the deferred-write support.

- The Account File Maintenance Program (ACNT) ADD and MODIFY commands behave differently. They prompt you for confirmation when you press the Return key in response to the password field prompt or the last name (that is, Username) field prompt. Because blank values for account passwords or user names may result in security problems, Digital recommends that you do not create accounts with blank values in these fields. However, when you add or modify account fields, you may inadvertently create blank values for fields if you press the Return key in response to the field prompt instead of the Escape key which leaves a field unchanged.

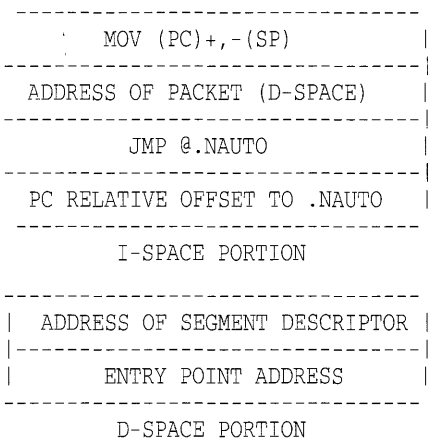
  Also, the password encryption routine has an improved handling of accounts with blank values in the user-name field or in the password field. The routine uses the user-name field and the password field to encrypt the password. Previously, the routine produced deficient results for accounts with a blank value either in the user-name field or in the password field.

  The enhancement improves security, but you are unable to log in to accounts with blank user names until you modify the user-name field, which re-encrypts the password.

2.14 RSX–11M–PLUS and Micro/RSX Task Builder Manual

Please make the following changes to the *RSX–11M–PLUS and Micro/RSX Task Builder Manual*:

- Change the fourth line of Figure 4–4 to .NAUTO as shown in the following figure:



- Change all references to FORTRAN IV–PLUS in Chapter 5 to FORTRAN–77. FORTRAN IV–PLUS is no longer supported.
- The example on page 4–16 in Section 4.5 is incorrect. Please replace it with the following correct example. (Note the corrected positioning of the period (.) in first line of the example.)

```
MOV @#N.OVPT,R0
BISB #200,N.FAST(R0)
```

- Change all references to F4PRES in Chapter 5 to F7FRES.
- Change all references to F4P in Chapter 5 to F77.
- Add the following note to Section 5.2.9.2:

Caution

This command file example will work only for FMS Version 2.0.

- Replace the TKB command sequence in Section 5.4.2 with the following:

```
TKB>VSECT,VSECT/-SP=VSECT,LB:[1,1]F77FCS/LB [Return]
TKB>/ [Return]
Enter Options:
TKB>WNDWS=1 [Return]
TKB>VSECT=MARRAY:160000:20000:200 [Return]
TKB>// [Return]
>
```

Or, if you use the LINK command, use the following command sequence:

```
$ LINK/TAS/MAP:VSECT/NOPRINT/OPT VSECT,LB:[1,1]F77FCS/LIB [Return]
Option? WNDWS=1 [Return]
Option? VSECT=MARRAY:160000:20000:200 [Return]
Option? [Return]
$
```

- Replace FOROTS.OLB with F77FCS.OLB in the second paragraph under the command sequence in Section 5.4.2.
- Replace the Note in Section 7.8.1 with the following text:

  Prior to Version 4.3, manually loaded overlaid libraries were not supported for use with I&D tasks. Instead, the overlaid library had to use autoload to load its overlays. If you tried to link an I&D task to an overlaid library that required manual loading, the following error would occur:

  ```
  TKB  -- *FATAL* -- Module name contains incompatible autoload vectors
  ```

  Support has been added for manually-loaded overlaid libraries.
- In Section 8.6.1, page 8–8, the examples are incorrect. Please replace them with the following correct examples:

  ```
  TKB>CSM/-HD/LI/PI,CMS/MA, CMS= Return
  TKB>LB:[1,1]SYSLIB/INCLUDE:CMPAL,SY:[301,55]CSM
  TKB>/ Return
  Enter Options:
  TKB>STACK=0 Return
  TKB>PAR=GEN:0:2000 Return
  TKB>CMPRT=$CMPCS Return
  TKB>BLGXCL=$SAVAL Return
  TKB>// Return
  >
  ```

  Or, you use the following LINK command sequence to build the same library:

  ```
  TKB>LINK/TAS:CSM/NOH/SHARE:LIB/CODE:PIC/MAP:CSM/SYS/SYM:CSM/OPT -
  Return
  ->LB:[1,1]SYSLIB/INCLUDE:CMPAL,SY:[301,55]CMS Return
  Option?STACK=0 Return
  Option?PAR=GEN:0:2000 Return
  Option?CMPRT=$CMPCS Return
  Option?GBLXCL=$SAVAL Return
  Option? Return
  $
  ```

  Also, add the following sentence to the end of page 8–8:

  Note that the SYSLIB module CMPAL contains both the $CMPCS and $CMPAL routines.
- In Section 11.26, page 11–36, the format of the /INCLUDE qualifier is incorrect. The correct format is shown next.

  Format

  $ LINK/TAS/MAP/SYM inputfile/INCLUDE:(MOD1,MOD2, . . . MOD8)

  Also, please remove note number 3. It is no longer applicable.

  In note number 5, the example of the /INCLUDE qualifier is incorrect. The correct example is as follows:

  ```
  $ LINK/TAS/MAP/SYM  INLIB1/INCLUDE:(MOD1,MOD2),- Return
  ->inputfile2,INLIB1/LIBRARY Return
  ```
- In Section 11.39, page 11–55, the description of the /SAVE qualifier is incorrect. All references to the file ATLINK.CMD should be changed to ATLINK.TMP.

  Also, please remove the clause "and it contains legitimate TKB command syntax" from the second paragraph.

- In Section 12.4, page 12–8, the definitions of the device-name and unit-num8 parameters are incorrect. The correct definitions are shown next.

  device-name
  Specifies a 2-character alphabetic device name followed by a 1- to 3-digit octal unit number.

  unit-num8
  Specifies decimal numbers indicating the logical unit numbers (LUNs). If your task uses more than six logical units, you should use the UNITS option to specify the number of logical units that your task will use.

  Also, the note is no longer applicable; please remove it.

- Please add the following text to Section 12.25, page 12–34 of the manual and to Section 12.29, page 12–37 of the manual:

  The RESSUP and SUPLIB options have a new parameter code /SW that allows you to write to data space in a supervisor-mode library when mapped to supervisor D-space with the MSDS$ directive. Use the /SW parameter code to specify read-write access when you build a task that links to a supervisor-mode library. You should use the parameter code /SV to specify read-only access.

  Note that, for the /SW parameter code, the supervisor-mode library must be installed with the /RON=NO switch.

  The formats of the /SW parameter code are shown next.

  Formats

  RESSUP file-specification/[-]SW[apr]

  SUPLIB file-specification/[-]SW[apr]

  In addition, the library flag word parameter, R$LFLG, has a new flag in label block 0. The definition of the flag is shown next.

Mask	Bit	Flag	Meaning
010000	12	LD$SMV	Include supervisor-mode vectors (1=NO)

- Please add the following text to Section 12.34, page 12–44 of the manual:

  For an I- and D-space task build, the task builder automatically assumes the program section specified in the VSECT option is a data program section.

- Please add the following information to Section 12.5 on page 12–10, Section 12.7 on page 12–13, and Section 12.21 on page 12–31 of the manual:

  TKB Supplementary Information
  TKB now allocates both data space APRs and instruction space APRs for libraries. At offset 404, label block 0 contains the additional field L$BAPR, which contains the data space APRs that the task or library requires. (Instruction space APR allocation information is stored in field $APRMP in psect $$TSKP. For more information on psect $$TSKP, refer to Appendix E in the *RSX–11M–PLUS and Micro/RSX Task Builder Manual*.)

When you build a library by using the TKB switch /LI, the default action is to allocate the corresponding instruction and data space APRs for the library. However, you can reserve specific data space APRs by using the new /LI subswitch /LI[:bitmask]. The appropriate bits in the bit mask should be set to specify the data space APRs that you want.

If you build a task that links to a library that was built with an earlier version of TKB, only the instruction space APRs are allocated for the library. Tasks that link to libraries built with TKB Version 4.2 have the specified library data space APRs reserved. However, you can override the library data space APR reservations by using the new bit mask option with the LIBR, RESLIB, and CLSTR options.

The formats for these options are shown next.

Formats

```
LIBR =name:accesscode[:baseAPR[:bitmask]]

RESLIB =file/accesscode[:baseAPR[:bitmask]]

CLSTR =lib1,lib2 . . . libn:accesscode[:baseAPR[:bitmask]]
```

The appropriate bits in the bit mask should be set to reserve the desired data space APRs. If you do not want to reserve data space APRs for your library, you should specify a bit mask of 000. Note that for clusters any data space APR reservation applies to all the libraries in the cluster. The bit mask for a position-independent code (PIC) library is shifted the same amount as the library. For example, if the bit mask for a 2-APR PIC library is 200 and the library is placed in APR 5 and APR 6, the bit mask is changed to 100.

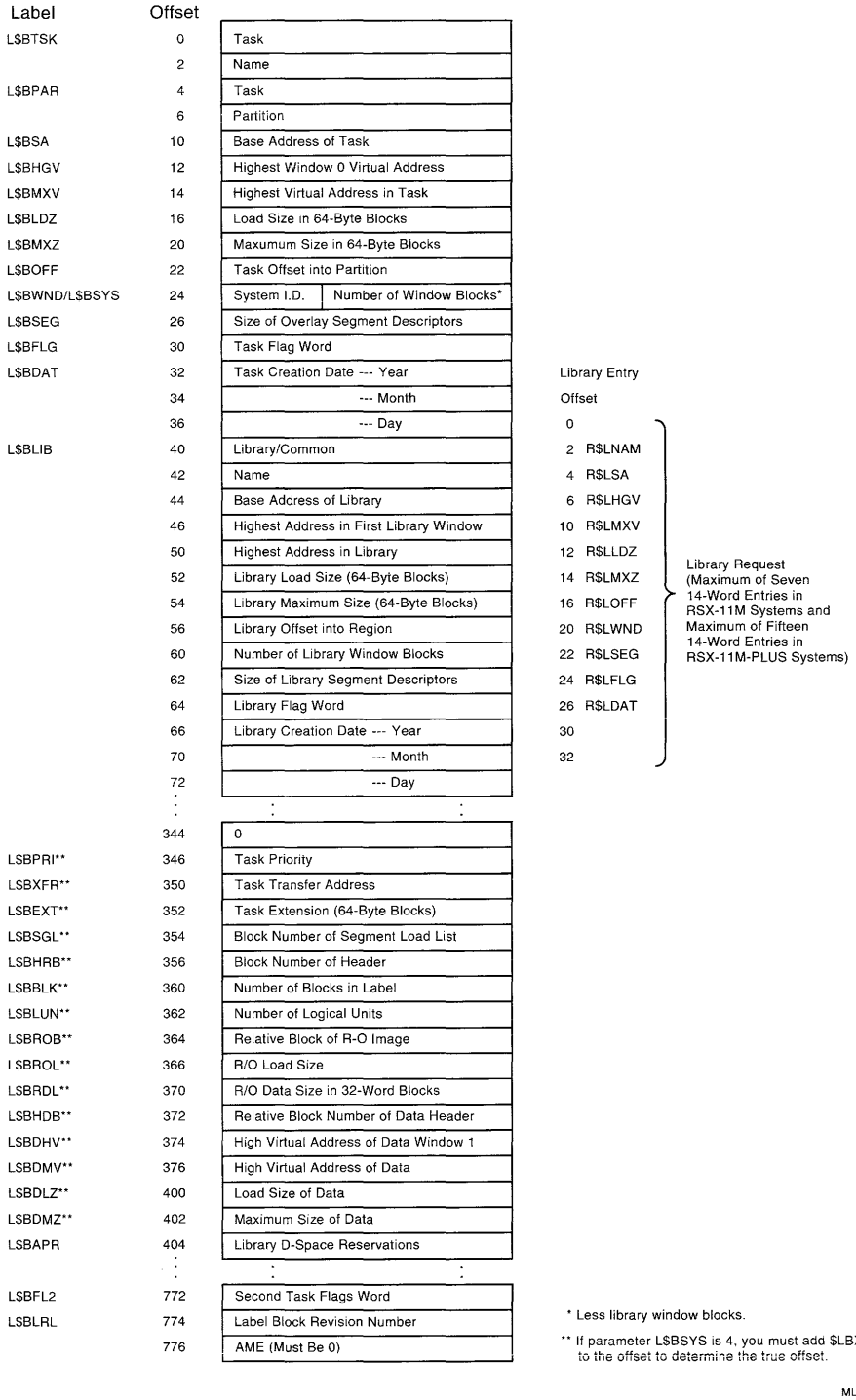
The bit mask uses the same format as the mask for the new EXTM$ directive: bit 0 represents APR0 and bit 1 represents APR1 (refer to Section 1.13.3.1). For example, if you specify 340, the APRs 7, 6, and 5 are reserved.

- In Appendix B, page B–6, the description of the Label Block Group is incorrect. Please remove the last sentence in the second paragraph, which states the following: "The LBLDF$ macro on your system will have the correct offsets."

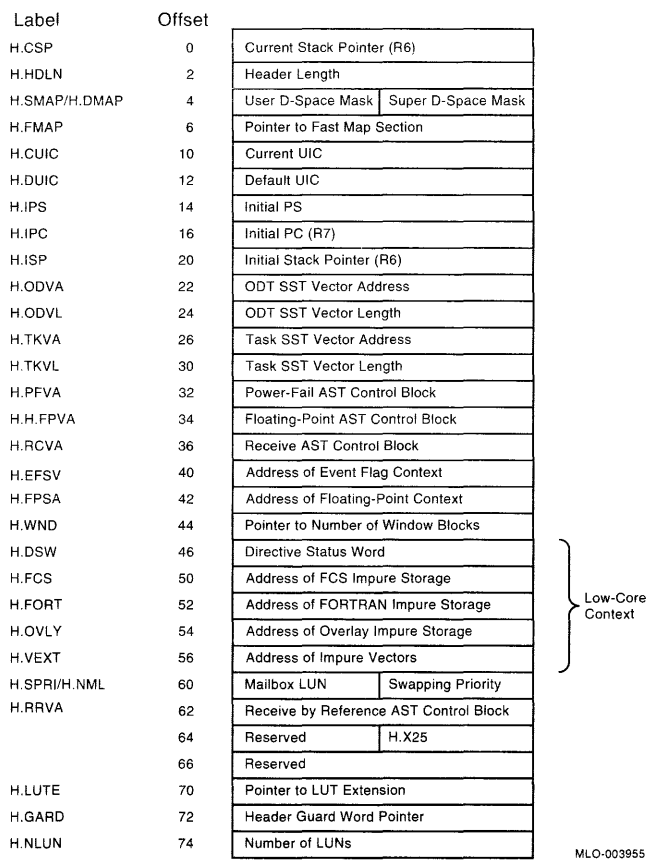
  Also, remove the third paragraph and replace it with the following paragraph:

  The LBLDF$ macro defines the label block offsets for a task that is built on an RSX–11M operating system. Tasks that are built on RSX–11M–PLUS systems have an additional eight library entries that are inserted in the label block after the entries at offset L$BLIB. Because of this, the label block offsets from L$BPRI to L$BDMZ must be adjusted by the size of the additional entries. Label block offset LBLDF$ defines the symbol $LBXL (label block extra length), which determines the size of the additional entries. Therefore, for tasks built on RSX–11M–PLUS operating systems (L$BSYS=4), you must add $LBXL to the offsets from L$BPRI to L$BDMZ in order to determine the true offset.

- In Appendix B, Figure B–5: Table Block 0–Task and Resident Library Data is incorrect. The correct figure is shown next.



- In Appendix B, Figure B–8: Task Header, Fixed Part is incorrect. The correct figure is shown next.



- In Appendix H, page H–2, the explanation for the "Illegal APR reservation" is not complete. Please make the following correction:

Illegal APR reservation

Explanation: An Active Page Register (APR) specified either with the /LI switch or in a COMMON, LIBR, RESCOM, or RESLIB option is outside the range 0 to 7.

2.15 RSX–11M–PLUS and Micro/RSX Utilities Manual

Please make the following corrections to the *RSX–11M–PLUS Utilities Manual*:

- Please add the following supplemental information to Chapter 3 in the description of the Backup and Restore Utility (BRU):

  BRU has been enhanced to improve its method of verifying or comparing data backed up from a disk to an MU-type tape device, such as the TK50. MUDRV, the driver for the MU-type devices, has also been enhanced to direct the hardware to perform a compare-host function.

  Previously, BRU used a single buffer to hold the data from the disk and another buffer to hold the data from the magnetic tape that it was comparing or verifying. This caused the tape to stop and restart frequently while data was being loaded into BRU's single buffer.

Now BRU uses both of its buffers to hold data from the disk, and a buffer created by the compare-host function in the hardware holds the data from the tape. BRU can now load data into one buffer while the other is in use, thus requiring fewer starts and stops of the tape. These changes increase the efficiency of compare and verify operations.

The new behavior is the default for MU-type devices only. You may override this default by using the SINGLEBUFFER option added to the existing BRU qualifiers /COMPARE and /VERIFY, which ensures that compare and verify operations are done as they were previously.

- Please add the following information to Table 3–3:

Table 2–6 Summary of BRU Command Qualifiers

Command Qualifiers	Options	Defaults
/COMPARE	/DOUBLEBUFFER /SINGLEBUFFER	/DOUBLEBUFFER with MU-type devices only. Otherwise, /SINGLEBUFFER is the default.
/VERIFY	/DOUBLEBUFFER /SINGLEBUFFER	/DOUBLEBUFFER with MU-type devices only. Otherwise, /SINGLEBUFFER is the default.

- Please add the following information to Section 3.3.5 on page 3–13:

The BRU qualifier /COMPARE has two options: DOUBLEBUFFER and SINGLEBUFFER. The options are used only with MU-type devices when comparing information from a disk to a tape. The input device must be a disk and the output device must be a tape or tapes.

Note

BRU does not support the /COMPARE:DOUBLEBUFFER qualifier for backup sets that contain more than one tape. Specify the /COMPARE:SINGLEBUFFER qualifier if you must compare a backup set that consists of more than one tape.

Specifying the DOUBLEBUFFER option may improve the performance of streaming tape devices such as the TK50 cartridge tape device by double buffering I/O operations. Specifying the SINGLEBUFFER option will cause BRU to single buffer I/O operations, which was the default action for versions of BRU prior to Version 4.1.

The format of the /COMPARE qualifier is shown next.

/COMPARE[:option]

Parameter

option

Specifies one of the following two options:

Option	Meaning
SINGLEBUFFER	If you use the SINGLEBUFFER option for a compare operation from a disk to a tape, BRU will print the first error found and continue comparing or verifying. It prints any other errors found. The SINGLEBUFFER option is the default for devices that are not MU-type devices.
DOUBLEBUFFER	The DOUBLEBUFFER option can be used only on MU-type devices. For most system configurations with MU-type devices, a compare operation from disk to tape will be more efficient if you use the DOUBLEBUFFER option. However, the error messages that are reported when you use the DOUBLEBUFFER option are less explicit than the messages reported when you use the SINGLEBUFFER option. As soon as an error occurs, BRU reports the actual error. If there is any subsequent data in the block, BRU cannot accurately compare the integrity of the remainder of that block. All the remaining file IDs or file names will be listed as potential errors. You must use an MU-type device drive that supports the compare host-data function. The DOUBLEBUFFER option is the default for MU-type devices.

Examples

```
BRU>/COMPARE:SINGLEBUFFER  DU0:  MU0: Return
```

Specifies a compare operation that reports a list of all actual errors found in the data block or header block.

```
BRU>/COMPARE:DOUBLEBUFFER  DU0:  MU1: Return
```

Specifies a compare operation with improved performance. As soon as an error occurs, BRU reports the actual compare error. If there is any subsequent data in the block, BRU cannot accurately compare the remainder of that block. All the remaining file IDs or file names will be listed as potential errors.

- Please add the following information to Section 3.3.33 on page 3–25:

  The BRU /VERIFY qualifier has two options: DOUBLEBUFFER and SINGLEBUFFER. The options are used only with MU-type devices when verifying information from a disk to a tape. The input device must be a disk and the output device must be a tape or tapes.

Note

During a verify operation with both MU-type and non-MU-type output devices, you must use the /VERIFY:SINGLEBUFFER qualifier.

Specifying the DOUBLEBUFFER option may improve the performance of streaming tape devices such as the TK50 cartridge tape device by double buffering I/O operations. Specifying the SINGLEBUFFER option will cause BRU to single buffer I/O operations, which was the default action for versions of BRU prior to Version 4.1.

The format of the /VERIFY qualifier is:

/VERIFY[:option]

Parameter

option

Specifies one of the following two options:

Option	Meaning
SINGLEBUFFER	If you use the SINGLEBUFFER option for a verify operation from a disk to a tape, BRU will print the first error found and continue verifying. It prints any other errors found. The SINGLEBUFFER option is the default for devices that are not MU-type devices.
DOUBLEBUFFER	The DOUBLEBUFFER option can be used only on MU-type devices. For most system configurations with MU-type devices, a verify operation from disk to tape will be more efficient if you use the DOUBLEBUFFER option. However, the error messages that are reported when you use the DOUBLEBUFFER option are less explicit than the messages reported when you use the SINGLEBUFFER option. As soon as an error occurs, BRU reports the actual error. If there is any subsequent data in the block, BRU cannot accurately verify the integrity of the remainder of that block. All the remaining file IDs or file names will be listed as potential errors. You must use an MU-type device drive that supports the compare host-data function. The DOUBLEBUFFER option is the default for MU-type devices.

Examples

```
BRU>/VERIFY:SINGLEBUFFER  DU0:  MU0: Return
```

Specifies a verify operation that reports a list of all actual errors found in the data block or header block.

```
BRU>/VERIFY:DOUBLEBUFFER  DU0:  MU1: Return
```

Specifies a verify operation with improved performance. As soon as an error occurs, BRU reports the actual verify error. If there is any subsequent data in the block, BRU cannot verify the integrity of the remainder of that block. All the remaining file IDs or file names will be listed as potential errors.

- Please add the following information to Section 3.10.3:

BRU–*WARNING*–Potential data record verify error
File ID number LBN number

Explanation: A data block on the input disk device did not match a data block on the output tape device. As soon as an error occurs, BRU cannot accurately compare or verify the integrity of the remainder of that data block. All the remaining file IDs will be listed as potential errors.

User Action: Repeat the backup operation. If the backup operation fails again, repeat the operation with a different disk or tape device.

To obtain a list of all actual errors in the data block, repeat the compare operation, specifying the /COMPARE:SINGLEBUFFER qualifier.

BRU–*WARNING*–Potential file header record verify error [directory] filename.type;version

Explanation: A header record on the input disk device did not match a header record on the output tape device. As soon as an error occurs, BRU cannot accurately compare or verify the integrity of the remainder of that header block. All the remaining file names will be listed as potential errors.

User Action: Repeat the backup operation. If the backup operation fails again, repeat the operation with a different disk or tape device.

To obtain a list of all actual errors in the header block, repeat the verify operation, specifying the /COMPARE:SINGLEBUFFER qualifier.

BRU–*WARNING*–Potential file ID area or data record verify error
File ID number LBN number

Explanation: The file ID area of a data block on the input disk device did not match the file ID area of a data block on the output tape device. As soon as an error occurs, BRU is unable to accurately compare or verify the integrity of the remainder of the data block.

User Action: Repeat the backup operation. If the backup operation fails again, repeat the operation with a different disk or tape device.

To obtain a list of all actual errors in the file ID area of a data block, repeat the verify operation, specifying the /COMPARE:SINGLEBUFFER qualifier.

BRU–*WARNING*–Double buffered compare or verify not supported

Explanation: The tape driver does not support the /COMPARE:DOUBLEBUFFER or /VERIFY:DOUBLEBUFFER qualifier.

User Action: No user action is required. BRU continues the verify or compare operation in the single buffer mode.

BRU–*FATAL*–Device conflict

Explanation: This error message can be caused by the following problems:

- You specified the /COMPARE:DOUBLEBUFFER or /VERIFY:DOUBLEBUFFER qualifier without all MU-type output devices.
- You specified both tape and disk drives as part of either the input device specification (for a restore operation) or the output device specification (for a backup operation).

User Action: If you want to mix types of output magnetic tapes (non-MU with MU), you must specify either the /COMPARE:SINGLEBUFFER qualifier or the /VERIFY:SINGLEBUFFER qualifier.

BRU–*FATAL*–Continuation volumes cannot be double buffered

Explanation: BRU does not support the /COMPARE:DOUBLEBUFFER qualifier for backup sets that contain more than one tape.

User Action: Specify the /COMPARE:SINGLEBUFFER qualifier if you must compare a backup set that consists of more than one tape.

- Please add the following information to Section 12.1.3.1 on page 12–7:

  PIP has been modified to use the embedded wildcard capability of the F11ACP, instead of processing embedded wildcards internally. This makes PIP 5 to 10 times faster when doing lookups of files with embedded wildcards.

In addition, PIP has been modified to accept embedded wildcards in a directory specification. For example:

```
>dir lb:[rms*]rms*.tsk

Directory lb:[RMS001001]
20-JAN-93 09:28

RMSRESAB.TSK;9      48.     C  30-JAN-93 13:57
RMSLBL.TSK;9        19.     C  30-JAN-93 13:57
RMSLBM.TSK;9        31.     C  30-JAN-93 13:57

Total of 98./98. blocks in 3. files


Directory lb:[RMS001054]
20-JAN-93 09:28

RMSCNV.TSK;5        176.    C  26-JAN-93 11:33
RMSIFL.TSK;5        136.    C  26-JAN-93 11:34
RMSRST.TSK;5        191.    C  26-JAN-93 11:34
RMSBCK.TSK;5        187.    C  26-JAN-93 11:34
RMSDES.TSK;5        248.    C  26-JAN-93 11:34
RMSDSP.TSK;5        183.    C  26-JAN-93 11:33
RMSDEF.TSK;5        139.    C  26-JAN-93 11:33

Total of 1260./1260. blocks in 7. files

Grand total of 1358./1358. blocks in 10. files in 2. directories
```

- Please add the following information to Section 14.5 on page 14–9:

  An abort AST has been added to VFY. If the user requests that VFY be aborted while a /UP, /RE, /DE, or /DH switch is being processed, a message warns the user that disk corruption may occur if the abort occurs at this time. If the user issues a second request to abort VFY, then VFY will be aborted.

2.16 RSX–11M/M–PLUS and Micro/RSX I/O Operations Reference Manual

Please make the following correction to the table of normal error returns and no error returns in Section 4.1:

Normal Error Return (Carry Bit and F.ERR)	No Error Return
.ASCPP	.RDFDR
.PARSE	.WDFDR
.PRSDV	.RDFUI
.PRSDV	.WDFUI
.PRSFN	.RDFFP
.ASLUN	.WDFFP
.FIND	.RFOWN
.ENTER	.WFOWN
.REMOV	.PPASC
.GTDIR	.MARK
.PGTDID	
.POINT	

Normal Error Return (Carry Bit and F.ERR)	No Error Return
.POSRC	
.POSIT	
.XQUIO	
.RENAM	
.EXTND	
.TRNCL	

The examples in Sections J.2 and J.3 are incorrect. The correct examples are as follows:

Program CRCOPA

The following sample program is titled, CRCOPA. The CRCOPA program uses a data-set descriptor instead of the default filename block used in CRCOPY.

```
        .TITLE  CRCOPA                  ; Card reader copy routine
        ;
        .MCALL  FDBDF$,FDAT$A,FDRC$A,FDOP$A,NMBLK$,FSRSZ$
        .MCALL  OPEN$R,OPEN$W,GET$,PUT$,CLOSE$,EXIT$S
        .MCALL  FINIT$
        ;
        INLUN=3
        OUTLUN=4                        ;Assign CR or file device
        FSRSZ$  2                       ;Assign to output device
FDBOUT: FDBDF$
        FDAT$A  R.VAR,FD.CR
        FDRC$A  ,RECBUF,80.
        FDOP$A  CUTLUN,OFDSPT
FDBIN:  FDBDF$
        FDRC$A  ,RECBUF,80.
        FDOP$A  INLUN,IFDSPT
RECBUF: .BLKB   80.

OFDSPT: .WORD   0,0                     ;Device descriptor
        .WORD   0,0                     ;Directory descriptor
        .WORD   ONAMSZ,ONAM             ;Filename descriptor
```

```
IFDSPT: .WORD   0,0                     ;Device descriptor
        .WORD   0,0                     ;Directory descriptor
        .WORD   INAMSZ,INAM             ;Filename descriptor
ONAM:   .ASCII  /OUTPUT.DAT/
        ONAMSZ=.-INAM
        .EVEN
INAM:   .ASCII  /INPUT.DAT/
        INAMSZ=.-INAM
        .EVEN
START:  FINIT$                          ;Init file storage region
        OPEN$R  #FDBIN                  ;Open the input file
        BCS     ERROR                   ;Branch if error
        OPEN$W  #FDBOUT                 ;Open the output file
        BCS     ERROR                   ;Branch if error
GTREC:  GET$    #FDBIN                  ;Note - URBD is all set up
        BCS     CKEOF                   ;Error should be EOF indication
        MOV     F.NRBD(R0),R1           ;R1=size of record read
        MOV     #RECBUF,R2              ;Strip trailing blanks
        ADD     R1,R2
10$:    CMPB    #40,-(R2)
        BNE     PTREC
        SOB     R1,10$
;At this point, R1 contains the stripped size of the
;record to be written.  If the card is blank,
;a zero-length record is written.

PTREC:  PUT$    #FDBOUT,,R1             ;R1 is needed to specify
        BCC     GTREC                   ;the record size

ERROR:  JMP     ERRSUB                  ;Jump to ERROR code

CKEOF:  DMPB    #IE.EFO,F.ERR(R0)       ;End of file?
        BNE     ERROR                   ;Branch if other error

        CLOSE$  #FDDIN                  ;Close the input file

        BCS     ERROR
        CLOSE$  #FDBOUT                 ;Close the output file
        BCS     ERROR
        EXIT$S                          ;Issue exit directive
        .END    START
```

Program CRCOPB

The following sample program is titled, CRCOPB. The CRCOPB program uses run-time initialization of the File Descriptor Block (FDB).

```
        .TITLE  CRCOPB                  ; Card reader copy routine
        ;
        .MCALL  FDBDF$,FDAT$A,FDRC$A,FDOP$A,NMBLK$,FSRSZ$
        .MCALL  OPEN$R,OPEN$W,GET$,PUT$,CLOSE$,EXIT$S
        .MCALL  FINIT$
        ;
        INLUN=3
        OUTLUN=4                        ;Assign CR or file device
        FSRSZ$  2                       ;Assign to output device
FDBOUT: FDBDF$
FDBIN:  FDBDF$
RECBUF: .BLKB   80.

OFDSPT: .WORD   0,0                     ;Device descriptor
        .WORD   0,0                     ;Directory descriptor
        .WORD   ONAMSZ,ONAM             ;Filename descriptor
```

```
IFDSPT: .WORD   0,0                         ;Device descriptor
        .WORD   0,0                         ;Directory descriptor
        .WORD   INAMSZ,INAM                 ;Filename descriptor
ONAM:   .ASCII  /OUTPUT.DAT/
        ONAMSZ=.-ONAM
        .EVEN
INAM:   .ASCII  /INPUT.DAT/
        INAMSZ=.-INAM
        .EVEN
START:  FINIT$                              ;Init file storage region
        OPEN$R  #FDBIN,#INLUN,#IFDSPT,,#RECBUF,#80.
                                            ;Runtime initialization
        BCS     ERROR                       ;Branch if error
        FDAT$R  #FDBOUT,#R.VAR,#FD.CR       ;Runtime initialization
        OPEN$W  R0,#OUTLUN,#OFDSPT,,#RECBUF,#80.
        BCS     ERROR                       ;Branch if error
GTREC:  GET$    #FDBIN                      ;Note - URBD is all set up
        BCS     CKEOF                       ;Error should be EOF indication
        MOV     F.NRBD(R0),R1               ;R1=size of record read
        MOV     #RECBUF,R2                  ;Strip trailing blanks
        ADD     R1,R2
10$:    CMPB    #40,-(R2)
        BNE     PTREC
        SOB     R1,10$

;At this point, R1 contains the stripped size of the
;record to be written.  If the card is blank,
;a zero-length record is written.

PTREC:  PUT$    #FDBOUT,,R1                 ;R1 is needed to specify
        BCC     GTREC                       ;the record size

ERROR:  JMP     ERRSUB                      ;Jump to ERROR code

CKEOF:  CMPB    #IE.EFO,F.ERR(R0)           ;End of file?
        BNE     ERROR                       ;Branch if other error

        CLOSE$  #FDDIN                      ;Close the input file

        BCS     ERROR
        CLOSE$  #FDBOUT                     ;Close the output file
        BCS     ERROR
        EXIT$S                              ;Issue exit directive
        .END    START
```

2.17 RSX–11M/M–PLUS RMS–11 Macro Programmer's Guide

Please make the following corrections to the *RSX–11M/M–PLUS RMS–11 Macro Programmer's Guide*:

- In Section 2.3, the argument for P$BUF is incorrectly given as "bufcount." The correct argument is "iopoolsize," as discussed in Section 2.3.4.
- In Section 5.19, the last paragraph incorrectly describes the use of the FID field in the NAM block. It should read as follows:

  "If this value is nonzero . . . "
- In Table 6–2, page 6–14, the value for the symbol XB$DAT is incorrectly given as 000003. The correct value is 000002.
- In Appendix A, page A–8, please add the following sentence:

  An attempt to insert a record that is too small to contain the whole primary key field may also cause the error ER$KEY.
- In Appendix A, Section A.1, add the following text to the description of the error ER$MRS:

Or, the sum of the fixed-length record size and the record overhead exceeds the bucket size. Or, No Span Blocks has been selected with an invalid total record size.

- In Appendix A, Section A.1, add the following text to the description of the error ER$NOD:

  Or, the remote node rejected the operation. (STV contains the Network Services Protocol (NSP) code and can be found in Appendix C of the *DECnet-RSX Programmer's Reference Manual.*)

- In Appendix A, Section A.1, add the following text to the description of the error ER$FUL:

  ER$FUL, Device or File Allocation Failure

  ```
  Octal: 176360
  Decimal: -784
  ```

  Explanation: The specified device or directory does not have enough room for file creation or extension. In the case of a contiguous request, it is also possible that there is not enough contiguous space on the device.

2.18 RSX–11M/M–PLUS RMS–11 User's Guide

Please add the following corrections to the *RSX–11M/M–PLUS RMS–11 User's Guide*:

- In Section 2.2.3.3, please add the following information to the discussion of deadlock:

  An application should use multistream rather than multichannel access to write to the same indexed file. When RMS–11 updates an RRV in a bucket that is currently locked, it must wait for that lock to be released. Control will not be returned to the program until this release occurs. Deadlock will occur when the lock is held on another channel within the same program; however, RMS–11 can update an RRV in a bucket that is locked on another stream within the same program. See your programming language documentation for details on the implementation of multistreaming.

- In Section 6.2.4, please add the following note to the discussion of writing a record:

  In the event that the record includes a partial alternate key but is not large enough to include space for the full alternate key field, RMS–11 will act as follows:

  RMS–11 will treat the alternate key as if it were not present in the record, making no entry in the alternate key index structure.

- According to the *RSX–11M/M–PLUS Macro Programmer's Guide*, RMS–11 cannot perform an UPDATE operation on an alternate key with the key characteristics CHANGES and NODUPLICATES. This description is misleading. To clarify the description, please add the following information to Chapter 6, Section 6.2.5.2:

  Although RMS–11 does not support the CHANGES and NODUPLICATES combination, it does not prevent you from performing an UPDATE operation on an alternate key with these characteristics. When an update causes a duplicate of an alternate key, RMS–11 returns the completion code ER$DUP. However, it does not terminate the UPDATE operation. Instead, RMS–11

updates the primary data level for the record without updating the alternate index. As a result, the file contains duplicates of the alternate key.

To prevent RMS–11 from creating duplicates when you make changes on alternate keys, modify your application as follows:

1. Create the file with the key characteristics DUPLICATES and CHANGES.
2. To disallow duplicates, perform a FIND operation on each alternate key. Then, perform an UPDATE operation on the modified record.

- In Section 6.3, please add the following information to the discussion of contiguous areas:

  You will gain a small benefit by setting areas to contiguous on a noncontiguous multiarea file; however, RMS–11 cannot determine if those areas remain contiguous. Consequently, RMSDSP and RMSDES will display them as noncontiguous. As long as the areas are preallocated, they will behave like contiguous areas; as soon as they need to be extended, they will behave like noncontiguous areas.

- In Section 8.1.2.1, in the discussion of task building against the RMS–11 resident library, incorrect syntax is documented for the cluster option in the Task Builder command file. The correct syntax is as follows:

  CLSTR = RMSRES,DAPRES:RO

- In Section 8.1.2.1, change paragraph 2, sentence 3 to read as follows:

  To add remote access (DAP) support to a task that is built against the RMSRES in supervisor mode, you must include the modules:

  ```
  LB:[1,1]RMSLIB/LB:R0EXSY:R0IMPA-LB:[1,1]RMSDAP/LB:R0AULS
  ```

  and include DAPRES as a LIBR or CLSTR option in the task builder command sequence.

  Note

  Unlike RMSRES, the DAPRES cannot be used in supervisor mode.

- In Appendix B, in the discussion of remote file and record access using the DECnet package, the documentation states that the RSTS/E file access listener (FAL) does not support remote record access to indexed files. This is no longer true.

2.19 RSX–11M/M–PLUS RMS–11 Utilities

Please make the following corrections to the *RSX–11M/M–PLUS RMS–11 Utilities* manual:

- In Chapter 2, Table 2–1, add the following RMSDES commands:

Command	Format and Function
EXIT_SUPERSEDE	EXIT_S[UPERSEDE] filename[.typ] Names the description file in which the file design is stored.

Command	Format and Function
SAVE_SUPERSEDE	SAVE_S[UPERSEDE] filename[.typ] Names the description file in which the file design is saved.

- In Chapter 2, Section 2.2.5, add the new command EXIT_SUPERSEDE, as follows:

  The EXIT_SUPERSEDE command stores the file design in the description file specified in the command string, superseding any existing file by the same name. EXIT_SUPERSEDE then terminates RMSDES and returns the system prompt.

  The format for the EXIT_SUPERSEDE command is as follows:

  ```
  EXIT_S[UPERSEDE] filename[.typ]
  ```

  EXIT_SUPERSEDE names the description file in which the file design is stored. The default file type is DES. If you do not want to supersede an existing description file, use the EXIT command.

- In Chapter 2, Section 2.2.5, add the following statement to the description of the EXIT command:

  To supersede an existing description file, use the EXIT_SUPERSEDE command.

- In Chapter 2, Section 2.2.9, add the new command SAVE_SUPERSEDE, as follows:

  The SAVE_SUPERSEDE command stores the file design in the description file specified in the command string, superseding any existing file by the same name. If you do not define areas when you issue the SAVE command, RMSDES prompts you for the areas.

  The format for the SAVE_SUPERSEDE command is as follows:

  ```
  SAVE_S[UPERSEDE] filename[.typ]
  ```

  SAVE_SUPERSEDE names the description file in which the file design is saved. The default file type is DES. If you do not want to supersede an existing description file, use the SAVE command.

  If you want to design another file, issue a CLEAR ALL command to restore the attribute values in the design buffer to their defaults.

- In Chapter 2, Section 2.2.9, add the following statement to the discussion of the SAVE command:

  To supersede an already existing description file, use the SAVE_SUPERSEDE command.

- In Chapter 2, Section 2.6.2, add the following corrections:

  - Modify paragraph 3 of the ALLOCATION field discussion to read as follows:

    If you intend to create a single-area indexed file and do not require RSX positioning, RMS–11 uses the allocation from the file section if no area section exists in your design buffer. If you are allowing RMSDES to define areas for an indexed file by default, RMSDES will automatically calculate an allocation value for each area it defines.

- Modify paragraph 4 of the EXTENSION field discussion to read as follows:

  If you intend to create a single-area indexed file and do not require RSX positioning, RMS–11 uses the extension from the file section if no area section exists in your design buffer. If you are allowing RMSDES to define areas for an indexed file by default, RMSDES will automatically calculate an extension value for each area it defines.

- Modify paragraph 5 of the BUCKETSIZE field discussion to read as follows:

  If you intend to create a single-area indexed file and do not require RSX positioning, RMS–11 will use the bucket size from the file section if no area section exists in your design buffer. If you are allowing RMSDES to define areas for an indexed file by default, RMSDES will assign a bucket size value for each area it defines. However, if you choose to define areas explicitly and specify a bucket size value for each area, you should accept the default for the file section and set the bucket size value in each area section.

- In Chapter 2, Section 2.8, add the following correction to the explanation of the error message "?DES-F-VOR":

  You entered a value in response to an attribute prompt that was not in the legal range of values for that attribute, or the values you entered resulted in a calculation that caused an overflow for RMSDES. If the value was not within the legal range, the error message is followed by a display of the incorrect value.

- In Chapter 4, Table 4–1, add the following switch and description to the table of RMSCNV switches:

/ER[:filespec]	Continue processing after encountering an exception record. If a file specification is provided, then write the primary keys of exception records into the specified file. If no file specification is provided, then output the exception records to the terminal. Default: Stop processing and report RMS error code.

- In Section 4.3, add the following information to the description of RMSCNV switches:

  /ER[:filespec]
  Directs RMSCNV to continue processing when it encounters an exception record in the input file that cannot be written to the output file (see Section 3.4). If you specify a file specification, the exception records will be written to that file. If you do not specify a file specification, the primary key of each exception record will be issued to the terminal. RMSCNV also issues exception record codes (see RMSIFL exception codes, Section 3.3.2).

  If you specify an exception file specification, RMSCNV will create the file as an RMS–11 Variable Fixed Control (VFC) sequential file upon encountering the first exception record. RMSCNV will then write the exception record with a 4-byte exception code to the fixed-control area of the record.

  By default, if you do not specify the /ER switch, RMSCNV will stop processing upon encountering the first exception record and will issue an error message indicating the type of exception record.

- In Table 5–1, add the following information to the description of RMSDSP switches:

/BR	Briefly displays attributes.
/SU	Supersedes existing output file.

- In Section 5.2, include the asterisk (*) and percent sign (%) in the description of wildcard characters permitted in the input file specification.
- In Section 5.3, add the following information to the discussion of RMSDSP commands:

/BR
Directs RMSDSP to issue basic displays for indexed files (see Section 5.4, Example 5–3) and container files (see Section 5.4, Example 5–6).

/SU
Directs RMSDSP to supersede any existing output file with the same name and version number as the output file specification. If this switch is not supplied and the version numbers are the same, RMSDSP will issue the following error message:

```
?DSP-F_OPNINP, Error opening DDnn:file.dat as output
-RMS-E-ER$FEX, File already exists
```

- In Table 6–1, add the following information to the table of RMSBCK switches:

/NV	Creates a new version of the output file.

- In Section 6.2, in the discussion of RMSBCK command line format, include the asterisk (*) and percent sign (%) in the description of wildcard characters permitted in the input file specification.
- In Section 6.3.2, add the following information to the description of RMSBCK output switches:

/NV
Directs RMSBCK to create a new version of the disk output file if a file currently exists with the same version number as the input file. The current file is not deleted. If you do not specify this switch and a file currently exists with the same file name and version number as the input file specification, RMSBCK will issue the following fatal error message:

```
?BCK-F-CREOUT, Error opening ddnn:file.dat;n as output
-RMS-E-ER$FEX, File already exists
```

- In Table 7–1, add the following information to the table of RMSRST switches:

/NV	Creates a new version of output file.

- In Section 7.2, in the discussion of the RMSRST command line format, include the asterisk (*) and percent sign (%) in the description of wildcard characters permitted in the input file specification.
- In Section 7.3.2, add the following information to the discussion of RMSRST commands:

/NV

Directs RMSRST to create the next higher version number if the expanded input file has the same version number as an existing output file. If this switch is not used and the file name and version number are the same, RMSRST will display the following error message:

```
?RST-F-CREOUT, Error opening ddnn:file.dat;n as output
-RMS-E-ER$FEX, File already exists
```

3

RMS–11 Version 2.0 Software Modifications, Restrictions, and Supplementary Information

This chapter describes RMS–11 software modifications, lists restrictions, and supplementary information for RSX–11M–PLUS.

RMS–11 Version 2.0 has not changed version numbers for this release of RSX–11M–PLUS. Information in this chapter has not been incorporated into the RMS–11 manuals.

3.1 RMS–11 Corrections

The following sections describe corrections for RMS–11. These problems have all been corrected.

3.1.1 RMS–11 Local Access

RMS–11 applications, that performed multistreaming asynchronous record operations resulting in a high rate of competition for the same buckets, encountered the following set of problems:

- Execution of breakpoint trap in RMS modules R0RLSB and R0RSET.
- Looping indefinitely in RMS modules R0RSET and R0RLCH.
- Stalled indefinitely while waiting on the RMS event flag.
- Receiving "Dynamic Memory Exhausted" error (ER$DME) on a $FIND, $GET, $PUT, $UPDATE or $DELETE operation (operations which do not require new use of dynamic memory).
- Receiving "Bucket Header Checkbyte" error (ER$CHK) when in fact the file does not have this problem.

3.1.2 RMS–11 Ease of Use

Two new ODL files have been added to simplify the building of RMS programs that utilize RMS in supervisor mode.

- LB:[1,1]RMSSLX.ODL – for programs performing local access only.
- LB:[1,1]DAPSLX.ODL – for programs performing local and remote access.

Note

The remote access library (DAPRES) cannot be used in supervisor mode.

3.1.3 RMS–11 Utilities Enhancements

A correction was made in RMSIFL to preserve worst case error on input file read errors. While the error was reported, if RMSIFL was able to continue, the exit status was incorrectly reported as success.

RMSIFL incorrectly reported exception records on packed decimal type records containing negative key values. This problem has been corrected.

Backup sets created on RSX–11M, or octal-based RSX–11M–PLUS systems, were restored with incorrect version numbers. RMSBCK incorrectly assumed that the source file came from a decimal-based system. For example: file FOO.DAT;10 was restored as FOO.DAT;12 even though both the input and output systems used octal version numbers. When this same file was restored to an RSX–11M–PLUS system that supported decimal version numbers, the version number remained ;10 rather than being converted to ;8. This problem did not exist if the input system supports decimal version numbers. This problem has been corrected.

3.2 Software Restrictions

This section describes software restrictions that apply to RMS–11 Version 2.0.

3.2.1 RMS–11 Access Methods

RMS–11 tasks built prior to Version 3.0 of RSX–11M–PLUS will return the error message "Directory not found" (ER$DNF) on certain file operations that are executed from an account set to nonamed directory mode. This will occur if one or both of the following conditions are true:

- The tasks were not built against the RMS–11 resident library.
- A directory is not provided in either the file specification or the default file specification.

You can resolve this problem by first providing a directory in the file specification or by setting your terminal to named directory mode. If that is not possible, you should rebuild those tasks by using the new version of RMS–11.

3.2.2 Restrictions to RMS–11 Remote Access Methods (RMSDAP)

The following sections describe restrictions to RMS–11 Remote Access Methods.

3.2.2.1 Incorrect Interpretation of Keysize Field Values

RMSDAP does not interpret a keysize field value of zero correctly. This causes keyed access to a remote indexed file to fail. The application passes the value zero for access to a nonstring key and issues the following error message:

```
ER$RNF (Record not found)
```

If your program is written in F77 or other high-level languages, you may have to use a USEROPEN routine in order to be able to set this value.

3.2.2.2 Using RMSDAP on Your System

The RMSDAP provided on your system uses a 576 byte buffer for transfers between your program's record/user buffer and your DECnet-RSX system network buffer. This buffer must be large enough to hold the record and the DAP message overhead (approximately 36 bytes). In certain rare cases, this buffer length may be insufficient.

In particular, if you access a sequential variable file with a Maximum Record Size (MRS) of zero (used when no maximum has been set), and the sum of the largest record length (LRL) and the DAP message overhead exceeds the 576 byte length, your record access will fail with the RMS error ER$MRS. This occurs because RMSDAP functions at a DAP protocol level that does not allow reconfiguration of buffersize without actually closing and reopening the file (and that is an unacceptable alternative for RMSDAP). If it is within your control to create the remote file, then an explicit MRS should be used. If not, the local RMSDAP buffer can be lengthened by the system manager (prior to run time) depending on the following considerations:

- If your program was built with in-task RMSDAP, consult your map to determine the location of the variable $BUF.M (current contents 1100 octal) in the module R0NFRT in your application. After saving a copy of your application, ZAP the desired buffer length (including DAP overhead) into your application. This change will only affect your application.
- If your program was built against the DAPRES library, consult the map (LB:[1,34]DAPRES.MAP) and determine the location of $BUF.M in module R0NFRT. Remove the DAPRES from memory. After saving a copy of LB:[1,1]DAPRES.TSK, ZAP in the desired buffersize. Reinstall the DAPRES library. This change will affect all programs built and run against the DAPRES library.

In both cases the increased size of the buffer will be reflected in the size of your RMSDAP applications.

Note

Network transfers rely on configuration of local RMSDAP and DECnet–RSX network buffers as well as remote FAL and DECnet–RSX network buffers. In all negotiations, the smaller buffer length is used. If your local RMSDAP buffer size exceeds the local DECnet–RSX buffer, the remote FAL, or the remote network buffer size the excess will be unusable, thus effectively eliminating the workaround.

3.2.3 RMS–11 Utilities

This section describes restrictions to the RMS–11 utilities for Version 2.0.

RMSCNV Restriction

RMSCNV ignores user-provided area extension quantities when loading a file. The values that it uses are large enough to reduce the number of file extensions in most cases.

RMSDEF Restriction

The DCL command DEFINE defines logical names; therefore, when you invoke RMSDEF, you need to either invoke MCR before typing DEF (to use the MCR mode of DEF instead of the default DCL mode) or insert the following in the LB:[1,2]STARTUP.CMD file:

```
INSTALL LB:[3,54]RMSDEF/TASK=...DFN (or a task name of your choice)
```

3.3 RMS–11 Supplementary Information

The following sections describe information that has been documented in the previous release notes, but is still applicable.

3.3.1 Enhancements to RMSDES Utility

The following enhancements have been made to the RMSDES utility:

- There are two new RMSDES commands: SAVE_S[UPERSEDE] and EXIT_S[UPERSEDE].
- RMSDES issues the warning message "%DES-W-CBK, Continuation buckets will be allocated for this key" when continuation buckets are required. (Continuation buckets may result in decreased performance; see the *RSX–11M/M–PLUS RMS–11 User's Guide.*)
- The informational message "?DES-F-VOR, Value out of legal range" is issued if overflow occurs during calculations of indexed file area allocations. Also, the "number of duplicates:" prompt has been expanded to "number of duplicates per record on this key:" for clarity.
- The error message "%DES-F-NHF, Help file is not available. Check release notes for the location of RMSDES.IDX on your installation media" is issued if the help file LB:[1,2]RMSDES.IDX is not found.

  This is a feature for small systems that have limited disk space.

3.3.2 RMS–11 Installation

The following sections describe information for installing RMS–11 on an RSX–11M–PLUS Version 4.2 or higher operating system.

On RSX–11M–PLUS operating systems, all RMS–11 files are automatically on your system.

The paragraph in Section 5.4.7 on page 5–43 in the *RSX–11M–PLUS System Generation Guide and Installation Guide* is incorrect. Please replace it with the following paragraph:

RMS–11 Version 2.0 is included on the pregenerated kit disk. The RMS–11 segmented library (RMSRES, and RMSLBL through RMSLBM) and all the RMS–11 utilities are already installed in the system image. No further installation is needed, unless you install the DECnet package on your system and you want to use the RMS–11 remote access facilities. See Section 5.4.1 in the *RSX–11M–PLUS System Generation Guide and Installation Guide* for information on installing the RMS–11 remote access package (DAPRES).

3.3.2.1 Location of RMS–11 Files on the Distribution Kit

Table 3–1 describes the location of RMS–11 Version 2.0 files on the RSX–11M–PLUS Version 4.4 distribution kit.

Note

Some file names are marked with an asterisk (*). This indicates that the files are not included on RL02 distribution kits. They are excluded because of space limitations on the kit, and because they are used only to rebuild components of RMS–11 Version 2.0.

Table 3–1 RMS–11 Files on RSX–11M–PLUS Distribution Kit

File Name	Destination	Comments
RMSMAC.MLB	LB:[1,1]	Can be deleted if you are not using MACRO–11 RMS–11 programs
RMSLIB.OLB	LB:[1,1]	Object library for RMS–11 local access
RMSDAP.OLB	LB:[1,1]	Object library for RMS–11 remote access (can be deleted if you do not need remote access)
RMSFUN.OBJ	LB:[1,1]	RMS function routines for the system library
RMSRESAB.MAP	LB:[1,34]	Map for library root
RMSRESAB.TSK	LB:[1,1]	Installed as the root of the resident library
RMSFAKAB.CMD RMSROTAB.STB	LB:[1,24]	For rebuilding RMSLBL and RMSLBM
RMSRESAB.CMD RMSRESAB.ODL	LB:[1,24]	For rebuilding RMSRESAB
RMSBCK.TSK	LB:[3,54]	RMSBCK utility; uses RMSRES
RMSRST.TSK	LB:[3,54]	RMSRST utility; uses RMSRES
RMSDEF.TSK	LB:[3,54]	RMSDEF utility; uses RMSRES
RMSDSP.TSK	LB:[3,54]	RMSDSP utility; uses RMSRES
RMSCNV.TSK	LB:[3,54]	RMSCNV utility; uses RMSRES
RMSDES.TSK	LB:[3,54]	RMSDES utility; uses RMSRES
RMSIFL.TSK	LB:[3,54]	RMSIFL utility; uses RMSRES
RMSDES.IDX	LB:[1,2]	Indexed help file used by RMSDES
RMS11.ODL	LB:[1,1]	Prototype ODL file
R0RMS1.MAC	LB:[1,1]	For use with the prototype ODL
RMS11S.ODL	LB:[1,1]	ODL file for sequential access
RMS12S.ODL	LB:[1,1]	ODL file for sequential access
RMS11X.ODL	LB:[1,1]	Standard indexed file ODL
RMS12X.ODL	LB:[1,1]	Indexed file ODL
RMSRLX.ODL	LB:[1,1]	ODL for use with RMSRES
RMSSLX.ODL	LB:[1,1]	ODL for use with RMSRES in supervisor mode
DAP11X.ODL	LB:[1,1]	ODL for use with overlaid RMSDAP
DAPRLX.ODL	LB:[1,1]	ODL for use with DAPRES
DAPSLX.ODL	LB:[1,1]	ODL for use with RMSRES in supervisor mode and DAPRES (not in supervisor mode)
RMSRES.TSK	LB:[1,1]	Task image for linking in nonsupervisor mode; *not* to be installed as the root of the library
RMSLBL.TSK	LB:[1,1]	Task image for RMSRES segment
RMSLBM.TSK	LB:[1,1]	Task image for RMSRES segment
RMSRES.STB	LB:[1,1]	STB file for RMSRES
DAPRES.TSK	LB:[1,1]	Task image for RMSDAP resident library
DAPRES.STB	LB:[1,1]	STB file for DAPRES

(continued on next page)

Table 3–1 (Cont.) RMS–11 Files on RSX–11M–PLUS Distribution Kit

File Name	Destination	Comments
RMSLBL.MAP* RMSLBM.MAP* DAPRES.MAP*	LB:[1,34]	Maps for segmented library
RMSRES.TSK	LB:[3,54]	Task image for linking in supervisor mode; *not* to be installed as the root of the library
RMSRES.STB	LB:[3,54]	STB file for linking supervisor-mode tasks
RMSLBL.CMD* RMSLBL.ODL*	LB:[1,24]	For rebuilding RMSLBL
RMSLBM.CMD* RMSLBM.ODL*	LB:[1,24]	For rebuilding RMSLBM
DAPRES.CMD* DAPRES.ODL*	LB:[1,24]	For rebuilding DAPRES
BCKNON.CMD* BCKNON.ODL*	LB:[1,24]	For rebuilding the overlaid version of RMSBCK
BCKNRN.CMD* BCKNRN.ODL*	LB:[1,24]	For rebuilding the resident-library version of RMSBCK
BCKNSN.CMD* BCKNSN.ODL*	LB:[1,24]	For rebuilding the supervisor-mode version of RMSBCK
CNVNON.CMD* CNVNON.ODL*	LB:[1,24]	For rebuilding the overlaid version of RMSCNV
CNVNRN.CMD* CNVNRN.ODL*	LB:[1,24]	For rebuilding the resident-library version of RMSCNV
CNVNOO.CMD* CNVNOO.ODL*	LB:[1,24]	For rebuilding the overlaid version of RMSCNV with RMSDAP
CNVNRR.CMD* CNVNRR.ODL*	LB:[1,24]	For rebuilding the resident-library version of RMSCNV with RMSDAP
CNVNSN.CMD* CNVNSN.ODL*	LB:[1,24]	For rebuilding the supervisor-mode version of RMSCNV
DEFNON.CMD* DEFNON.ODL*	LB:[1,24]	For rebuilding the overlaid version of RMSDEF
DEFNRN.CMD* DEFNRN.ODL*	LB:[1,24]	For rebuilding the resident-library version of RMSDEF
DEFNSN.CMD* DEFNSN.ODL*	LB:[1,24]	For rebuilding the supervisor-mode version of RMSDEF
DESNON.CMD* DESNON.ODL*	LB:[1,24]	For rebuilding the overlaid version of RMSDES
DESNRN.CMD* DESNRN.ODL*	LB:[1,24]	For rebuilding the resident-library version of RMSDES
DESNSN.CMD* DESNSN.ODL*	LB:[1,24]	For rebuilding the supervisor-mode version of RMSDES
DSPNON.CMD* DSPNON.ODL*	LB:[1,24]	For rebuilding the overlaid version of RMSDSP
DSPNON.CMD* DSPNON.ODL*	LB:[1,24]	For rebuilding the resident-library version of RMSDSP

(continued on next page)

Table 3–1 (Cont.) RMS–11 Files on RSX–11M–PLUS Distribution Kit

File Name	Destination	Comments
DSPNSN.CMD* DSPNSN.ODL*	LB:[1,24]	For rebuilding the supervisor-mode version of RMSDSP
IFLNON.CMD* IFLNON.ODL*	LB:[1,24]	For rebuilding the overlaid version of RMSIFL
IFLNRN.CMD* IFLNRN.ODL*	LB:[1,24]	For rebuilding the resident-library version of RMSIFL
IFLNSN.CMD* IFLNSN.ODL*	LB:[1,24]	For rebuilding the supervisor-mode version of RMSIFL
RSTNON.CMD* RSTNON.ODL*	LB:[1,24]	For rebuilding the overlaid version of RMSRST
RSTNRN.CMD* RSTNRN.ODL*	LB:[1,24]	For rebuilding the resident-library version of RMSRST
RSTNSN.CMD* RSTNSN.ODL*	LB:[1,24]	For rebuilding the supervisor-mode version of RMSRST
RMSUTL.OLB* RMSODL.ODL*	LB:[1,24]	For rebuilding the utilities
GSA.MAC	LB:[USER]	Demonstration program included as an illustration of how to extend an RMS–11 task in the event of pool exhaustion
PARSE.MAC SEARCH.MAC RENAME.MAC ERASE.MAC PARSE.TSK SEARCH.TSK RENAME.TSK ERASE.TSK	LB:[USER]	Demonstration programs for the directory and wildcarding facilities

Note

All RMSDAP files can be deleted if you are not using RMS–11 to access files on remote nodes.

3.3.2.2 Startup Command Procedures

To install RMS–11, install the resident libraries and RMS–11 utilities at system startup. To aid you in the installation process, the file LB:[1,2]STARTUP.CMD contains commands and sample comments that can be edited to become system startup commands. Please note the following items:

- On RL02 pregenerated systems, the RMS–11 segmented library (RMSRES, RMSLBL, RMSLBM) and all the RMS–11 utilities are already installed in the system image. For these kits, the only installation needed is the optional installation of the DAPRES resident library if the system has DECnet support and if RMS–11 remote access facilities are used.
- On systems other than RL02, the startup file contains commands that install the segmented resident library in the system image. Note that these are commands, not comments; if you do not want to install the library, you

should edit the file to change these commands to comments. Please note the following items:

- All resident libraries should be installed using the option /RON=YES.
- The root of the library, RMSRESAB, must be the task image contained in LB:[1,1]]. The [1,1]RMSRES.TSK should never be installed in the system; it is only used when tasks are linked to use the non-supervisor-mode version of RMSRES.
- Failure to install the root segment RMSRES will cause the error "INS – Common block not Loaded RMSRES" when a referencing task or utility is invoked. Failure to install any of the remaining resident library segments will not give an error at invocation. However, it may cause the error codes ER$LIB or ER$ENV, or a BPT trap (with R0 containing ER$LIB), when the missing segment is needed by RMS–11.
- There are several circumstances involving "inconsistent or incomplete resident libraries" that can cause a BPT trap to be generated with R0 containing the error code ER$LIB. This can occur if not all the segments of the library are installed or if the version numbers of one or more segments do not match the root segment, the RMSDAP code, or the task itself. In particular, this can happen to the RMS–11 utilities if they are built to use the segmented resident library and the segments are installed incorrectly.
- The resident libraries have been built with the PAR=parname option in the TKB command file, where parname is the name of the resident library. This feature was included for compatibility with RSX–11M systems. You should install the resident libraries in the GEN partition. Do not generate individual partitions for the libraries on the RSX–11M–PLUS operating system; doing so would negate much of the benefit of having a "demand-paged" segmented library.

- If you are using RMSRES in supervisor mode, you do not need to install a different task. You simply need to link your task.
- If you want to use the RMS–11 utilities or the RMSDAP resident library (RMSDAP), you must edit the startup command file, which contains template commands (comments) that you can edit to install these facilities.

3.3.2.3 Utility Configurations

The RMS–11 utilities that are provided on the distribution kit are built to use the segmented resident library. Consequently, RMSRES, RMSLBL, and RMSLBM should be installed before you use any of the RMS–11 utilities.

3.3.2.4 Utility Command and ODL Files

For each utility, a command file and one or more ODL files are provided to build the utility using disk-overlaid RMS–11. The names of the files are as follows:

Utility	Files
RMSBCK	BCKNON.CMD, BCKNON.ODL, and RMSODL.ODL
RMSRST	RSTNON.CMD, RSTNON.ODL, and RMSODL.ODL
RMSCNV	CNVNON.CMD and CNVNON.ODL
RMSDSP	DSPNON.CMD, DSPNON.ODL, and RMSODL.ODL

Utility	Files
RMSDES	DESNON.CMD, DESNON.ODL, and RMSDES.ODL
RMSDEF	DEFNON.CMD, DEFNON.ODL, and RMSODL.ODL
RMSIFL	IFLNON.CMD and IFLNON.ODL

In addition, a command file and ODL file are provided to build each utility with the resident library RMSRES. The names of the files are as follows:

Utility	Files
RMSBCK	BCKNRN.CMD and BCKNRN.ODL
RMSRST	RSTNRN.CMD and RSTNRN.ODL
RMSCNV	CNVNRN.CMD and CNVNRN.ODL
RMSDSP	DSPNON.CMD and DSPNON.ODL
RMSDES	DESNRN.CMD and DESNRN.ODL
RMSDEF	DEFNRN.CMD and DEFNRN.ODL
RMSIFL	IFLNRN.CMD and IFLNRN.ODL

Finally, a command file and ODL file are provided to build each utility using the supervisor-mode library RMSRES. The names of the files are as follows:

Utility	Files
RMSBCK	BCKNSN.CMD and BCKNSN.ODL
RMSRST	RSTNSN.CMD and RSTNSN.ODL
RMSCNV	CNVNSN.CMD and CNVNSN.ODL
RMSDSP	DSPNSN.CMD and DSPNSN.ODL
RMSDES	DESNSN.CMD and DESNSN.ODL
RMSDEF	DEFNSN.CMD and DEFNSN.ODL
RMSIFL	IFLNSN.CMD and IFLNSN.ODL

For RMSCNV, two additional configurations are available if you want to use RMSCNV to access files on remote nodes. To build RMSCNV using the clustered RMS–11 and RMSDAP resident libraries, use the files CNVNRR.CMD and CNVNRR.ODL. To build RMSCNV using disk-overlaid RMS–11 and RMSDAP, use the files CNVNOO.CMD and CNVNOO.ODL.

Note

These command files and ODL files are not supplied with the RL02 kits.

3.3.2.5 Rebuilding the Utilities

To rebuild the utilities, perform the following steps:

1. Log in to a privileged account.
2. Set your default account to directory [1,24] on the system disk.
3. Use the Task Builder (TKB) to build the utility or utilities.

The utilities and the corresponding map files will be built in the current account. You may then want to put the utility in the system account ([3,54]).

Each command file for a particular utility creates a utility of the correct name. For example, CNVNRN.CMD and CNVNON.CMD both produce task images called RMSCNV.TSK. One is built to use the resident library and one is built with disk-overlaid RMS–11.

4

Layered Product Support

4.1 Correction Files

Table 4–1 lists the layered products and correction files that have been updated for the RSX–11M–PLUS Version 4.4 operating system. The application of the updated correction files to each layered product will result in improved software performance. Table 4–1 also includes the corresponding directories and documentation files for the layered products. The documentation files describe the procedure for applying corrections to the specific layered product.

Table 4–1 Layered Product Correction File References

Layered Product	Directories Used	Backup Set	Documentation File Name
PDP–11 BASIC–PLUS–2 Version 2.7	[211,200]	BP2	[211,200]BP2.DOC
DATATRIEVE–11/RSX Version 3.3	[222,200]	DTR	[222,200]DTRREL.DOC
DCX KMV11-A Development Tools Version 1.03	[224,200]	DCX	[224,200]DCX.DOC
DECnet–11M–PLUS Version 4.4	[230,200] [225,20] [230,20]	DECNET	[230,200]DECNET.DOC
KMV11-A HDLC Framing Software Version 1.02	[253,200]	KMV11	[253,200]KMV11.DOC
KMV11 X25 Link Level Version 1.03	[322,200]	KXI	[322,200]KXI.DOC
PDP–11 FORTRAN–77 Version 5.4	[246,200]	F77	[246,200]F77.DOC

4.2 Installing Layered Products Restriction

Because of support for named directories, the MCR command SET /NONAMED or the DCL command SET DEFAULT/NONAMED should be used before installing layered products.

4.3 Invalid BASIC–PLUS–2 Installation Warning

BASIC–PLUS–2 Version 2.6 checks for the presence of the slow Task Builder (STK) even though the installation does not need that utility. You may ignore any warnings displayed when you install BASIC–PLUS–2 Version 2.6 regarding the absence of the slow Task Builder. Version 2.7 of BASIC–PLUS–2 does not exhibit this problem.

A

Reporting Problems

This appendix describes the procedure for submitting a Software Performance Report (SPR). An SPR allows you to report any problems with or questions about your system directly to Digital.

An SPR can be used for:

- Software errors
- Documentation errors (when the Reader's Comments form is not appropriate)
- Follow-up on a previous SPR
- Questions
- Suggestions

An SPR cannot be used for:

- Software license and price policies
- Obvious hardware problems
- Logistical or clerical problems with kits, such as blank media, or failure to receive the *Software Dispatch*
- Problems with user-written software

In general, when you complete an SPR, use the following guidelines:

- Describe only one problem per form. This will facilitate a more rapid response because it allows the person answering the problem to concentrate more fully on that particular problem. One problem per form also helps simplify record-keeping.
- Define as accurately as possible the state of the system and circumstances when the problem occurred.
- Illustrate the problem with specific examples.
- If you report a documentation error, specify the title of the manual, and include the section and page number where the error occurred. Include a table or figure number if appropriate.

SPRs are assigned a priority of 1 to 5. An SPR that is assigned a priority of 1 receives the highest priority. Priorities are described as follows:

1. Most production work cannot be run.
   - Major system functions are unusable.
   - You cannot boot the system.
   - Necessary peripherals cannot be used.

2. Some production work cannot be run.
   - Certain functions are unusable.
   - System performance has declined.
   - Installation does not have excess capacity.
3. All production work can be run with some user impact.
   - Significant manual intervention is required.
   - System performance has declined.
   - Installation has excess capacity.
4. All production work can be run with no significant impact on user.
   - Problem can be patched or easily bypassed.
5. No system modifications are needed to return to normal production.
   - Suggestions are supplied.
   - Consultations are requested.
   - Errors in documentation are noted.

You can submit the following categories of SPRs:

- Problem/Error SPR

  This type of SPR contains a software problem. It is assigned a priority of 1 to 5. You receive an answer to this report.

- Suggested Enhancement

  This type of SPR contains a suggestion. It is assigned a priority of 5. You do not receive an answer to this report.

- Other

  This type of SPR contains a question or suggestion. It is assigned a priority of 5. You may or may not receive an answer to this report.

Please supply the following information (in machine-readable form where applicable) when you report a problem:

- CRASH—Supply a copy of the Executive task-build map, output from the console terminal, the SYSGEN saved-answer file, the Executive STB file, and the crash dump. If the crash is reproducible, accurately describe the details and supply a hard copy or user source code when necessary.
- DRIVERS—Supply controller/device information, software options, error log output, a copy of device registers, and a sample program.
- UTILITIES—Supply a copy of your terminal output, showing setup commands, before and after effects, and relevant file information.
- TASK BUILDER—Supply a copy of your terminal output command files, the task map, and a dump of the first few blocks of the task image.
- FILE SYSTEM—For a corrupted volume, supply output from the File Structure Verification Utility (VFY) and dump of the volume. For improper results, supply the error code, a file header dump, and a sample program.
- ERROR LOG REPORT GENERATOR (RPT)—Supply a copy of the report file generated by RPT, either on a hardcopy listing or on machine-readable media.

If a failure occurs when you are running privileged, add-on software (for example, the DECnet package), try to reproduce the failure without the additional software. Then, when you write the SPR, indicate how the system operated with and without the add-on software.

The SPR process is sometimes lengthy. Therefore, if you have a critical problem, contact your local Digital office. In the meantime, read the *Software Dispatch* for news on the operating system, which includes changes and problems other people have found and solved.

B

Applying Corrections to Source Files

Interim changes to the Executive, MCR, and device drivers are made by creating correction files that are processed by the Source Language Input Program (SLP). SLP generates a new copy of the modules that contain the errors by applying the corrections to the source file on the distribution kit. (See the *RSX–11M–PLUS Utilities Manual* for complete information on SLP.)

After you have applied the corrections and have obtained a new version of the file, *do not* delete the original source file. Interim changes that may be distributed later are cumulative and depend on the availability of the original sources.

B.1 Updating an Executive Source Module

To update an Executive source file (ABCDEF.MAC, for example), mount the disk on which you performed your system generation and create a SLP correction file named ABCDEF.COR in the directory [11,40]. (All the following instructions assume that you are working on the disk on which you performed your system generation.) Then, while your system is running under User Identification Code (UIC) [11,10], submit the correction file to SLP. For example, you could follow this sequence to create REQSB.MAC:

```
>SET /UIC=[11,40] Return
>EDI REQSB.COR Return
[Creating new file]
Input
REQSB.MAC;2/AU/-BF=REQSB.MAC;1
.
.
.
*EX
[Exit] Return

>SET /UIC=[11,10] Return
>SLP @[11,40]REQSB.COR Return
```

If the updated Executive module in your system is not a loadable driver, use the following procedure:

1. Assemble the new module, using the RSXMC.MAC file for the target system. For example, type the following commands:

   ```
   >SET /UIC=[11,24] Return
   >MAC REQSB,[11,34]REQSB/-SP=[1,1]EXEMC/ML,[11,10]RSXMC/PA:1,REQSB Return
   ```

2. Use the Librarian Utility Program (LBR) to replace the old version of the module in the RSX11M.OLB file on the target system.

   For example:

   ```
   >SET /UIC=[1,24] Return
   >LBR RSX11M/RP/-EP=[11,24]REQSB Return
   ```

3. Perform the following sections of SYSGEN:
   - Building the Executive and Drivers
   - Building the Privileged Tasks
   - Creating the System Image File

If the modified file in your system is a loadable device driver (ZZDRV.MAC, for example), the updated module can be replaced without rebuilding the Executive. Assemble the updated module and replace the resulting object file in the RSX11M.OLB file of your target system. For example, assume that [11,10] contains the RSXMC.MAC file resulting from your system generation and do the following:

```
>SET /UIC=[11,24] Return
>MAC ZZDRV=[1,1]EXEMC/ML,[11,10]RSXMC/PA:1,ZZDRV Return
>SET /UIC=[1,24] Return
>LBR RSX11M/RP/-EP=[11,24]ZZDRV Return
```

Use [200,200]ZZDRVBLD.CMD to rebuild the driver. If necessary, copy ZZDRV.TSK and ZZDRV.STB into the directory that corresponds to the system UIC, or the library UIC if the driver is vectored.

For example:

```
>ASN SY:=OU: Return
>TKB @[200,200]ZZDRVBLD Return
>SET /SYSUIC Return
SYSUIC=[g,m]
>SET /UIC=[g,m] Return
>PIP /NV=[1,54]ZZDRV.TSK,ZZDRV.STB Return
```

Use the Virtual Monitor Console Routine (VMR) to unload the old device driver and to load the new one. If the new driver is larger than the old one, it may not fit into the same locations as the old one. It may be necessary to unload and reload all of the loadable drivers in that partition to create enough room.

For example:

```
>VMR Return
Enter filename:RSX11M Return
VMR>UNL ZZ: Return
VMR>LOA ZZ: Return
VMR> Ctrl/Z
>RUN $SHUTUP Return
```

If the driver is vectored, you must use the /VEC switch with the UNL and LOA commands. In the preceding example, you would substitute the following VMR commands:

```
VMR>UNL ZZ:/VEC Return
VMR>LOA ZZ:/VEC Return
VMR> Ctrl/Z
```

Hardware boot the modified system.

B.2 Updating an MCR Source Module

To update the MCR source file SETOV.MAC, use the following procedure:

1. Create the SLP correction file [12,40]SETOV.COR and use it to update [12,10]SETOV.MAC. Assemble SETOV for the target system, as follows:

```
>SET /UIC=[12,40] Return
>EDI SETOV.COR Return
[Creating new file]
Input
SETOV.MAC;2/AU/-BF=SETOV.MAC;1
.
.
.
*EX Return
[Exit]

>SET /UIC=[12,10] Return
>SLP @[12,40]SETOV.COR Return
```

2. Assume that directory [11,10] contains the RSXMC.MAC file resulting from your target system generation and do the following:

```
>SET /UIC=[12,24] Return
>MAC SETOV=[1,1]EXEMC/ML,[11,10]RSXMC/PA:1,[12,10]SETOV Return
```

   All of the Task Builder command files output a map to the logical device MP, which must be assigned to NL or another device to avoid a diagnostic error message from the Task Builder.

3. If it was necessary to modify the MCR Task Builder command file (MCRBLD.CMD) during the last system generation, it may now be necessary to repeat those changes. To rebuild the secondary portion of MCR (. . . MCR) and replace the module SETOV, use the following procedure:

```
>SET /UIC=[1,24] Return
>LBR MCR/RP/NOEP=[12,24]SETOV Return
>PIP SETOV.OBJ;*/DE Return
>ASN SY:=MP: Return
>ASN SY:=IN: Return
>TKB @MCRBLD Return
>VMR Return
Enter filename:RSX11M Return
VMR>REM ...MCR Return
VMR>INS [3,54]MCR Return
VMR> Ctrl/Z
```

   To rebuild the MCR dispatcher (MCR . . .) and to replace the module MCRDIS, use the following procedure:

```
>SET /UIC=[1,24] Return
>LBR MCR/RP/NOEP=[12,24]MCRDIS Return
>PIP MCRDIS.OBJ;*/DE Return
>ASN SY:=MP: Return
>ASN SY:=IN: Return
>TKB @MCDBLD Return
>VMR Return
Enter filename:RSX11M Return
VMR>REM MCR... Return
VMR>INS [3,54]MCD/XHR=NO Return
VMR> Ctrl/Z
>RUN $SHUTUP Return
```

There is only one procedure for replacing an external MCR task. It involves the following steps:

1. Create the SLP file, apply it, and create the object file.
2. Incorporate the updated module into the task's object library.
3. Rebuild the task and install it in the system using MCR or VMR. Before using VMR, you must assign SY and LB to the disk containing the target system.

The following example replaces the module INSLB of the external MCR task INSTALL:

```
>SET /UIC=[1,24] Return
>LBR INS/RP=[12,24]INSLB Return
```

4. If it was necessary to modify the external task's Task Builder command file during the last system generation, it may be necessary at this time to repeat those changes:

```
>SET /UIC=[1,24] Return
>ASN SY:=MP: Return
>TKB @INSBLD Return
>VMR Return
Enter filename:RSX11M Return
VMR>REM ...INS Return
VMR>INS [3,54]INS/IOP=NO Return
VMR> Ctrl/Z
>RUN $SHUTUP Return
```

Hardware boot the system.

B.3 Updating a DCL Source Module

The procedure is different for the DCL task. There are two DCL object libraries: DCLR.OLB for modules in the root segment of the DCL task, and DCLO.OLB for modules in DCL's overlay segments. (One module, COMMAND, has versions in both the root and overlay libraries.) These libraries are located in directory [1,24] on your distribution kit. If you are not sure whether a module belongs in the root or the overlay library, use the Librarian Utility Program (LBR) to scan the module names in the libraries. For instance, to view the module names in DCLO.OLB, enter the MCR or DCL command shown next.

```
>LBR [1,24]DCLO/LI Return
```

```
$LIBRARY/LIST [1,24]DCLO Return
```

The modules you are most likely to need to modify are the DCL syntax tables, which are in DCLO.OLB. (All DCL syntax tables are in overlay segments.) See the *RSX–11M–PLUS and Micro/RSX System Management Guide* for more information on the structure of the DCL task and on the DCL task-building process.

File DCL.CMD, located in directory [23,24] on your distribution kit, can be used either to assemble DCL overlay modules and to rebuild DCL, or as a template for your own commands. The file contains its own instructions. Note that to use this command file directly, you must copy various files into directory [23,24] and appropriately modify any directories that they reference. File DCL.CMD also shows how to assemble the special module COMMAND.

The following process shows how to reassemble a DCL overlay module and rebuild DCL. The process is similar for root modules; simply specify DCLR rather than DCLO. See the DCL.CMD command procedure for details on rebuilding COMMAND.

If your command line interpreter (CLI) is MCR, use the following commands:

```
>SET /UIC=[23,24] Return
>MAC xxx=[11,10]RSXMC/PA:1,[23,10]DCLMAC/PA:1,xxx Return
>SET /UIC=[1,24] Return
>LBR DCLO/RP=[23,24]xxx Return
>TKB @DCLBLD Return
```

If your CLI is DCL, use the following commands:

```
$ SET DEFAULT [23,34] Return
$ MACRO [11,10]RSXMC/PASS:1,[23,10]DCLMAC/PASS:1,xxx Return
$ SET DEFAULT [1,24] Return
$ LIBRARY/REPLACE DCLO [23,24]xxx Return
$ LINK @DCLBLD Return
```

When you have built a new version of DCL.TSK, you must replace the old copy of DCL as a system CLI. First, any terminals whose CLI is DCL must be logged out or set to another CLI. Then, the following commands must be executed (note that these commands are for MCR only):

```
>CLI /ELIM=DCL Return
>REM DCL Return
>INS $DCL/CLI=YES Return
>CLI /INIT=DCL Return
```

The REMOVE and INSTALL commands are also valid VMR commands and can be used to modify your system's image on disk. However, the CLI /INIT command is for MCR only, and it must be put in your system's startup file (if it is not there already).

B.4 Applying Corrections to the File Systems (F11ACP)

All F11ACP updates begin by creating a SLP correction file in directory [13,40]. The following example patches a module called WTRN1.

1. Boot your system and log in to a privileged account, as follows:

   ```
   >HELLO SYSTEM Return
   Password:
   ```

2. If necessary, restore the required files from the distribution tape (the following example assumes that the files in [13,10] have been deleted, but the files in [1,24] have not):

   ```
   >UFD DB0:[13,10] Return
   >UFD DB0:[13,40] Return
   >BRU Return
   BRU>/NOINITIALIZE/SUPERSEDE/NOPRESERVE/BACKUP_SET:MPBL40SRC Return
   From: MM0:[13,10]F11PRE.MAC,WTRN1.MAC Return
   To: DB0: Return
   BRU--Completed
   BRU> Ctrl/Z
   ```

3. Create the correction file, as follows:

   ```
   >SET /UIC=[13,40] Return
   >EDT WTRN1.COR Return
   ```

4. Apply the correction, as follows:

```
>SET /UIC=[13,10] Return
>SLP @[13,40]WTRN1.COR Return
```

5. Assemble the corrected module with the Executive macro library, the Executive prefix file RSXMC.MAC, and the prefix file F11PRE.MAC, as follows:

```
>SET /UIC=[13,24] Return
>MAC WTRN1=[1,1]EXEMC/ML,[11,10]RSXMC,[13,10]F11PRE,WTRN1 Return
```

6. Replace the defective module in the file control processor (FCP) library, as follows:

```
>SET /UIC=[1,24] Return
>LBR FCP/RP=[13,24]WTRN1 Return
Modules replaced
WTRN1
```

All of the task-build command files require that the logical device MP be assigned to the appropriate device.

In the following example, xxx must be replaced by the 3-character designation for your desired FCP, that is, MDL or LRG.

1. Task build the new FCP by using the updated library as follows:

```
>ASN NL:=MP: Return

>TKB @FCPxxxBLD
```

2. Install the updated FCP in the system image as follows:

```
>RUN $VMR Return
Enter filename: RSX11M Return
VMR>REM F11ACP Return
VMR>INS [3,54]FCPxxx/PAR=GEN/IOP=NO/CKP=NO Return
VMR> Ctrl/Z
>RUN $SHUTUP Return
```

3. Reboot the system to place the new FCP in use.

B.5 Applying Corrections to the Reconfiguration Tasks CON and HRC

The following example illustrates how to patch module CNCMR for the CON task. Unless indicated otherwise, use a similar command sequence for correcting module HRONL for the HRC task.

This example assumes that:

- You have deleted the source files in [27,10].
- You have not deleted the object libraries and command files in [1,24] and [1,20].
- The object library for both CON and HRC is [1,24]OLR.OLB.
- The disk to which you will apply the patches is mounted Files–11 on drive DB0.
- The distribution tape is mounted foreign on MM0.

Perform the following steps to patch CNCMR:

1. Boot your system, and log in to a privileged account.
2. If necessary, restore the required files from the distribution tape as follows:

```
>UFD DB0:[27,10] Return
>UFD DB0:[27,40] Return
>UFD DB0:[27,24] Return

>BRU Return
BRU>/NOINITIALIZE/SUPERSEDE/NOPRESERVE/BACKUPSET:MPBL40SRC Return
From:MM0:[27,10]CNPRE.MAC,CNCMR.MAC Return
To:DB0: Return
BRU -- Completed
BRU> CTRL/Z
```

   If you were correcting module HRONL, you would substitute the following for the From: line in the previous command sequence:

```
From:MM0:[27,10]HRPRE.MAC,HRONL.MAC Return
```

3. Create the SLP correction file in UIC [27,40], as follows:

```
>SET /UIC=[27,40] Return
>EDT CNCMR.COR Return
```

4. Apply the SLP correction file to CNCMR as follows:

```
>SET /UIC=[27,10] Return
>SLP @[27,40]CNCMR.COR Return
```

5. Assemble the corrected CNCMR module by using the Executive macro library and the Executive prefix file RSXMC.MAC; you also use these when assembling a corrected HRC module. In addition, use either the prefix file CNPRE.MAC, if assembling a CON module, or HRPRE.MAC, if assembling an HRC module.

   For example:

```
>SET /UIC=[27,24] Return
>MAC CNCMR=[1,1]EXEMC/ML,[11,10]RSXMC/PA:1,[27,10]CNPRE,CNCMR Return
```

6. Replace the CNCMR object module in the OLR library as follows:

```
>SET /UIC=[1,24] Return
>LBR OLR/RP/NOEP=[27,24]CNCMR.OBJ Return
```

7. Task build CON by using the updated library as follows:

```
>ASN DB0:=IN: Return
>ASN DB0:=OU: Return
>ASN NL:=MP: Return

>TKB @CONBLD Return
```

Note

If you do not have the task-build command files [1,24]CONBLD.CMD and [1,24]CONBLD.ODL, or [1,24]HRCBLD.CMD and [1,24]HRCBLD.ODL, you must use the section of SYSGEN that rebuilds system-supplied tasks.

8. Install the updated reconfiguration utility in the system image as follows: For CON, type the following:

```
>RUN $VMR Return
Enter filename:RSX11M Return
VMR>REM ...CON Return
VMR>INS [3,54]CON Return
VMR> Ctrl/Z
```

For HRC, type the following:

```
>RUN $VMR Return
Enter filename:RSX11M Return
VMR>REM HRC... Return
VMR>INS [3,54]HRC/IOP=NO Return
VMR> Ctrl/Z
```

9. In order to use the new reconfiguration tasks, follow these steps:
   a. Shut the system down using the SHUTUP utility as follows:

```
>RUN $SHUTUP Return
```

   b. Reboot the system.

B.6 Applying FCS Corrections

Correcting the File Control Services (FCS) modules on an RSX–11M–PLUS system can be done by updating the source files, by assembling them, and by replacing modules in the system library, usually LB:[1,1]SYSLIB.OLB.

This process is complicated by the fact that there are three kinds of FCS, as follows:

- ANSI – Supports ANSI-format magnetic tape and big buffers.
- Non-ANSI – Does not support ANSI tape or big buffers.
- Multibuffered – Supports ANSI tape, big buffers, and multiple buffers

An FCS source file like CLOSE.MAC contains conditional assembly directives that can produce three different CLOSE objects, depending on the global symbols defined when CLOSE.MAC is assembled. These three different CLOSE objects correspond to the three kinds of FCS. Other FCS source files, like DELETE.MAC, have no such conditional assembly directives. They are only assembled one way; that is, only one DELETE object exists.

The SYSLIB.OLB file provided on the kits contains the ANSI FCS. Thus, this system library contains modules like CLOSE, assembled with the ANSI tape conditionals and big buffer conditionals enabled, and modules like DELETE, which have no such conditionals and are the same in any FCS.

An alternate system library called NOANSLIB.OLB is also provided. It contains an FCS that does not support ANSI tape, big buffers, or multiple buffers. Be sure that you know whether the system library on your system contains the ANSI FCS, or if it has been replaced with the non-ANSI or multibuffered FCS.

Details of the correcting procedure follow. MCR syntax is used throughout.

B.7 Updating the FCS Sources

The FCS source files are found in [50,10] on the kits. Updating a source is done by entering the correction file into [50,10], and by entering "SLP @filename" to apply the correction.

B.8 Assembling FCS

Assembling the updated source or sources can be done in either of two ways. One way is simple and time-consuming; the other is quick, but it must be done with great care. The simple way is to set your UIC to [50,24] and enter MAC @FCSASM. This will assemble every FCS variant properly, producing over 100 object files.

The other choice is to assemble only the sources that have been updated. This is not as simple, but it saves machine time. To assemble only a particular file, look at the five *.CMD files in [50,24] on the kit. Search the command files for all references to the file you want to assemble. A description follows of each command file and what you do with its contents.

1. FCSBOTH.CMD assembles files like DELETE.MAC, which contain no code specifically written to support or deny support to ANSI tape, big buffers, or multiple buffers. If the source file you have updated is mentioned in FCSBOTH.CMD, then it is assembled the same way regardless of which kind of FCS is in your system library. For example, the following lines assemble DELETE:

   ```
   [50,24]DELETE,[50,34]DELETE/-SP=-
   [50,10]FCSPRE,DELETE
   ```

   (FCSPRE.MAC defines necessary macros and global symbols.)

   To assemble DELETE.MAC, you should set your UIC to [50,24] and enter the following command line:

   ```
   >MAC [50,24]DELETE,[50,34]DELETE/-SP=[50,10]FCSPRE,DELETE
   ```


   As an alternative, you could put the command line in a file and enter "MAC @filename". A file like DELETE.MAC, which has no conditional assembly directives, is mentioned only in FCSBOTH.CMD.

2. FCSANSI.CMD assembles source files that contain ANSI tape, big buffer, or multibuffering conditionals. It produces objects for the ANSI kind of FCS. If your system library contains the ANSI FCS, and the source file you have updated is mentioned in FCSANSI.CMD, then you should use the command line you found in FCSANSI.CMD to assemble the source file. For example, the following lines from FCSANSI.CMD assemble CLOSE.MAC:

   ```
   [50,24]CLOSE.MTA,[50,34]CLOSE.MTA/-SP=-
   [50,10]FCSANSI/PA:1,FCSBIGBUF/PA:1,FCSPRE,CLOSE
   ```

   FCSANSI.MAC enables the ANSI tape conditional assembly directives, and FCSBIGBUF.MAC enables the big buffer conditionals. Note the sequence of input file names. It is absolutely imperative that the source file is the last file name specified, and that FCSPRE is next to it, in every FCS assembly. Otherwise, the proper conditionals will not be enabled, with potentially confusing and dangerous results. Any file that is assembled by FCSANSI.CMD is also assembled by FCSNOANSI.CMD and FCSMULBUF.CMD. Note the MTA file types above, which distinguish between the three kinds of CLOSE objects.

3. FCSNOANSI.CMD also assembles source files that contain ANSI tape, big buffer, or multibuffering conditionals. It produces objects for the non-ANSI kind of FCS. If your system library contains the non-ANSI FCS, and the source file you have updated is mentioned in FCSNOANSI.CMD, then you should use the command line you found in FCSNOANSI.CMD to assemble the source file. For example, this command line from FCSNOANSI.CMD assembles CLOSE.MAC:

```
[50,24]CLOSE.NMT,[50,34]CLOSE.NMT/-SP=-
[50,10]FCSPRE,CLOSE
```

4. FCSMULBUF.CMD assembles source files that contain ANSI tape, big buffer, or multibuffering conditionals. It produces objects that support all of these features. For example, this command line from FCSMULBUF.CMD assembles CLOSE.MAC:

```
[50,24]CLOSE.MBF,[50,34]CLOSE.MBF/-SP=-
[50,10]FCSANSI/PA:1,FCSMULBUF/PA:1,FCSBIGBUF/PA:1,FCSPRE,CLOSE
```

   FCSMULBUF.MAC enables the multibuffering conditionals.

5. FCSASM.CMD simply causes the other command files to be executed, to assemble FCS in all ways. If you are assembling single sources, ignore FCSASM.CMD.

More information is necessary to correctly assemble a few FCS sources that require additional prefix files. These sources are GET.MAC, PUT.MAC, OPEN.MAC, FINIT.MAC, and RDWRIT.MAC. GET.MAC can be assembled in the normal way (ANSI, non-ANSI, or multibuffered) to produce GET.MTA, GET.NMT, or GET.MBF. An additional file, GPSEQ.MAC, will enable conditional assembly directives in GET.MAC to produce GETSQ, the sequential GET module, for each kind of FCS. For example, this is the command line from FCSANSI.CMD, which produces GETSQ.MTA:

```
[50,24]GETSQ.MTA,[50,34]GETSQ.MTA/-SP=-
[50,10]GPSEQ/PA:1,FCSANSI/PA:1,FCSBIGBUF/PA:1,FCSPRE,GET
```

If you correct GET.MAC, be sure that you assemble it to produce objects GET and GETSQ for whatever kind of FCS your system library contains.

Similarly, GPSEQ.MAC will produce PUTSQ, the sequential PUT module, for each kind of FCS. This is the command line from FCSNOANSI.CMD, which produces PUTSQ.NMT:

```
[50,24]PUTSQ.NMT,[50,34]PUTSQ.NMT/-SP=-
[50,10]GPSEQ/PA:1,FCSPRE,PUT
```

If you correct PUT.MAC, be sure that you assemble it to produce objects named PUT and PUTSQ.

OPEN.MAC can be assembled in the normal way (ANSI, non-ANSI, or multibuffered) to produce OPEN.MTA, OPEN.NMT, or OPEN.MBF. Three additional prefix files exist: OPFID.MTA, OPFNB.NMT, AND OPENR.MAC.

OPFID.MAC produces the open-by-file-id module, OPFID. This is the command line from FCSANSI.CMD, which produces OPFID.MTA:

```
[50,24]OPFID.MTA,[50,34]OPFID.MTA/-SP=-
[50,10]OPFID/PA:1,FCSANSI/PA:1,FCSBIGBUF/PA:1,FCSPRE,OPEN
```

OPFNB.MAC produces the open-by-file-name block module, OPFNB. This is the command line from FCSANSI.CMD, which produces OPFNB.NMT:

```
[50,24]OPFNB.NMT,[50,34]OPFNB.NMT/-SP=-
[50,10]OPFNB/PA:1,FCSPRE,OPEN
```

OPENR.MAC produces the open module for resident libraries, OPENR. This is the command line from FCSMULBUF.CMD, which produces OPENR.MBF:

```
[50,24]OPENR.MBF,[50,34]OPENR.MBF/-SP=-
[50,10]OPENR/PA:1,FCSANSI/PA:1,FCSMULBUF/PA:1,FCSBIGBUF/PA:1,FCSPRE,OPEN
```

If you correct OPEN.MAC, be sure that you assemble it to produce objects OPEN, OPFID, OPFNB, and OPENR for whatever kind of FCS your system library contains.

FINIT.MAC and RDWRIT.MAC can be assembled to produce FINIT.OBJ and RDWRIT.OBJ. The prefix file FCSSUP.MAC enables conditionals to produce objects FINTSL.SUP and RDWRSL.SUP to be used in supervisor-mode libraries. These objects are necessary for correct execution of the user asynchronous system trap (AST) completion routines, which may be specified for FCS READ$ and WRITE$ functions. This is the line from FCSANSI.CMD that produces FINTSL.SUP:

```
[50,24]FINTSL.SUP,[50,34]FINTSL.SUP/-SP=[50,10]FCSSUP/PA:1,FCSPRE,FINIT
```

Object files FINTSL.SUP and RDWRSL.SUP contain the same entry point names as FINIT.OBJ and RDWRIT.OBJ. Therefore, they must be replaced in the system library with their entry points deleted so that they are only used when they are explicitly called when a supervisor-mode library is task built.

B.9 Replacing the FCS Object Modules

Set your UIC to [1,1], make a backup copy of the libraries, and use LBR to replace the corrected FCS objects that you have assembled. For example, suppose you have updated and assembled DELETE, PUT, and FINIT on a system with an ANSI FCS in the system library. You would then use the following procedure:

```
>SET /UIC=[1,1] Return
>PIP /NV=SYSLIB.OLB Return
>LBR SYSLIB/RP=[50,24]DELETE.OBJ,PUT.MTA,PUTSQ.MTA Return
>LBR SYSLIB/RP=[50,24]FINIT.OBJ,FINTSL.SUP/-EP Return
```

If you use NOANSLIB.OLB, also use the following procedure:

```
>PIP /NV=NOANSLIB.OLB Return
>LBR NOANSLIB/RP=[50,24]DELETE.OBJ,PUT.NMT,PUTSQ.NMT Return
>LBR NOANSLIB/RP=[50,24]FINIT.OBJ,FINTSL.SUP/-EP Return
```

If you want to incorporate the corrected modules into an FCSRES resident library or FCSFSL supervisor-mode library, first rebuild the library, and then rebuild every task that links to it with SYSGEN. If you rebuild and install a resident library and run a task that linked to the old resident library, then the task will call routines in the resident library at the wrong addresses. The results will be uncertain and potentially damaging, especially for privileged tasks.

Index

A

B

C

D

E

F

G

H

I

K

L

M

N

O

P

Q

R

S

T

U

V

W

X

Y

How to Order Additional Documentation

Technical Support

If you need help deciding which documentation best meets your needs, call 800-DIGITAL (800-344-4825) and press 2 for technical assistance.

Electronic Orders

If you wish to place an order through your account at the Electronic Store, dial 800-234-1998, using a modem set to 2400- or 9600-baud. You must be using a VT terminal or terminal emulator set at 8 bits, no parity. If you need assistance using the Electronic Store, call 800-DIGITAL (800-344-4825) and ask for an Electronic Store specialist.

Telephone and Direct Mail Orders

From	Call	Write
U.S.A.	DECdirect Phone: 800-DIGITAL (800-344-4825) FAX: (603) 884-5597	Digital Equipment Corporation P.O. Box CS2008 Nashua, New Hampshire 03061
Puerto Rico	Phone: (809) 781-0505 FAX: (809) 749-8377	Digital Equipment Caribbean, Inc. 3 Digital Plaza, 1st Street Suite 200 Metro Office Park San Juan, Puerto Rico 00920
Canada	Phone: 800-267-6215 FAX: (613) 592-1946	Digital Equipment of Canada Ltd. 100 Herzberg Road Kanata, Ontario, Canada K2K 2A6 Attn: DECdirect Sales
International	———	Local Digital subsidiary or approved distributor
Internal Orders[1] (for software documentation)	DTN: 241-3023 (508) 874-3023	Software Supply Business (SSB) Digital Equipment Corporation 1 Digital Drive Westminster, MA 01473
Internal Orders (for hardware documentation)	DTN: 234-4325 (508) 351-4325 FAX: (508) 351-4467	Publishing & Circulation Services Digital Equipment Corporation NR02-2 444 Whitney Street Northboro, MA 01532

[1]Call to request an Internal Software Order Form (EN–01740–07).

Reader's Comments

RSX–11M–PLUS
Release Notes

AA–H427J–TC

Your comments and suggestions help us improve the quality of our publications.

Thank you for your assistance.

I rate this manual's:	Excellent	Good	Fair	Poor
Accuracy (product works as manual says)	☐	☐	☐	☐
Completeness (enough information)	☐	☐	☐	☐
Clarity (easy to understand)	☐	☐	☐	☐
Organization (structure of subject matter)	☐	☐	☐	☐
Figures (useful)	☐	☐	☐	☐
Examples (useful)	☐	☐	☐	☐
Index (ability to find topic)	☐	☐	☐	☐
Page layout (easy to find information)	☐	☐	☐	☐

I would like to see more/less _____

_____

_____

What I like best about this manual is _____

_____

_____

What I like least about this manual is _____

_____

_____

I found the following errors in this manual:

Page	Description
_____	_____
_____	_____
_____	_____
_____	_____
_____	_____

Additional comments or suggestions to improve this manual:

_____

_____

_____

_____

For software manuals, please indicate which version of the software you are using: _____

_____

Name/Title _____ Dept. _____

Company _____ Date _____

Mailing Address _____

_____ Phone _____

--- Do Not Tear - Fold Here and Tape ---

digital™

No Postage Necessary if Mailed in the United States

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 33 MAYNARD MASS.

POSTAGE WILL BE PAID BY ADDRESSEE

DIGITAL EQUIPMENT CORPORATION
Information Design and Consulting
NUO1–1/G10
55 NORTHEASTERN BLVD.
NASHUA, NH 03062–3187

--- Do Not Tear - Fold Here ---