RSX-11M/RMS-11/RSX-11S
Release Notes

Order No. AA-2573H-TC

RSX-11M Version 4.1
RMS-11 Version 2.0
RSX-11S Version 4.1

digital equipment corporation · maynard, massachusetts

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

The following are trademarks of Digital Equipment Corporation:

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

ZK2255

CONTENTS

CONTENTS

PREFACE

MANUAL OBJECTIVES

This manual summarizes the new software features for RSX-11M Version 4.1, RSX-11S Version 4.1, and RMS-11 Version 2.0. It also includes information on problems and restrictions applicable to the new software and incompatibilities with the previous version. Therefore, it is important to read this document before generating and using your system.

INTENDED AUDIENCE

The RSX-11M/RMS-11/RSX-11S Release Notes are intended for all users of the RSX-11M or RSX-11S operating systems.

STRUCTURE OF THE DOCUMENT

CHAPTER 1   RSX-11M VERSION 4.1 RELEASE NOTES

Section 1.1   New Features RSX-11M V4.1

Describes briefly the major new features of RSX-11M V4.1

Section 1.2   Summary of Technical Changes

Provides brief summaries of the new software features and refers you to the appropriate manuals in the documentation set for complete information. Any changes made to the documentation since V4.0 including new, revised, or updated manuals, are also indicated.

Section 1.3   Notes on Software

Describes the incompatibilities between the previous and the new versions of RSX-11M, and provides additional miscellaneous information pertinent to the successful performance of the software.

Section 1.4   Reporting Problems and Errors

Provides information on reporting any problems you encountered with the new software, and describes procedures for applying corrections to files on the source disk.

Section 3.2   Notes on Software

Describes the incompatibilities between the previous
and the new versions of RSX-11S, and provides
additional miscellaneous information pertinent to the
successful performance of the software.

Section 3.3   Notes to the RSX-11S V4.1 System Generation and
Installation Guide

Describes errors and omissions in the RSX-11S V4.1
System Generation and Installation Guide and refers you
to the appropriate places in the documentation to make
corrections.

Section 3.4   SYSGEN Dialogue from an RSX-11M Host

Contains an example of an RSX-11S V4.1 SYSGEN on an
RSX-11M V4.1 host system.

Section 3.5   SYSGEN on an RSX-11M-PLUS Host

Briefly describes how to perform an RSX-11S V4.1 SYSGEN
on an RSX-11M-PLUS host system.

Section 3.6   SYSGEN on a VAX/VMS Host

Briefly describes how to perform an RSX-11S V4.1 SYSGEN
on a VAX/VMS host system.

CHAPTER 1

RSX-11M VERSION 4.1 RELEASE NOTES

CHAPTER 1

RSX-11M V4.1 RELEASE NOTES

1.1 NEW FEATURES FOR RSX-11M V4.1

RSX-11M Version 4.1 software is an update of the disk-based RSX-11M operating system. The major new features for RSX-11M V4.1 include the following:

- Software Performance Monitor support

- Support for Micro/PDP-11

- New device support

- Multivolume backups for disks

- RMS-11 Version 2.0

1.1.1 RSX-11M Executive

The following items are changes to the RSX-11M V4.1 Executive:

- Some Executive routines have been moved out of the ERROR module and into an Executive Common. This change only affects systems with both error logging support and Executive Common support. See Section 1.3.8 for more information.

- The Software Performance Monitor (SPM-11M) hooks are integrated into the RSX-11M V4.1 Executive. As a result, you do not need to perform another SYSGEN to install SPM-11M.

- The RSX-11M V4.1 Executive no longer supports attaching a device with IQ.X functionality.

1.1.2 SYSGEN Features

The following SYSGEN features are new for V4.1:

- RSX-11M SYSGEN must be done on a mapped host system.

- To do a SYSGEN, you must have a host system with at least 64K words of memory.

- A mapped target system is now the default target system.

- A new question asks whether you want RT-11 emulation support.

- Devices with the mnemonic DU are now valid crash devices. As with all devices, crash support is restricted to removable-media devices.

- A new question asks whether you want Software Performance Monitor (SPM-11M) support. SPM-11M, which must be purchased separately, provides data on system usage. This question is always asked (even if you choose the Standard Function System). If you answer yes, the SPM-11 hook points are included in the system. You can then install the separate SPM-11M layered product.

- Micro/PDP-11 has been added to the list of supported processors.

- RSX-11M V4.1 distribution kits are now in BRU format.

- RSX-11M is no longer distributed on RK05s.

- The stand-alone DSC system (DSC64K) and the stand-alone BAD system (BADSYS) have been removed from the distribution kits.

- RSX-11M V4.1 SYSGEN now supports all the new devices listed in Section 1.2.5.

1.1.3 Backup and Restore (BRU)

BRU has three new features for RSX-11M V4.1: multivolume backups for disks, a new sort algorithm, and additional device support. See Section 1.2.4.1 for additional information.

1.1.4 MCR and VMR

RSX-11M now allows you to set the round-robin scheduling interval and priority using either MCR or VMR.

RSX-11M now allows you to set the interval length and priority range for Executive swapping. See Section 1.2.2 (MCR) and Section 1.2.7.1 (VMR) for more information.

MCR supports all the new devices listed in Section 1.2.5.

1.1.5 Task Builder (TKB)

The following is an abbreviated description of the functionality of the Task Builder for RSX-11M V4.1:

- The Autoload entry point has been modified. This entry point is now accessed by an indirect reference through the Overlay Run-Time System's impure area rather than by direct references in the Autoload vector.

- The Symbol Table File (.STB) now contains two forms of symbol definition. To maintain backward compatibility, all autoloadable symbols are entries in the global symbol directory and the vector itself is defined in associated text records. Additionally, a new internal symbol directory record, which is TKB-generated, supports selective inclusion

of autoloadable symbols for conventional tasks as well as the information needed to generate autoload vectors for I- and D-space tasks in RSX-11M-PLUS systems.

● Autoload vectors supplied by the symbol table (.STB) files of resident libraries are now selectively included in the user's task image. Previously, all autoload vectors resulting from autoloadable symbols were present in the task image. Now, only referenced symbols result in autoload vectors that are present in the task image.

● Libraries created with older versions of TKB do not have the ISD records in the .STB file that newer versions of TKB use to create autoload vectors. Therefore, TKB must include vectors from the .STB file for every entry point in the library.

● The rules concerning cluster libraries have changed so that the first library of the cluster, the default library, can now have a non-null root.

● The following option is new in TKB:

DSPPAT   Allows object-level patching of a conventional task or the D-space part of an I- and D-space task.

● The following switch is new in TKB:

/EL   Specifies the maximum possible size for the library according to the size specified in the PAR option. The switch specifies a larger library virtual address range than is actually present in the library to allow RMS to map its vectored library segments.

1.1.6 **New Device Support**

RSX-11M V4.1 supports a number of new devices. See Section 1.2.5 for a complete list.

1.1.7 **Support for the LA50, LA100, and LN01 Printers**

Support has been added to RSX-11M/M-PLUS for the LA50, LA100, and LN01 printers. These printers are capable of printing either 132 or 80 columns on 8 1/2 inch wide paper.

You can select support for these printers while performing SYSGEN. During SYSGEN, this support is a task-build option for the queue manager (QMG) and the line printer processor (LPP), and it is the default.

In order to use an LA50, LA100, or LN01 printer, you set the printer type when you initialize the print processor by using a form type in the command line. The valid form types are as follows:

Form types	Printer type
125.	LA50
126.	LA100
127.	LN01

For example, if you have a LA50 printer connected to terminal line
TT6:, you use the following command to initialize the print processor.

     (MCR)   QUE TT6:/SP/FO:125./FL:1/LOWER

     (DCL)   INITIALIZE/PROCESSOR TT6:/FORMS:125/FLAG:1/LOWER

When you wish to print a file, select your desired printer mode by
including one of the following form types in the command line:

Form types	Printer mode
3.	132-column, draft quality. Also landscape mode for LN01.
4.	80-column, letter quality. Also portrait mode for LN01.
5.	132-column, letter quality. Also landscape mode for LN01.
6.	80-column, draft quality. Also portrait mode for LN01.

For example, to print the 132-column draft quality TEST.LST file (that
is, a normal listing file) and the 80-column letter quality TEST.TXT
file on an LA50, use the following commands. Note that these commands
assume that you have assigned the default print queue, PRINT, to the
processor handling the LA50.

     (MCR)   PRI /FO:3.=TEST.LST
                    PRI /FO:4.=TEST.TXT

     (DCL)   PRINT/FORM:3 TEST.LST
                    PRINT/FORM:4 TEST.TXT

Modifying or Removing Support for the Printers

The printer options are controlled by task-build parameters in the
build files for the queue manager and the line printer processor.
These build files are, respectively, [1,20]QMGBLD.BLD and
[1,20]LPPBLD.BLD on the distribution kits. You can modify these prior
to building the queue manager and the line printer processor during
SYSGEN. You can modify these files for the following reasons:

1. You can change the form types that are used to represent
   printer types. To do so, modify the GBLDEF options that
   define the symbols Q$LSPS and Q$HSPS. The comments in the
   build files explain the values of the symbols. The normal
   defaults are 125. through 127.

2. You can change the form types that are used to represent
   document types. To do so, modify the GBLDEF options that
   define the symbols Q$LSPF and Q$HSPF. The comments in the
   build files explain the value of the symbols. The normal
   defaults are 3. through 6. If you modify the range of form
   types, you must change the GBLPAT options that define the
   forms to include offsets appropriate for the new form types.
   (See the comments in the command file regarding form
   definition.)

3. You can remove the support for these printers altogether.
   This may be necessary if you do not have any of these
   printers and if you use form types in the range 3. through
   6. or 125. through 127. In order to remove the support,
   set the values of Q$LSPS and Q$LSPF to 1 and the values of
   Q$HSPF and Q$HSPS to 0. You may also wish to remove the four
   GBLPAT definitions for the form types 3. through 6.

Note to LA100 Users

> If you have a LA100 printer and you use
> it for standard 132-column, 14 1/2 inch
> paper, you probably will want to
> establish it as your standard printer
> --with form type 0--and not use this
> support. This support is intended
> primarily for applications including all
> three printers and involving 8 1/2 inch
> wide forms.

1.1.8 Files-11 ACP

Two new features have been added to Files-11 (F11ACP) in RSX-11M V4.1
to support RMS-11 V2.0:

- F11ACP supports new file attribute codes (creation, revision, expiration, and backup dates), which read and write the data in the file header using a 64-bit date format. F11ACP performs a computation to convert the date between 64-bit binary and ASCII form.

- F11ACP also supports the $SEARCH function of RMS-11 V2.0 by providing character wildcarding assistance. This support substantially reduces processing time when portions of a filename include wildcards.

Because of these features, you cannot use RMS-11 V2.0 on a version of
RSX-11M prior to V4.1.

1.1.9 PDP-11 PASCAL/RSX

RSX-11M V4.1 includes DCL support for PDP-11 PASCAL. Note however,
that PDP-11 PASCAL is scheduled to be released at a date later than
that of RSX-11M V4.1. Therefore, until PDP-11 PASCAL/RSX is available
on your system, the use of the PASCAL command will result in the
informational message "PASCAL -- Sorry task not installed."

RSX-11M V4.1 lists PASCAL as one of the available HELP topics.
However, until PDP-11 PASCAL/RSX is released and becomes available on
your system, typing "HELP PASCAL" will result in the message
"PASCAL currently is unavailable on your system." PASCAL commands will
be documented in the PASCAL User's Guide, distributed at the time of
the PDP-11 PASCAL/RSX release. Complete HELP files for PDP-11
PASCAL/RSX will be provided with the PASCAL distribution kit at the
time of its release.

1.2 SUMMARY OF TECHNICAL CHANGES

The following sections list the new features for RSX-11M as described
in your documentation. A summary of technical changes also appears in
the preface of each manual.

The title of each manual is followed by a letter indicating whether
the manual is a NEW (N), REVISED (R), or UPDATED (U) version.

1.2.1 RSX-11M System Generation and Installation Guide (R)

The following new options and features have been added to the RSX-11M system generation and installation procedure:

- SYSGEN must be performed on a mapped host system with at least 64K words of memory.

- SYSGEN no longer supports the RK05 distribution kit.

- SYSGEN now supports SPM-11M.

- SYSGEN now supports RA60 removable media and RA81 fixed media under the UDA50 hardware controller.

- SYSGEN now supports RC25[1] fixed/removable media.

- SYSGEN now supports RD51 fixed media and RX50 floppy disk drives under the RQDX1 hardware controller.

- SYSGEN now supports TSV05 and TU80 magnetic tape drives under the TS11 hardware controller.

- SYSGEN now supports the LN01, LP07, LP26, and LP27 line printers under the LP11 hardware controller.

- SYSGEN now supports the DHV11 hardware controller.

- The distribution kits now only contain a mapped baseline system.

- The default for the target system configuration is now mapped.

- SYSGEN will always install the V4.1 TKB from the distribution kit with the task name ...TKL.

- SYSGEN now allows DU devices (RA60, RC25, and RX50 only) to be chosen as the crash dump device.

- In an RL01/02 PREPGEN, the EXCPRV disk is mounted by SYSGEN.

- The RSX-11M magnetic tape distribution kits are now in BRU format.   Also, BRU is now used to copy all the RSX-11M V4.1 distribution kits.

- RMS-11 V2.0 is now included in the RSX-11M V4.1 distribution kit.

1.2.2 RSX-11M/M-PLUS MCR Operations Manual (U)

The following are new MCR features for RSX-11M V4.1:

- MCR commands now support the following new devices: RA60, RA81, RC25, RX50, RD51, TSV05, and TU80.

- The following MCR SET keywords have been modified:

---

1. There are a number of references in the RSX-11M/M-PLUS manuals, including these Release Notes, to the RC25 device. This device is, as of the first publication date of the manuals, not available for sale or distribution; nor is its mention in the documents a commitment by DIGITAL to sell or distribute this device.

- /RNDC - Defines the length of the Executive round-robin scheduling interval.

- /RNDH - Defines the highest priority that will be considered for Executive round-robin scheduling.

- /RNDL - Defines the lowest priority that will be considered for Executive round-robin scheduling.

- /SWPC - Defines the length of a single Executive swapping interval.

- /SWPR - Defines a new priority range for Executive swapping.

- The /REMOTE and /SPEED keywords now support new terminal speeds and speed combinations for the DHV11 multiplexer. The new speeds are 75, 110, 134.5, 150, 300, 600, 1200, 1800, 2000, 2400, 4800, 9600, and 19200.

- The /TERM keyword now supports the LA50. For more information, see Section 1.5.1 of the RSX-11M/RMS-11/RSX-11S Release Notes.

● The following error message is new to the MCR REDIRECT command:

    TT redirect error

This message means that an attempt was made to redirect a terminal (TTn:) to a null device.

1.2.3 RSX-11M/M-PLUS Command Language Manual (R)

This revision of the RSX-11M-PLUS Command Language Manual incorporates changes and additions listed in the following sections.

1.2.3.1 COBOL-81

1.2.3.1 **COBOL-81** - The following lists the new features and modifications to DCL COBOL-81 commands:

● COBOL-81, rather than COBOL-11, is now the default compiler.

● COBOL-C11 invokes the current COBOL-11 compiler to perform the compilation.

● COBOL[/C81] invokes the COBOL-81 compiler to perform the compilation.

DCL command line for COBOL-81:

    COBOL[/C81] [/qualifier[s] filespec [/qualifier[s]]

The DCL qualifiers and their MCR translation follow. A (D) indicates the default.

DCL Qualifier	MCR Syntax
/OBJECT[=file] (D)	COBOL file,[list],[diag]=source
/NOOBJECT	COBOL ,[list],[diag]=source

Specifies whether an object file is created.

```
/LIST[=file]                        COBOL [obj],file,[diag]=source
/NOLIST (D)                         COBOL [obj],,[diag]=source
```

   Specifies whether a compile listing is created.


```
/DIAGNOSTICS[=file]                 COBOL [obj],[list],file=source
/NODIAGNOSTICS (D)                  COBOL [obj],[list]=source
```

   Specifies whether a diagnostics file is created.


```
/OVERLAY_DESCRIPTION                /BLD
/NOOVERLAY_DESCRIPTION (D)          /-BLD
```

   Specifies whether an indirect command file (.CMD) and an
   Overlay Descriptor Language (.ODL) file are produced. If you
   use the LINK/C81 command to task-build a program, do not use
   this qualifier. LINK/C81 also produces a .CMD and an .ODL
   file.


```
/CODE:CIS                           /CIS
/CODE:NOCIS (D)                     /-CIS
```

   Specifies whether the compiler should use CIS instructions in
   the object code it produces.


```
/CROSS_REFERENCE                    /CRF
/NOCROSS_REFERENCE (D)              /-CRF
```

   Specifies whether cross-reference tables should be appended to
   the compile listing.


```
/ANSI_FORMAT                        /CVF
/NOANSI_FORMAT (D)                  /-CVF
```

   Indicates whether the source program is in conventional ANSI
   format or DIGITAL's terminal format.


```
/DEBUG                              /DEB
/NODEBUG (D)                        /-DEB
```

   Specifies whether the symbolic debugger is used. If this
   qualifier is used, the LINK/C81/DEBUG command must also be used
   to link the file.


```
/SHOW                               /MAP
/SHOW:MAP                           /MAP
/SHOW:NOMAP (D)                     /-MAP
/NOSHOW                             /-MAP
```

   Indicates whether the Data Division and Procedure Division
   offset maps are to be included in the list file.


```
/TRUNCATE                           /TRU
/NOTRUNCATE (D)                     /-TRU
```

   Indicates whether the compiler is to perform decimal truncation
   on the values of COMP data items.

```
/CHECK                          /PER/BOU
/NOCHECK (D)                    /-PER/-BOU
/CHECK:ALL                      /PER/BOU
/CHECK:NONE                     /-PER/-BOU
/CHECK:BOUNDS                   /BOU
/CHECK:NOBOUNDS                 /-BOU
/CHECK:PERFORM                  /PER
/CHECK:NOPERFORM                /-PER
```

Deals with run-time error checking. /CHECK enables subscript
(index) range checking and nested PERFORM checking. /NOCHECK
suppresses both of these. /CHECK:NOPERFORM and /CHECK:NOBOUNDS
suppress each check individually.


```
/NAMES:aa                       /KER:aa
```

Tells the compiler to use the two alphanumeric characters
specified as the PSECT kernel for this program.


```
/WARNINGS (D)                   /INF
/NOWARNINGS                     /-INF
/WARNINGS:INFORMATIONAL         /INF
/WARNINGS:NOINFORMATIONAL       /-INF
```

Specifies whether the compiler should issue informational
diagnostics during the compilation.


```
/SKELETON (D)                   <no translation>
/NOSKELETON                     /-SKL
```

Specifies whether a skeleton Overlay Description Language
(.SKL) file is created.


```
/SUBPROGRAM                     /SUB
/NOSUBPROGRAM (D)               <no translation>
```

Indicates whether this program is a subprogram.


```
/TEMPORARY:device               /TMP:device
```

Tells the compiler to store its temporary work files on the
specified device.

1.2.3.2 **LINK/C81 Command** - The LINK/C81 command links COBOL-81 object
files to produce a task image. The command will not work with
versions of COBOL-81 prior to V4.0.

DCL LINK/C81 command line:

    LINK/C81[/qualifier(s)] [filespec(,s)]

    LINK/COBOL[/qualifier(s)] [filespec(,s)]

The new DCL qualifiers for this command follow.

/[NO]FMS
/FMS:NORESIDENT

> The /FMS qualifier causes LINK/C81 to include Forms Management Services (FMS) library support in your task image. You must use this qualifier if you call FMS routines in your program. /NOFMS tells LINK/81 not to include FMS support. This is the default.

> The /FMS:NORESIDENT qualifier causes LINK/C81 to include support for a non-memory-resident FMS library in your task image.

/OTS:[NO]RESIDENT

> The /OTS:RESIDENT qualifier includes memory-resident OTS in your task image.

/RMS:[NO]RESIDENT
/NORMS:NORESIDENT

> The /RMS:RESIDENT qualifier creates a reference to the shared RMS-11 memory-resident library, RMSRES. This library includes input and output support for sequential, indexed, and relative file organizations.

/[NO]MAP

> The /MAP qualifier causes LINK/C81 to produce a Task Builder map file with the file type .MAP. /NOMAP is the default.

/[NO]DEBUG

> The /DEBUG qualifier tells LINK/C81 to include the COBOL-81 Symbolic Debugger in your task image. You must have assembled the object module with the /DEBUG switch. /NODEBUG is the default. You must have assembled the object module with the /DEBUG switch.

1.2.3.3 **CORAL 66** - RSX-11M systems support CORAL 66, a high-level block-structured programming language. Following is the format and descriptions of the qualifiers available with CORAL 66.

Format:

> CORAL[/qualifier] filespec[s]

Command qualifiers:

> /[NO]CHECK

>> The /CHECK qualifier specifies that the code generated will check that all subscripts used in array and switch references are within bounds. The /NOCHECK qualifier is the default.

> /CODE:arg
>> EIS
>> FIS
>> FPP
>> PIC

>> The /CODE qualifier allows you to choose (at compile time) the instruction set for which the code is to be generated regardless of which machine is used to compile the source program.

/[NO]EXTEND_SOURCE

>The /NOEXTEND_SOURCE qualifier causes the compiler to process the first 72 characters of source input per line. The default is the /EXTEND_SOURCE which causes all characters in an input line to be processed.

/[NO]LIST

>The /LIST qualifier allows you to specify whether a compiler listing should be generated. The default is the /NOLIST qualifier which causes no compiler listing to be generated.

/[NO]MACHINE_CODE

>The /MACHINE_CODE qualifier specifies that you want the compiler listing to include a representation of the machine code generated. The default is the /NOMACHINE_CODE qualifier.

/NAMES:x

>The /NAMES qualifier causes the first character of compiler-generated program-section names to be the character 'x', in order to make them unique for this compilation.

/[NO]OBJECT[:filespec]

>The /OBJECT qualifier allows you to specify whether or not the compiler must generate an object module. The default is the /OBJECT qualifier, which generates an object file.

/[NO]OPTIMIZE[[:LEVEL]:n]
/OPTIMIZE=n

>The /OPTIMIZE qualifier allows you to optimize the function of CORAL 66 source programs.

/READ_ONLY[:arg]
>>>>ALL
>>>>NONE
>>>>PURE_DATA

>The /READ_ONLY qualifier allows you to alter the READ-ONLY or READ-WRITE attributes of program sections.

/[NO]SHOW[:arg[,s]]
>>>>ALL
>>>>EXPANSION
>>>>NONE
>>>>OVERRIDE
>>>>SOURCE
>>>>STATISTICS
>>>>SYMBOLS

>The /SHOW and /NOSHOW qualifiers specify elements you want included in or excluded from the listing file.

/[NO]STANDARD

>The /STANDARD qualifier highlights non-IECCA keywords in the listing as warning messages. The default is the /NOSTANDARD qualifier, which suppresses this function.

/TEST[:n]

>   The /TEST qualifier allows you to retain or omit
>   declarations and statements from a particular compilation
>   without editing the source text.

/[NO]TRACEBACK

>   The /TRACEBACK qualifier generates additional code to print
>   source-file and line-number information in run-time error
>   messages. The /NOTRACEBACK qualifier suppresses this
>   function and is the default.

/WIDTH:n

>   The /WIDTH:n qualifier allows you to set the listing file to
>   a specific width in the decimal range 8 through 132. The
>   /WIDTH:132 qualifier is the default.

1.2.3.4 **BASIC-PLUS-2** - BASIC-PLUS-2 is now the default BASIC
compiler. The compiler can create Task Builder Command and Overlay
Descriptor Language files and a listing file in addition to the object
module file for the program it compiles. (Note that BASIC/B11 runs
BASIC-11.)

1.2.3.5 **FORTRAN-77** - FORTRAN IV-PLUS has been superseded by
FORTRAN-77, although both compilers are supported by DCL (as well as
FORTRAN-IV). Qualifiers added for FORTRAN-77 are as follows:

>   /[NO]STANDARD:arg
>   ALL
>   NONE
>   SOURCE
>   SYNTAX

Directs the compiler to look in your source code for extensions
to ANSI standard FORTRAN at the full language level.

1.2.3.6 **BACKUP Command** - Two new qualifiers have been added to the
BACKUP command and several qualifiers have new functions because of
multivolume disk backup and restore operations. They are as follows:

● /APPEND

>   Appends new data to a tape or disk that already has one or
>   more backup sets.

● /DIRECTORY

>   Directs BACKUP to create UFDs (if they do not already exist)
>   on a mounted output volume, then copy into the UFDs the files
>   from the same UFD on the input volume.

● /IMAGE:arg
>   SAVE
>   RESTORE

>   Specifies that you want to do a multivolume disk-to-disk
>   backup or restore operation.

- /SAVE_SET

    Specifies the name of the backup set to be placed on tape or
    disk.   For tapes and unmounted disks, the default name is the
    volume name for the disk being backed up. For mounted input
    or output disks during an image backup or restore operation,
    you can specify the full backup set name with SAVE_SET.

1.2.3.7 **SHOW   Processortype   Command** – The   DCL   SHOW   processortype
command line has been modified so that you issue the command line as
follows:

        SHOW processortype processorname[/qualifier]
             BATCH
             CARDREADER
             DEVICE
             INPUT
             PRINTER
             PROCESSOR

The SHOW processortype command displays information about processors,
card readers, batch processors, and other devices under the control of
the Queue Manager.

CARDREADER and INPUT are synonyns.   DEVICE   refers   to   all   nonbatch
output processors and is a synonym for PRINTER. BATCH refers to all
batch processors.   (Batch   processing   is   available   only   on
RSX-11M-PLUS.)

A simple SHOW PROCESSOR command displays all processors in the system.

1.2.3.8 **Miscellaneous   Items** – The   following   lists   various
modifications to DCL for RSX-11M V4.1:

- EDT is now the default editor.

- The following qualifiers have been modified:

    - /CHECKPOINT_FILE and /WIDTH

        These qualifiers when used with SET DEVICE now accept
        decimal numbers by default.

- The /HFILL and /VFILL qualifiers for the SET TERMINAL and SHOW
  TERMINAL commands have been eliminated. However, the existing
  /CRFILL and /LFFILL qualifiers perform the same functions,
  respectively, as the two removed qualifiers.

- The INITIALIZE/QUEUE command has a new qualifier, /NOWARNINGS.
  This qualifier inhibits error messages.

1.2.4 **RSX-11M/M-PLUS Utilities Manual (U)**

Following are the additions and changes for the Backup and Restore
Utility (BRU) and the Task/Image Patch Program (ZAP).

1.2.4.1 **Backup and Restore Utility (BRU) (R)** - BRU for RSX-11M V4.1 has three new features:

● BRU now does multivolume backups. You can copy a single disk onto multiple smaller disks. However, when you do this, the smaller disks are in BRU format, not Files-11. This means that before you can read what is on the smaller disk, you must restore it with BRU. This process is similar to magnetic tape backup and restore operations.

- To backup and restore a single disk to multiple disks, you use the following new BRU switch:

        BRU /IMAGE:SAVE
                 :RESTORE

  Specifies that you want to do a multiple disk-to-disk backup or restore operation.   If you are doing a backup operation, you must specify the SAVE option on the command line.   If you are doing a restore operation, you must specify the RESTORE option on the command line.

Because of the addition of multivolume disk backups, the meanings of the following BRU qualifiers have changed.

- /APPEND

  Directs BRU to append a backup set from the input disk volume to the last backup set on the output tape, or on the output disk if you are using the /IMAGE qualifier.

- /BACKUP_SET:name

  Specifies the name of the backup set to be placed on tape or disk.   For a mounted input or output disk during an image backup or restore operation, you can specify the full backup set file name with the /BACKUP_SET qualifier.

- /DIRECTORY

  Lists at your terminal the backup set names or files on the specified tape or disk volume.

- /INITIALIZE

  Specifies that you want to initialize the output disk during a tape-to-disk or disk-to-disk operation.

- /INVOLUME

  Specifies the volume label of the input disk.

- /MOUNTED

  Allows you to back up files from a disk that is mounted as a Files-11 volume.

- /NEW_VERSION

  Resolves file specification conflicts that occur during either backup or restore operations to a mounted disk.

- /SUPERSEDE

  Specifies that when file specifications on the mounted
  output volume are identical to the file specifications on
  the input volume, the file on the output volume is deleted
  and replaced with the file from the input volume.

● BRU has a new sort algorithm, which increases the speed of the
  sorting procedure.

● In Version 4.1, if you use the /MOUNTED qualifier when the
  input device is a tape, BRU issues a syntax error message.
  The purpose of this error message is to clarify to the user
  that tapes cannot be mounted as Files-11 volumes. Previously,
  BRU ignored this qualifier if the user issued it for a tape
  input device.

● New examples have been added to BRU. They are as follows:

  - Disk-to-disk multivolume backup operations.

  - Disk-to-disk multivolume restore operations.

  - Disk-to-disk multivolume backup and append operation.

  - Disk-to-disk multivolume restore operation for restoring an
    appended backup set.

● New error messages for multivolume backup and restore
  operations have been added to the BRU chapter.

● BRU now supports the following additional devices:

  Mnemonic   **Type**

  MS   TSV05/TU80 magnetic tape

  EM   ML11 electronic memory

  DU   RA60/RA81/RC25/RD51/RX50 disk

● Changes have been added to bootstrapping stand-alone BRU for
  RSX-11M.

  On RSX-11M, you can now find BRU64K.SYS and BRU64K.STB in UFD
  [1,51] on the following disk volumes:

      BIG DISK KIT - RSXM35

      RK06/RK07 KIT - CLISRC

      RL01/RL02 KIT - RLUTIL

1.2.4.2 **Task/File Patch Program (ZAP)** - A new section has been added
to Chapter 15 (ZAP) describing the use of the /LI qualifier with I-
and D-space tasks and a new description of the /LI switch and
Multiuser Task Image files has been added.

1.2.5 RSX-11M/M-PLUS I/O Drivers Reference Manual (R)

The following new devices are supported by RSX-11M V4.1.

Supported terminal devices:

Type	Mnemonic
LQP02 letter quality printer	TT
LA50 personal printer	TT
DHV11 multiplexer	YH

Supported disks:

Type	Mnemonic
RA81 fixed media disk	DU
RA60 removable media disk	DU
RC25 fixed media/removable cartridge disk	DU
RD51 fixed media disk	DU
RX50 floppy disk	DU

Supported magnetic tapes:

Type	Mnemonic
TU80 magnetic tape	MS
TSV05 magnetic tape	MS

Supported line printers:

Type	Mnemonic
LN01 line printer	LP
LP07 line printer	LP
LP26 line printer	LP
LP27 line printer	LP

1.2.6 RSX-11M/M-PLUS Task Builder Manual (R)

Following are the major changes documented in the Task Builder manual:

- New Switch

  - /EL -- Extend Library

    This switch specifies the maximum possible size for the
    library according to the size specified in the PAR option.
    The switch specifies a larger library virtual address range
    than is actually present in the library to allow RMS to map
    its vectored library segments.

● New Options

  - DSPPAT -- Allows object-level patching of a conventional task or the D-space part of an I- and D-space task.

● Changed Options

  - ABSPAT -- Allows object-level patching of a conventional task or the I-space part of an I- and D-space task.

● New Error Messages

  Module module-name contains incompatible autoload vectors

● Changed Error Messages

  Original: Lookup failure resident library file

  Revised: Lookup failure resident library file - filename.ext

● Miscellaneous Technical Changes

  - Autoload vectors for conventional tasks have changed. The call to $AUTO is now made indirectly through .NAUTO in the overlay impure area.

  - You now can write an .ODL specification by applying certain rules to the virtual address space diagram.

  - The overlay run-time system routines have changed size from the previous release.

  - For versions of TKB that support I- and D-space tasks and that were used to build libraries, TKB allocates autoload vectors in the root of the task only for those autoloadable entry points in the library referenced by the task.

  - Internal Symbol Directory Records, along with their formats, are described in Appendix A of the RSX-11M/M-PLUS Task Builder Manual. They consist of:

    ● Type 1 records, generated by TKB and output to the .STB file

    ● Type 2 records, generated by language processors

    ● Type 3 records, created from type 2 records and output to the .STB file.

    ● Type 4 records, written to the .STB file without modification.

  - A new bit called LD$TYP distinguishes between a library or common. It is at the offset R$LFLG in the resident library name block data in Appendix B of the manual.

  - The first library in a cluster may be overlaid and contain a non-null root.

  - New Task Builder reserved symbols have been added to Appendix E.

- The Fast Task Builder supports the /EA switch and the TASK= option.

- Other, minor technical and editorial changes have been made.

1.2.7 RSX-11M/M-PLUS System Management Guide (U)

The following are changes or additions to the System Management Guide:

1.2.7.1 Virtual Console Routine (VMR) - RSX-11M now supports the following VMR SET keywords:

- /RNDC - Defines the length of the Executive round-robin scheduling interval.

- /RNDH - Defines the highest priority that will be considered for Executive round-robin scheduling.

- /RNDL - Defines the lowest priority that will be considered for Executive round-robin scheduling.

- /SWPC - Defines the length of a single Executive swapping interval.

- /SWPR - Defines a new priority range for Executive swapping.

- /LA50=ttnn: - Establishes ttnn: as an LA50 (Note that this support is not documented in the VMR chapter.)

The SAVE command supports the following new devices for saving system images: RX50, RA60, RC25, TSV05, and TU80.

The REDIRECT command has a new error message:

    TT redirect error

The message means that an attempt was made to redirect terminal (TTn:) to the null device.

1.2.8 RSX-11M/M-PLUS Error Logging Manual (R)

The following are changes or additions to Error Logging:

- Error Logging now allows hard and soft limits to be reached independently. Previously, reaching one of the limits would disable logging of either kind of error on that device. Now, reaching the soft limit will not affect the logging of hard errors and vice versa.

- Device timeouts are now logged as hard errors if unrecoverable and as soft errors if recoverable.

- When generating a report, RPT looks first for LX:[1,6]ERRLOG.ULB. LX: is a new pseudo-device used by Error Logging. If it fails to find that file, it looks for LB:[1,6]ERRLOG.ULB.

● There have been a number of minor changes in the Control File Language (CFL) component of the Error Logging System. The following lists the differences between CFL in RSX-11M V4.1 and the previous releases.

- %CNV$xxx functions:

  The field_width parameter is now optional and interacts with the optional fill_character parameter to determine whether the resulting string is simply printed as is or is left- or right-justified. In the earlier version, the digits in the string were always right-justified and blank filled if no fill_character was specified.

- %LOK$LENGTH function:

  This function always returns the length of the data in a packet or subpacket. The length word for the packet or subpacket is not considered part of the data and is not counted in determining the length value returned.

- %LOK$BYTE, %LOK$WORD, %LOK$LONGWORD functions:

  The offset parameter is the offset within the data of the packet/subpacket at which the byte, word, or longword begins. The offset unit is always in bytes, with the first byte of data in the packet/subpacket being offset 0.

- %STR$UPCASE function:

  %STR$UPCASE accepts an ASCII string as a parameter, and returns the ASCII string with all the lowercase ASCII characters converted to uppercase.

- DYNAMIC_TABLE statement:

  This new statement is a synonym for the FILE statement and should be used instead of FILE in any new code.

- FILE statement:

  The FILE statement will be removed in a future release. You should replace any FILE statement presently in your code with the DYNAMIC_TABLE statement.

- WRITE and WRITE_GROUP statements:

  Because of overlay restrictions, the following operators and functions cannot be used in expressions in WRITE or WRITE_GROUP statements:

        single and double operand numeric operators
        the MATCH operator
        %COD$xxx functions
        %CTL$xxx functions
        %PKT$xxx functions
        %RPT$xxx functions
        %STR$xxx functions
        %TIM$xxx functions
        %USR$xxx functions

- The RSX-11M/M-PLUS Error Logging Manual has two new chapters.

  - Chapter 4, Error Logging Control File Architecture, explains the control file modules in detail, including flow of program control, interface between modules, and module dispatching. A knowledgeable system programmer can use the information here to add user-written modules to the Error Logging System. The chapter includes extensively annotated examples of DIGITAL-supplied modules.

  - Chapter 5, Control File Language Guide, describes the Control File Language, which is used to write control-file modules.

1.2.9 RSX-11M Mini-Reference (R)

This revision to the RSX-11M Mini-Reference consists of a new section on RMS-11, as well as technical improvements to several other sections of the manual.

The RMS-11 section includes both procedures for invoking the following RMS-11 utilities, and summaries of their commands and keywords.

- RMSDES (RMS-11 File Design Utility)

- RMSIFL (RMS-11 Indexed File Load Utility)

- RMSCNV (RMS-11 File Conversion Utility)

- RMSDSP (RMS-11 File Display Utility)

- RMSBCK (RMS-11 File Back-Up Utility)

- RMSRST (RMS-11 File Restoration Utility)

In addition, the RMS-11 section lists the RMS-11 completion codes and fatal error codes.

The sections Executive Directives and I/O Error Codes have been updated to include new or corrected information.

1.3 NOTES ON SOFTWARE

The following sections describe the incompatibilities between RSX-11M V4.1 and previous versions, and provide miscellaneous information pertinent to the successful use of the software.

1.3.1 Baseline Systems

The V4.1 baseline system in [1,54], on the disk labeled RSXM35, requires a minimum of 64K words. This is an increase of 36K words over the minimum memory requirement for V4.0.

1.3.2 System Generation and Installation

The host machine for a system generation must meet the following minimum requirements:

- Must have at least 64K words of memory

- Must be mapped and have a null device (NL:)

1.3.2.1 Performing an RSX-11M V4.1 SYSGEN on a V3.2 System – If you intend to perform an RSX-11M V4.1 system generation while running under a V3.2 system, you must do the following:

1. Update several modules in the V3.2 indirect object library (IND.OLB).

2. Rebuild big indirect (BIGIND).

3. Use the TKB and VMR tasks from the V4.1 distribution kit when performing the SYSGEN.

There is a concatenated object file, INDFIX.OBJ, in UFDs [12,20] and [12,24] of the RSX-11M V4.1 distribution kit that simplifies the updating of IND.OLB. This object file consists of the individual updates to four indirect modules: INDPRC, INDPAR, INDOPN, and INDSET. The procedure for updating is as follows:

1. Set the UIC to [1,20] for an unmapped system or [1,24] for a mapped system.

2. Copy INDFIX.OBJ to the disk that has the most recent version of V4.0 IND.OLB.

3. Issue the following command: LBR IND/RP=INDFIX.

4. Assign the logical names TK (task image device) and MP (map device) to their proper devices: for example, ASN LB:=TK: and ASN NL:=MP.

5. Task-build the big indirect task as follows: TKB @BIGINDBLD.

6. Remove the current indirect (REM ...AT.).

7. Install the newly created big indirect (INS $BIGIND).

8. Mount the V4.1 baseline distribution kit (RSXM35), and install the TKB and VMR tasks from it. The tasks are in [1,54].

9. If your system does not have extend task support, and there is ample memory available, install the V4.1 TKB with an increment: for example, /INC=20000.

10. Use the V3.2 BIGMAC (or the V4.1 MAC) for the system generation. As with TKB, installing BIGMAC with an increment will speed up processing if your V3.2 system does not have extend task support.

Once you have completed this sequence of steps, you are ready to perform an RSX-11M V4.1 SYSGEN. Refer to the RSX-11M System Generation and Management Guide for instructions on performing an on-line SYSGEN.

1.3.2.2 Restriction to V4.1 On-Line SYSGEN - When performing a V4.1 on-line SYSGEN, SYSGEN will always install the Task Builder supplied on the distribution kit (...TKL).

This restriction is due to changes made in TKB to generate a new autoload vector format for overlaid tasks. These changes are closely coupled with the overlay run-time system found in SYSLIB and result in incompatibility problems between the old TKB and SYSLIB. Therefore, in order for SYSGEN to execute successfully, the tasks that SYSGEN needs must be built with the V4.1 TKB.

1.3.2.3 V4.0 Saved Answer Files - A V4.0 saved answer file (SYSSAVED.CMD) can be used when generating a V4.1 system. These files are now upward compatible. However, two new SYSGEN questions regarding checkpointing and SPM-11M support will be asked.

1.3.2.4 Determining the Number of RK05 Units - Fixed platter RK05 drives (RK05F) are handled as two units. The first of each RK05F pair must be an even unit number; for example, 0 and 1, 2 and 3. Be sure to take this into account when calculating the number of RK05 units.

1.3.2.5 SYSGEN on RL01s or RL02s - Due to the space limitations of the RL01 and the RL02, you should not perform more than one SYSGEN on the disk without first clearing some space on it. This clearing can be accomplished by copying files to another media and then deleting the unnecessary files.

1.3.3 User Account for the Warm-Up Session

The Introduction to RSX-11M and RSX-11M-PLUS includes a warm-up session for new users. A number of files that are associated with the warm-up session are included on the RSX-11M kits, along with a few other files intended for the convenience of all users.

During the post-SYSGEN clean-up procedure, while extracting the HELP files, you are asked if you wish to extract the INTRODUCTORY files. If you answer Yes to this question, the command procedure will extract a module from HELP.ULB called INTRO.ULB.

All the introductory and convenience files are contained in INTRO.ULB. Extract the module called INTROF, using the following procedure:

        MCR> LBR INTROFIL.CMD=INTRO.ULB/EX:INTROF

or

        DCL> LIBRARY/EXTRACT INTRO.ULB INTROF

This module is an indirect command file called INTROFIL.CMD, which you can run to extract all the other files by typing:

> @INTROFIL

The procedure extracts 20 files, including WHATSHERE.TXT, which explains what all the other files are. You can use these files in two ways, depending on the number of new users you have coming on to your system.

In UFD [200,1] on your target system disk are introductory files used with the warm-up session for new users presented in the Introduction to RSX-11M and RSX-11M-PLUS and RSX-11M-PLUS manual. When new users come onto the system, the system manager should have them use the account with the UIC [200,1] and the account name USER which has been supplied with the distribution kit. New users can log in to this account and follow through the entire warm-up session. See Section 6.2.9 of the RSX-11M System Generation and Installation Guide and pages 1-5 to 1-7 in the Introduction to RSX-11M and RSX-11M-PLUS.

Alternatively, the system manager can copy the introductory files to the new user's own account, then instruct the new user to delete these introductory files when he or she has finished with the warm-up session.

Use the convenience files as you wish. They are marked with a * in the list below.

*A.A;1	*SUCCESS.TSK;1
*WARNING.TSK;1	*ERROR.TSK;1
*SEVERE.TSK;1	LOGIN.CMD;1
DELETE.CMD;1	*CLOCK.MAC;1
SHOW.CMD;1	*CLKGEN.CMD;1
STARS.MAC;1	FLY.TXT;1
TEXT.TXT;1	FLY.TXT;2
CLEAN.CMD;1	*WHATSHERE.TXT;1
*TMCLI.MAC;1	SHAVE.CMD;1
*TMCLI.FTN;1	HIYA.MAC;1

For more information on extracting HELP files from RSX-11M kits, see Chapter 6 of the RSX-11M System Generation and Installation Guide.

1.3.4 BRU64K Restriction

On an BRU64K stand-alone system, you should not run BRU and BAD simultaneously. Because BRU and BAD use common buffer space, running both tasks at the same time yields unpredictable results.

1.3.5 DSC64K, DSCSYS, and BADSYS

The DSC64K, DSCSYS, and BADSYS stand-alone systems have been removed from the distribution kits. These stand-alone systems are no longer needed, as BRU64K is now used as the stand-alone system.

1.3.6 Power-Fail Recovery

A disk driver is called at its power-fail recovery entry point when a power-fail occurs. This recovery routine gives a disk sufficient time to spin back up and be ready to accept I/O requests.

There are three possible recovery procedures:

1. If a device is busy prior to the power-fail and the requested I/O has not completed, the routine times out and checks the device status until the device is ready for I/O operations to resume.

2. If a device is not busy prior to the power-fail and receives an I/O request after the power-fail, the routine times out and checks the device status until the device is ready for I/O operations to resume.

3. If a device is not busy prior to the power-fail and there are no pending I/O requests, operations resume with no special handling by the recovery routine.

Note in the first two cases that, if the maximum time-out count is reached before the disk spins back up and is ready, an unsuccessful I/O completion code is returned.

1.3.7 Change to Executive's Task Abort Code

The Executive's task abort code has been modified. If you install an Ancillary Control Processor (ACP) task with post-mortem dump (PMD) enabled and this task aborts, then the system will crash. A crash dump is produced, which shows the state of the I/O data structures at the time the ACP task aborted. By default, most DIGITAL-supplied ACPs are installed with PMD enable.

1.3.8 Restriction to Error Logging

By moving the Error Logging routines into the Executive Common, the following restriction is introduced: If a driver of an error logging device calls the $CRPKT routine to create an error logging packet, the data address for the data subpacket must not be an address within the driver. Specifically, the address must not be mapped by APR 5, as that APR is used to map the common. Any user-written driver that performs such a function must allocate a piece of pool, fill in the appropriate information, and pass the pool address to the create packet routine.

1.3.9 TDX (Catch-All Task)

Support for TDX, the catch-all task, is provided with RSX-11M systems. The primary purpose of the catch-all facility is to give you the means to run uninstalled tasks.

Any task installed with the task name ...CA. is treated as a catch-all task. If the Monitor Console Routine (MCR) receives an unrecognized command, it searches for a task with that name and passes the command line to this task. To use TDX as the catch-all task for your system, install it as follows:

```
>INS $TDX/TASK=...CA.
```

TDX checks the typed command against its list of commands. If the commands match, TDX translates the command into a legal MCR command. These commands and their MCR translations are:

Command	Translation	Meaning
ATS	ACT /ALL	Display the names of all active tasks in the system.
ATS ttnn:	ACT /TERM=ttnn:	Display the names of all active tasks on the specified terminal.
CHD	SET /UIC	Display the current default UIC for terminal TI:.
CHD g m	SET /UIC=[g,m]	Change the default UFD to the UFD specified.
CLR		Clear the issuing terminal's screen and set cursor to 0,0. Returns exit status of EX$SUC if terminal is a CRT, and EX$WAR if it is not.
CRE file	PIP file=TI:	Create a new file without invoking an editor.
CVT val		Evaluate an arithmetic expression, convert that expression into different formats, and display all the formats on your terminal. CVT accepts input in octal (nnn or nn,nn) or decimal (nnn. or nn.,nn.) words or bytes, hexadecimal numbers ($nnnn), radix-50 (%ccc) or ASCII ('c or "cc) characters, or arithmetic expressions using +,-,/,* and < >.
DEL file(s)	PIP file(s)/DE	Delete the specified file(s).
DIR [file(s)]	PIP [file(s)]/LI	Display a directory listing at the terminal.
DLG	DEV /LOG	Display information about all the logged-in terminals on the system.

Command	Translation	Meaning
DLN	NCP SHOW KNOWN NODES	Display all known DECnet nodes.
FRE	PIP /FR	Display the amount of available space on SY:, the largest contiguous space on SY:, the number of available file headers, and the number of file headers used.
FRE ddu:	PIP ddu:/FR	Display the same information as FRE for a specified device.
PUR file(s)	PIP file(s)/PU	Delete all but the latest version of a file.
SHQ	SHOW QUEUE ALL	Display information about all entries in all print queues.
SYS	SET /SYSUIC	Display the current system UIC.
TDX		Display the current version of TDX.
TYP file(s)	PIP TI:=file(s)	Print files on your terminal.

You may want to add other commands to TDX by modifying the source file in [24,10]TDX.MAC. The routines for the commands are at the end of the source file and serve as examples for user-tailored routines. To reassemble the source file after making additions, use the TDXASM.CMD file located in [24,20]. To rebuild TDX, follow the procedure in the Phase III section of the RSX-11M System Generation and Installation Guide. TDX comes preassembled ([1,24]TDX.OBJ) if you do not wish to make any changes.

Location of files on multivolume kits

Distribution	TDX.MAC	TDX.OBJ, TDXBLD.BLD
RL01/02	HLPDCL	EXCPRV
RK06/07	CLISRC	RSXM35
(all other)	RSXM35	RSXM35

If the user's typed command does not match any of TDX's commands, TDX attempts to issue one of the following two MCR command options. In the examples, XXX represents the first three characters of your command.

The option TDX uses depends on whether the logical devices ZZ1: or ZZ2: exist on your system. You can assign one of these devices in your LOGIN.CMD file. Note that the presence of the ZZ1: assignment masks the presence of the ZZ2: assignment. If no assignment is made, TDX will not exercise either option.

1.

      MCR> RUN $XXX/TASK=XXXTTN/CMD="params..."

This option installs, runs, and then removes a task. "Params..." represents the command line that you enter. Your command line cannot exceed 39 characters, because TDX adds other characters to this line. If you want this option, place the following in your LOGIN.CMD file:

      ASN SY:=ZZ1:

2.

      MCR> @SY:[loginuic]XXX.CMD or
      MCR> @LB:[libuic]XXX.CMD or
      MCR> @SY:[loginuic]CATCHALL.CMD or
      MCR> @LB:[libuic]CATCHALL.CMD

For this option, TDX searches for one of the @[...]XXX.CMD indirect command files, in the order indicated. As soon as TDX locates one of these files, the search stops. You can design the indirect command file to perform a variety of functions, such as installing tasks, providing HELP, or issuing error messages.

If you want this option, place the following in your LOGIN.CMD file:

      ASN SY:=ZZ2:

RSX-11M V4.1 contains a sample indirect command file, [24,24]CATCHALL.CMD.

The ZZn option may be controlled for all users by placing the following command in the system startup file:

      ASN SY:=ZZn:/GBL

where n equals 1 or 2, depending upon which ZZn option you prefer.

TDX issues the following message if the user's command does not match a TDX command or if TDX cannot locate the necessary task.

      MCR -- Task not in system

You may wish to install some tasks permanently, rather than invoking TDX to install them on an as-needed basis. It is advisable to permanently install:

● Tasks requiring a larger increment than the default (see the INSTALL command in the RSX-11M/M-PLUS MCR Operations Manual)

● Frequently used tasks

There are two cases, when noninstalled tasks are invoked by indirect command files, where TDX may affect the execution of the command file:

● If the command line's length exceeds 39 characters

● In systems without Parent Offspring tasking, TDX has no means of returning the exit status of called tasks to Indirect. In this case, Indirect proceeds immediately to the next command line without waiting for the called task to finish. This has the same effect as preceding the command line with .XQT.

1.3.10 Restriction for DCL as the Catch-All Task

If your command line interpreter is set to MCR and DCL is installed as the catch-all task (...CA.), do not use the DCL INITIALIZE/UPDATE command to modify your disk parameters. When you enter this command, MCR interprets the INI portion of the command to mean INITIALIZE VOLUME, and consequently destroys all existing files on your disk. As alternatives, either use the MCR HOME command or set your command line interpreter to DCL before using the INITIALIZE/UPDATE command.

1.3.11 Indirect Command Procedure Library

Users wishing to use the Indirect command procedure library INDSYS.CLB (as documented in the RSX-11M/M-PLUS MCR Operations Manual) must copy it from the MCR source volume directory to their system volume.

On single disk distribution kits (RP04/5/6 or RM03, for example), copy the file as follows:

    PIP LB:[1,2]=LB:[12,10]INDSYS.CLB

On multidisk distribution kits (RL01/02, for example), mount the MCR source volume and copy the file as follows:

    PIP LB:[1,2]=dd:[12,10]INDSYS.CLB

The characters dd represent the drive on which the MCR source volume has been mounted.

For the RK06/07 kit, the file is on CLISRC; and for the RL01/02 kit, the file is on MCRSRC.

1.3.12 Indirect Command Processor Support for V4.1

Following are descriptions of the three versions of the Indirect Command Processor that are supported in RSX-11M V4.1.

- The first version is called ICP and contains the same functional parameters as the IND task released in V4.0. This version requires a 12K partition size and must be used when you are doing a V4.1 SYSGEN. ICP is the standard Indirect created during the Phase II privileged task build. There is no FCSRES version of this task.

- The second version is called ICQ and is more heavily overlaid in order to fit in an 8K partition. This version supports the minimum of functional parameters necessary to execute a command file that does not exceed the following:

  - 1 concurrently open data file

  - 1 parallel XQT directive

  - 2 Direct Access Labels

  - 1 GOSUB nesting level

  - 2-level depth of open command files

  - 108(decimal)-byte command file input buffer

ICQ is designed for systems that lack sufficient memory space to support a 12K task. ICQ's execution time is slower than ICP's, and its symbol table space and functional parameters are insufficient to complete a SYSGEN. ICQ is built during SYSGEN only if the target system contains less than 24K of memory.

- The third type of task is the resident FCS (FCSRES) version of ICQ. This task is also highly overlaid and is slower than ICP; however, this version has the same functional parameters as ICP with approximately the same symbol table space as ICQ. The task resides in an 8K partition and has the following parameters:

  - 4 concurrently open data files

  - 16 parallel XQT directives

  - 16 Direct Access Labels

  - 8 GOSUB nesting levels

  - 4-level depth of open command files

  - 132(decimal)-byte command file input buffer

This version is designed for systems that lack sufficient memory space to support a 12K task but still want to retain the full capabilities of the ICP task. However, it does not contain enough symbol table space to complete a full SYSGEN procedure, and it must be built as a privileged task with FCSRES support using the SYSGEN3 procedure.

1.3.13 Task Builder Restrictions

The following two restrictions apply to the V4.1 Task Builder.

- You may experience problems if you use the V4.1 Task Builder (task version M40.10) with a SYSLIB other than the one supplied with V4.1. Changes and additions have been made to the overlay run-time system modules that make them incompatible with previous versions of SYSLIB.OLB.

- The first library in a cluster of libraries may or may not be overlaid and may or may not have a null root. The remaining libraries in the cluster must be overlaid with a null root. The first library in a cluster is the one named first in the Task Builder cluster (CLSTR) option.

  Severe performance degradation can occur with cluster libraries if you have one of the following two conditions in your task:

  1. The first library is overlaid and has a null root.

  2. The first library called by the task is not the first library specified in the CLSTR option.

If the first library in the CLSTR option is overlaid and has
a null root, the overlay run-time system cannot distinguish
the first library from the remaining libraries in the
cluster. The Task Builder therefore assumes that the first
library called by the task is the first library; this first
library is the one that Task Builder consistently maps by
default. Therefore, although the preceding two conditions
do not cause errors, they may cause performance degradation
because of excessive mapping and unmapping of libraries in
the cluster.

To avoid this problem, the library first called by the task
should be the first library specified in the CLSTR option.

1.3.14 Modifications to System Library Task Builder Data Structures

Beginning with the V4.0 release, the Task Builder and the associated
overlay run-time system incorporate new overlay functionality and
modified data structures.

New modules are included in the system library to support cluster
libraries (see the RSX-11M/M-PLUS Task Builder Manual for a complete
description of this feature). In order to minimize the virtual
memory impact of the new features, some changes were made in the
older modules. The customer product that used the run-time routines
may not build correctly or may malfunction unless minor changes are
made. In particular, the following points are relevant:

- The run-time routines reference the register save routine,
  .SAVR1 instead of .SAVRG. In some overlaid applications,
  this routine may have been explicitly loaded in an overlay
  (for example, as part of a tightly overlaid FCS-11). This
  reference results in a "Multiply defined global" message
  from TKB. The change is to remove explicit .SAVR1
  references from overlay specifications in the .ODL file and
  to let TKB place this module in the root with the overlay
  run-time routines.

- The linkages between the various overlay run-time modules
  have been modified slightly to support the new library
  mapping; therefore, users who have developed debugging
  tools that "hook" themselves into the overlay system should
  refer to a copy of the source microfiche or the source kits.

1.3.15 Transportability among 22-bit Systems

RSX-11M supports transportability among all 22-bit systems. These
systems can differ as to whether the Executive handles NPR transfers
by performing memory management functions, or the hardware has its
own 22-bit physical memory addressing capabilities.

To make these systems transportable, you must boot your target
system on the processor selected at SYSGEN and then SAVe it. The
saved system will then be hardware bootable and transportable among
the different 22-bit systems.

1.3.16 Device Drivers and User Tasks

The addition of new and enhanced device support to RSX-11M V4.1 may cause problems for some user tasks in V4.1. The characteristics of the device drivers and the procedures recommended for handling such problems are described as follows:

- The UDA50 (DUDRV) dynamically updates the system data base to reflect the characteristics of the disk.

  Recommended action: User tasks should issue a QIO Attach function before requesting the device's characteristics with the GET LUN directive.

- The TM11A/B device driver (MTDRV) dynamically updates the system data base to reflect the density characteristics of TE10/TU10.

  Recommended action: User tasks should issue a QIO function using the function codes for the powerfail recovery procedure, as described in the RSX-11M/M-PLUS I/O Drivers Reference Manual, before requesting the device's density characteristics with the GET LUN directive.

- The RX02 device driver (DYDRV) dynamically updates the system data base to reflect the characteristics of the media in the RX02 drive.

  Recommended action: User tasks should issue a QIO Sense Characteristics function before requesting the device's media characteristics with the GET LUN directive.

- Users issuing QIOs directly to MSDRV must be aware of the following:

  1. The completion of the IO.RWD request occurs when the MS: device reaches BOT.

  2. When the MS: device changes status from either off-line to on-line or from on-line to off-line, the MS: device inhibits further physical I/O operations. After such a change, the user must issue either an IO.RWD or an IO.SMO request that succeeds before I/O operations resume.

  3. MS: read/write data transfer features are as follows:

     - The data buffer starting address must be on a word boundary.

     - The data transfer size can be an odd or even byte count.

     - The user can swap the odd and even data bytes by using the tape characteristic bit 1 of IO.SMO or IO.STC requests. When bit 1 is set to 0, no byte swap occurs. When bit 1 is set to 1, the byte swap does occur. If you use byte swapping, it is recommended that the data buffer size be an even byte count.

1.3.17 Changes to Code and Data Structures for Privileged Tasks

The changes documented in this section were new for RSX-11M V4.0 rather than for V4.1. This information is included here because it has not yet been incorporated into the documentation set.

Many of the changes made to RSX-11M Version 4.0 may affect the upgrading of privileged tasks and drivers that were written for a previous release.

The following sections describe the most significant changes to code and data structures since the previous release.

Note that nonprivileged tasks should not be affected by any of these changes.

1.3.17.1 Changes Made to Executive Code -

- The module IOSUB.MAC has been divided into four modules: IOSUB, EXESB, MEMAP, and MDSUB. The last three are new.

- The Get Partition Parameters directive now returns the real virtual starting address of a privileged task's task region instead of 0 as it did in Version 3.2.

- Many Executive routines have been moved into the Executive commons (if Executive common support was selected). Therefore, it may not be possible for privileged code to reference these routines as they may have done in a previous release. To determine if an Executive entry point is in an Executive common, consult RSX11M.MAP. The entry point is in a common if its address falls between 120000 and 140000.

  If the entry points are inaccessible to the task because they reside in the Executive commons, a procedure for accessing that routine is available. A transfer routine must be written by the user that will permanently reside in the mapped Executive address space. See module DRSUB for examples of transfer routines. Note that transfer routines are already provided for the most commonly called subroutines now moved into the commons and their existence is transparent to any task referencing them.

- The RSXMC.MAC symbol P$$D70 which indicated cache memory on a 22-bit processor has been changed to C$$CHE.

1.3.17.2 Changes Made to Executive Data Structures -

- The Task Control Block (TCB) has undergone many changes. The present format can be found in an appendix of the RSX-11M/M-PLUS Crash Dump Analyzer Reference Manual. The following offsets are new:

  - T.TIO   Buffered I/O count (byte)

  - T.TKSZ   Initial task size (root)

  - T.EFLM   Event flag waitfor mask and mask address

  - T.HDLN   Task header length in 32 word blocks (byte)

  - T.GGF   Task group global event flag use count (byte)

- To enhance asynchronous buffered I/O, the bit T2.TIO that allowed for only one buffered I/O per task, has been changed to the byte T.TIO. This byte allows a task to queue many buffered I/O requests simultaneously.

- The word at offset O.STAT+2 of the offspring control block is now used to contain a secondary exit status, the TKTN abort code, which is filled in whenever the offspring exits. Previously, only the word at O.STAT was used. Note that more words in the exit status region of the OCB may be used in the future and therefore should be considered as reserved for future expansion.

- The addition of alternate CLI support has changed how CLIs are associated with individual users. On systems that include alternate CLI support, the word previously at offset U.CLI in terminal UCBs, (DV.TTY is set) has been redefined. On these systems it is referred to as U.MUP, since it now contains a bit mask for various multiuser features. Bits 1 - 4 in this word are now used to identify the CLI associated with that terminal. The bits form an offset into a table located at $CPTBL in the Executive module SYSCM. This module contains a pointer to the CPB (CLI Parser Block) for the CLI. Since only four bits are used to form this offset, the system is limited to 16 CLIs. The bit that is used for the NOBROADCAST feature is also in this word, which is why this feature is only available on systems supporting alternate CLIs.

- There are now two additional feature mask words. These words are documented in an appendix of the RSX-11M/M-PLUS Crash Dump Analyzer Reference Manual. These words are accessed as positive offsets from the original feature mask word, $FMASK.

1.3.17.3 Changes Made to I/O Data Structures -

- The bit DV.MXD in offset U.CW1 of the UCB (previously for mixed MASSBUS) has been changed to DV.MSD. This bit is set for all mass storage devices, and must be turned on for all user-written drivers that want to support error logging. In addition, error logging devices must have four words preceding each UCB (negative offsets).

- The bit DV.SWL in offset U.CW1 of the UCB (previously unused) is now used to maintain the software write-lock status of the device.

- Prior to V4.0, two words were added to the end of each DH11 UCB to contain the vector and CSR of an associated DM11BB. In V4.0, these two words were eliminated. The information is now contained in fork block offsets in the SCB.

1.3.18 MACRO-11

The maximum number of relocatable terms in a complex expression has been changed from the old maximum of 6 to 20. This change was made by increasing the maximum size of the .OBJ record that MACRO-11 produces from 42(decimal) bytes to 128(decimal) bytes.

You cannot verify that code generation is correct by comparing .OBJ files created by different versions of MACRO-11. The correct way to verify code generation is to task-build the .OBJ files, then compare the .TSK image files.

Because the .OBJ files produced by the new version of MACRO-11 differ from those of previous versions, you must recompute checksums using the PAT utility for any object patches assembled with this version of MACRO-11.

See the Release Notes at the end of the PDP-11 MACRO-11 Language Reference Manual for documentation of the changes to MACRO-11.

1.3.19 Restriction for EDT Version 2.0

These restrictions apply to V2.0 of EDT, which is a layered product bundled on your RSX-11M V4.1 distribution kit.

Before EDT can be run in keypad mode, your terminal must be set to lowercase (SET /LOWER). If the terminal is not set lowercase, the terminal driver converts the escape sequences sent by the keypad keys into uppercase and EDT does not recognize them.

Also, if you type CTRL/C while EDT is writing the output file (that is, after you type the EXIT command but before the CLI prompt appears), EDT stops writing the output file. No message is displayed to indicate that your output file has been only partially written.

1.3.20 Changes to the Command String Interpreter (CSI)

The Command String Interpreter (CSI) was rewritten for Version 4.0 of RSX-11M to incorporate new functionality. The IAS/RSX-11 I/O Operations Reference Manual documents a list of changes to CSI.

Two additional changes are:

1. .CSI1 no longer references the module ODCVT.

2. .CSI2 no longer references the module ODCVT and now references OD2CT.

These changes are normally not visible to the user, but could be a problem for some overlaid tasks.

1.3.21 Bootstrapping a DB-, DM-, or DR-Based System

The MCR BOOT command can loop indefinitely when it boots a system from an RP04/05/06 or an RK06/07 or an RM03/05/80 or an RP07 if it encounters an ECC correctable error when reading the system image. The bootstrap driver is not large enough to include ECC logic. Therefore, it retries the read operation, and never succeeds.

To recover from this error, use PIP to copy the system image to another area on the disk and retry the boot operation. Alternatively, you can copy the entire image to another disk and boot from that one.

1.3.22 Correction to Multiuser MCR on an Unmapped System

The multiuser version of MCR will not task-build properly for an unmapped system. Before doing a SYSGEN, locate the file [1,20]MCMBLD.BLD. Edit the file and change the line

    .DATA PAR=GEN:'$GPB':20000

to

    .DATA PAR="$GEN'

For the RL01 distribution kit, the file can be found on the EXCPRV disk.

1.3.23 File Transfer Program (FLX)

The behavior of FLX in handling an error during a copy operation to an RT-11 device has changed. Previously, FLX allowed a partial file to be created on the output device when insufficient space existed on the output disk with the same name as the newly created partial file. FLX deleted the preexisting file and substituted the partial file. This replacing of an existing file with a partial file resulted in a loss of information.

Now, if an error occurs during the copy operation, the partial file is not substituted for an existing file on the output disk. The partial file no longer is closed as a permanent entry, and the definition and contents of the existing file remain intact.

1.3.24 Incompatibilty Problem with BRU Tapes

Because of BRU's new sort algorithm, V4.1 can read V4.0 BRU backup tapes, but V4.0 BRU cannot read V4.1 BRU backup tapes.

1.3.25 Device Restrictions

The following device restrictions apply to RSX-11M V4.1:

- MCR/DCL

  The MCR/DCL SET and SHOW commands do not recognize the LPQ02 letter-quality printer.

  The DCL SET and SHOW commands do not support the LA50 personal printer.

- File Transfer Program (FLX)

  The following devices do not support DOS-11 or RT-11 format:

      RC25
      RA60
      RA81
      RD51
      RX50

- Disk Volume Formatter (FMT)

  The following devices are not supported:

          RD51
          RX50
          RA60
          RA81
          RC25
          RP07

- Bad Block Replacement Control Task (RCT)

  The RX50 does not have an RCT area.

- Virtual Monitor Routine (VMR)

    - The SAVE command does not recognize the following
      devices:

            RA81
            RD51

    - The SET command does not recognize the following devices:

            LN01
            LP26
            LP27
            LP07
            LQP02

- Error Logging

  The Error Logging control files do not have device-specific
  information for the following devices:

          RD51
          RX50

- Crash Dump Analyzer (CDA)

  The following fixed disks cannot serve as CDA devices:

          RA80
          RA81
          RD51
          RC25 (nonremovable)

1.3.26 Queue Manager

The MCR /NM switch suppresses messages from the Queue Manager (QMG)
such as "Queue already exists."

In DCL, the equivalent switch is /NOWARNINGS.

Once a queue has been established, it will remain until eliminated,
in spite of system crashes.   But it is desirable to attempt to
reestablish the standard print queues each time the system is
bootstrapped, in case a queue has been inadvertently eliminated.
Issuing a QUE LP0:/CR command, though, yields the "Queue already
exists" message from the QMG if the LP0 queue has not been
eliminated, as is generally the case.

The /NM switch, therefore, is used in the prototype STARTUP.CMD supplied on the distribution kit to reduce the "clutter" that is printed on the console terminal when STARTUP.CMD is invoked.

1.3.27 Building the COBOL-81 Compiler

If you are installing COBOL-81, you must answer the following dialogue prompts as indicated in order to build the COBOL-81 compiler correctly:

    Do you want to build the default compiler <YES>?

Do not take the default. You should type: NO

The next two questions in the dialogue concern CIS and RMS.

The first of these two questions is:

    What should be the compiler task size in K words (legal range
    from "26" to "31") <31>?

The legal range for the compiler task size actually is 25 to 30 K words. Do not use the default for this question; type the numerical value of the compiler size from the revised legal range.

Note that you should not respond with 25 to this question either.

The next question asks whether you want to change the default of any of the compiler switches. You can either take the defaults or change them.

The entire COBOL-81 dialogue is documented in the COBOL-81 Installation Guide.

1.4 REPORTING PROBLEMS AND APPLYING CORRECTIONS

RSX-11M/RSX-11S includes several features to use if you encounter questions or problems with your system or if DIGITAL provides temporary patches to improve its software performance.

Software Performance Reports (SPRs) allow you to report any problems directly to DIGITAL. The procedure for submitting an SPR is described in Section 1.4.1.

Sections 1.4.2 and 1.4.3 describe how to apply temporary software corrections to your system. You should use instructions to apply manual patches that are given either in a DIGITAL response to an SPR or in a Software Dispatch article.

Note that any patches in the Software Dispatch are temporary and will be overriden by future DIGITAL Updates. An Update is a periodic software fix to your system, distributed in the form of machine-readable code. DIGITAL customers who are under warranty or who have purchased any level of software product service will receive the Software Dispatch and Updates.

1.4.1 Software Performance Report

When you complete an SPR form, describe only one problem per form,
to simplify record keeping and facilitate a more rapid response.

An SPR can be used for the following:

- Software errors

- Documentation errors when the documentation comment form is
  not appropriate

- Inquiries

- Suggestions

- Follow-up on a previous SPR

An SPR cannot be used for the following:

- Software license and price policies

- Obvious hardware problems

- Logistical or clerical problems with kits or with Software
  Dispatch, such as blank media, or failure to receive the
  Software Dispatch

- Problems with user-written software

There are three categories of reports that you can send:

- Problem/error

  This type of report should have a priority of one to five.
  (Answer is generated.)

- Suggested enhancement

  This type of report contains suggestions or queries that are
  brought to the development group's attention and has a
  priority of five. (No answer is generated.)

- Other

  This type of report contains queries and suggestions that
  are priority five. (Answer may be generated.)

The priority numbers are as follows:

1. Most production work cannot be run.

    a. Major system functions are unusable.

    b. System does not boot.

    c. Necessary peripherals cannot be used.

2. Some production work cannot be run.

    a. Certain functions are unusable.

    b. System performance has declined.

    c. Installation has insufficient excess capacity.

3. All production work can be run with some user impact.

    a. Significant manual intervention is required.

    b. System performance has declined but installation has excess capacity.

4. All production work can be run with no significant impact on user.

    a. Problem can be patched.

5. No system modifications are needed to return to normal production.

    a. This category includes suggestions, consultations, or errors in documentation.

Because problems are often difficult to reproduce with a different system configuration, please define as precisely as possible the state of the system when the problem occurred. Illustrate a general problem with specific examples. When you refer to a manual, specify the full title and section, table, or page number.

Please supply the following information (in machine-readable form where applicable) when you report a problem:

1. Crash - 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, please accurately describe the details, supplying hard copy or user source code when necessary.

2. Drivers - Controller/device information, software options, error log output, copy of device registers, and a sample program.

3. Utilities - A copy of your terminal output, showing setup commands, before and after effects, and relevant file information.

4. Task builder - A copy of your terminal output command files, the task map, and a dump of the first few blocks of the task image.

5. File system - Corrupted Volume: Output from Verify utility and dump of the volume. - Improper results: Error code, file header dump, sample program.

6. Error Log Report generator (RPT) - A copy of the report file generated by RPT, either on a hard copy listing file or on machine-readable media.

If a failure occurs when you are running privileged, add-on software (for example, DECnet) try to reproduce the failure without the additional software. Indicate on the SPR the behavior of the system with and without the add-on software.

The SPR process takes time; therefore, if you have a critical problem, contact the local DIGITAL office. In the meantime, read the Software Dispatch for news on RSX-11M, which includes changes and problems other people have found and solved.

1.4.2 Applying Corrections to Source Files

Interim changes to the Executive, MCR Error Log Control Files, and device drivers are made by creating correction files, which are processed by the Source Language Input Program (SLP). SLP generates a new copy of the modules that contained errors by applying corrections to the source file on the distribution kit. (See the RSX-11M/M-PLUS Utilities Manual for complete information on SLP.)

NOTE

A new update policy is in effect. All software fixes will be distributed as machine-readable updates instead of patches to be manually entered. Also, any patches published in the Software Dispatch are temporary fixes that will be superseded by the next update.

After you have applied the corrections and 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 contents of the distributed sources.

Because RSX-11M is distributed on a wide variety of media, it is not practical to provide comprehensive examples for applying source corrections. In the sample procedures that follow, several assumptions are made: You must be a privileged user executing on a multiuser, mapped, dual RL01 system with an RL01 distribution kit. If your system disk is large, it already contains all the necessary files and file transfers are not required.

1.4.2.1 Updating an Executive Source Module

To update an Executive source file (ABCDEF.MAC for example), mount the disk labelled EXCPRV and create a SLP correction file named ABCDEF.COR in the UFD [11,40]. Then running under UIC [11,10] submit the correction file to SLP. For example, you could follow this sequence to update REQSB.MAC:

```
>SET /UIC=[11,40]
>ALL DL1:
>MOU DL1:EXCPRV
>ASN DL1:=SY:
>EDI REQSB.COR

[CREATING NEW FILE]
INPUT
  .
  .
  .
*EX
[EXIT]
>SET /UIC=[11,10]
>SLP @[11,40]REQSB.COR
```

If the updated Executive module in your system is not a loadable driver, and the module is not resident in one of the Executive commons, (EXCOM1 and EXCOM2), rebuild the Executive and privileged tasks to incorporate the modification.

To rebuild, do the following:

1. Assemble the new module, using the RSXMC.MAC file for the target system.

2. Use LBR to replace the old version of the module in RSX11M.OLB on the target system.

3. Invoke SYSGEN Phase II.

If the updated Executive module in your system is resident in one of the Executive commons, EXCOM1 or EXCOM2, the process of incorporating the modification is as follows:

1. Execute the first two steps in the procedure listed above. There is no need to rebuild the loadable drivers or privileged tasks in this case.

2. Rebuild the Executive as follows:

   >SET /UIC=[1,24]

   >TKB @RSXBLD

3. Create a new system image file as follows:

   SET /UIC=[1,54]

   PIP RSX11M.SYS/NV/CO/BL:XXX.=RSX11M.TSK

   Note that XXX should be equal to whatever value was used in the original SYSGEN. This value is usually equal to 4 times your system memory size in 1024 decimal-word blocks, plus 2, up to a maximum of 498.

4. VMR @SYSVMR.CMD

If the modified file in your system is a loadable device driver (ZZDRV.MAC for example), the updated module can be replaced without doing another system generation. Assemble the updated module and replace the resulting object file in the RSX11M.OLB of your target system.

Boot the target disk:

```
>ALL DL1:
>MOU DL1:EXCPRV
>SET /UIC=[1,24] ! [1,20] for unmapped systems
>MAC ZZDRV=[1,1]EXEMC/ML,[11,10] RSXMC/PA:1, DL1:ZZDRV
>DMO DL1:
>DEA DL1:
>LBR RSX11M/RP=ZZDRV
>PIP ZZDRV.OBJ;*/DE
```

Use the indirect command file [200,200]SGNBLDDRV.CMD to produce the files ZZDRV.TSK and ZZDRV.STB. If necessary, copy ZZDRV.TSK and ZZDRV.STB into the UFD corresponding to the system UIC.

```
>@[200,200]SGNBLDDRV
  .
  .
  .
>SET /SYSUIC
 SYSUIC=[g,m]
>SET /UIC=[g,m]
>PIP /NV=[1,54]ZZDRV.* ! or [1,50] for unmapped systems
```

Use 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 did the old one. It may be necessary to unload and reload all of the loadable drivers in that partition to create enough room. If the system is unmapped, some other drivers may have to be rebuilt.

```
>VMR
ENTER FILENAME:RSX11M
VMR>UNL ZZ:
VMR>LOA ZZ:
VMR> ^Z
>RUN $SHUTUP
```

Hardware bootstrap the modified system.

1.4.2.2 **Updating an MCR Source Module** - Updating an MCR source file is similar to updating an Executive source file.

NOTE

> For illustration purposes, the following discussion applies to the RL01/RL02 kit. The illustration is adaptable to other types of distribution kits. Please refer to the table at the end of this section for disk labels of other types of distribution kits.

The MCR source files are on a disk labeled MCRSRC in UFD [12,10]. The source files for the following modules (referred to as terminal driver-dependent modules) appear on two disks, MCRSRC and EXCPRV:

```
DV2OV
SPSOV
SX1OV
SX2OV
SX3OV
SX4OV
STSUB
```

The steps to update a source file for MCR (single-user and multiuser) and external MCR tasks are the same. The procedures for rebuilding the single-user and multiuser MCR differ slightly. Furthermore, if the module is one of the terminal driver-dependent modules, the corrections should be applied to source files on both disks. An example is provided below for each procedure.

Convention: The notation [1,2x] represents [1,24] for mapped system and [1,20] for unmapped system.

The following example explains the procedure to update MCR for a RL01/RL02 kit. To update the MCR source file SPSOV.MAC, mount the RL01/RL02 disk labeled MCRSRC.

Create the SLP correction file SPSOV.COR and use it to update SPSOV.MAC. Assemble SPSOV for the target system. Since SPSOV is a terminal driver-dependent module, the correction must be applied to source files on MCRSRC and EXCPRV.

```
!
! DL0: is mounted with RSXM35 baseline disk
!
>SET /UIC=[12,40]
>ALL DL1:
>MOU DL1:MCRSRC ! mount disk with source file
>ASN DL1:=SY:
>EDI SPSOV.COR ! creating the SLP correction file
[CREATING NEW FILE]
 .
 .
*EX
[
>SET /UIC=[12,10]
>SLP @[12,40]SPSOV.COR ! apply correction on MCRSRC
!
! assemble the updated module
!
>SET /UIC=[1,2x]
>ASN DL0:=SY:  ! assign SY: to baseline disk
>ASN DL0:=LB:  ! assign LB:
>MAC SPSOV=[11,10]RSXMC/PA:1,LB:[1,1]EXEMC/ML,DL1:[12,10]SPSOV
>
!
! if module is terminal-driver-dependent,
! repeat above procedure with the EXCPRV disk
!
>SET /UIC=[12,40]
>MOU DL1:EXCPRV ! mount second disk with terminal-driver
    ! dependent source file
>ASN DL1:=SY:
>EDI SPSOV.COR ! create same SLP correction file as above
[CREATING NEW FILE]
 .
 .
*EX
[EXIT]
>SET /UIC=[12,10]
>SLP @[12,40]SPSOV.COR ! apply correction on EXCSRC
>
```

Single-user and multiuser versions of MCR require different procedures for incorporating the modified module into the system. The procedures involve the following steps:

1. Incorporate the updated module into MCR's object library. If the module is terminal driver-dependent, append it to the file of concatenated MCR terminal driver-dependent modules.

2. Invoke SYSGEN3 to rebuild MCR and install it into the system with VMR, ensuring that SY: and LB: are first assigned to the disk containing the target system.

To rebuild single-user MCR, use the following procedure:

```
!
! DL0: is mounted with RSXM35 baseline disk
!
>MOU DL1:EXCPRV
>SET /UIC=[1,2x]
>LBR DL1:MCR/RP=SPSOV  ! replace object module into MCR.OLB
>PIP DL1:MCR.OBJ/AP=SPSOV.OBJ ! for terminal-driver dependent
    ! modules only
>PIP SPSOV.OBJ;*/DE ! clean up
```

```
!
! invoke SYSGEN3 to build privileged task MCR
!
>SET /UIC=[200,200]
>@SYSGEN3  ! refer to SYSGEN Manual for details
     .
     .
>ASN DL0:=SY:  ! ensure proper assignments
>ASN DL0:=LB:  !    for VMR
>VMR
Enter filename: RSX11M
VMR>REM MCR...  ! remove old MCR
VMR>INS MCR  ! install new MCR
VMR>^Z
!
! end of procedure
!
```

To rebuild the multiuser MCR, use the following procedure:

```
!
! DL0: is mounted with RSXM35 baseline disk
!
>MOU DL1:EXCPRV
>SET /UIC=[1,2x]
>LBR DL1:MCR/RP=SPSOV ! replace object module into MCR.OLB
>PIP DL1:MCR.OBJ/AP=SPSOV.OBJ ! for terminal-driver dependent
     !   modules only
>PIP SPSOV.OBJ;*/DE ! clean up
!
! invoke SYSGEN3 to build privileged task MCM
!
>SET /UIC=[200,200]
>@SYSGEN3  ! refer to SYSGEN Manual for details
     .
     .
>ASN DL0:=SY:  ! ensure proper assignments
>ASN DL0:=LB:  !    for VMR
>VMR
Enter filename: RSX11M
VMR>REM MCR...  ! remove old MCR primary
VMR>REM ...MCR  ! remove old MCR secondary
VMR>INS MCRMU  ! install new MCR primary
VMR>INS SYS  ! install new MCR secondary
VMR>^Z
!
! end of procedure
!
```

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]
>LBR INS/RP=[12,24]INSLB
```

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]
>ASN SY:=MP:
>TKB @INSBLD
>SET /SYSUIC
SYSUIC=[g,m]
>PIP /NV=[1,54]INS.TSK
>VMR
ENTER FILENAME: RSX11M
VMR>REM ...INS
VMR>INS INS
VMR>^Z
>RUN $SHUTUP
```

Hardware bootstrap the system.

The following tables summarize where the relevant MCR files appear on the multiple-media distribution kits:

For RL01/RL02 distribution kit:

MCRSRC	[12,10]*.MAC	All MCR source files
EXCPRV	[12,10]*.MAC	MCR terminal-driver dependent source files
EXCPRV	[1,2x]MCR.OLB [1,2x]MCR.OBJ	MCR object library Concatenated terminal-driver dependent object modules (used by SYSGEN)
RSXM35	[1,1]EXEMC.MLB [11,10]RSXMC.MAC	Prefix files to assemble MCR modules

For RK06/RK07 distribution kits:

CLISRC	[12,10]*.MAC	All MCR source files
RSXM35	[12,10]*.MAC	MCR terminal-driver dependent source files
RSXM35	[1,2x]MCR.OLB [1,2x]MCR.OBJ	MCR object library Concatenated terminal-driver dependent object modules (used by SYSGEN)
RSXM35	[1,1]EXEMC.MLB [11,10]RSXMC.MAC	Prefix files to assemble MCR modules

1.4.2.3 **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 UFD [1,24] of your distribution kit. If you are not sure whether a module belongs in the root or the overlay library, use LBR to scan the module names in the libraries. For instance, to view the module names in DCLO.OLB, enter the following command:

```
MCR:

LBR [1,20]DCLO/LI

DCL:

LIBRARY/LIST [1,20]DCLO
```

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 Chapter 12 of the RSX-11M/M-PLUS System Management Guide for more information on the structure of the DCL task and on the DCL task-building process.

File DCL.CMD, in UFD [23,24] of your distribution kit, can be used either to assemble DCL overlay modules and to re-task-build DCL, or as a template for your own commands. The file contains its own instructions. Note that to use this command directly, you must copy various files into its UFD, and you must appropriately modify any UFDs that they reference. DCL.CMD also shows how to assemble the special module COMMAND.

The following process shows how to reassemble a DCL overlay module and re-task-build DCL. The process is similar for root modules; simply specify DCLR rather than DCLO. See DCL.CMD for details on rebuilding COMMAND.

```
MCR:

SET /UIC=[23,24]
MAC XXX=[11,10]RSXMC/PA:1,[23,10]DCLMAC/PA:1,XXX
SET /UIC=[1,20]
LBR DCLO/RP=[23,24]XXX
SET /UIC=[1,24]
TKB @DCLBLD

DCL:

SET DEFAULT [23,34]
MACRO [11,10]RSXMC/PASS:1,[23,10]DCLMAC/PASS:1,XXX
SET DEFAULT [1,20]
LIBRARY/REPLACE DCLO [23,24]XXX
SET DEFAULT [1,24]
LINK @DCLBLD
```

When you have built a new version of DCL.TSK, you must replace the old copy of DCL as a system CLI, as follows: First, any terminals whose CLI is DCL must be logged off or set to another CLI. Then the following commands must be executed. (Note that these commands are MCR only.)

MCR:

```
CLI /ELIM=DCL
REM DCL
INS $DCL/CLI=YES
CLI /INIT=DCL
```

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 MCR only, and must be put in your system's startup file (if it is not there already).

1.4.3 Patching Object Modules

The Object Module Patch Utility (PAT) incorporates an object patch into an existing object module. DIGITAL publishes a patch file written in MACRO-11 assembly language to correct an error or make a change. You then create and assemble the published file and input it to PAT along with the object file being patched.

You must include the published checksum values in the specification for the input file and the correction files. If the checksum value does not agree with the computed result, PAT reports an error. If the correction file caused the error, check the source file against the original published source file to verify that it was copied correctly. If the input file caused the error, verify that the correct version of the object file is being patched.

Note that it normally is necessary to use LBR to extract the original object module from a library. PAT produces a new object module file containing the corrected object code. When you patch modules from a library, take the input object module from the distributed library and put the patched object file into a copy of that library.

The following guidelines apply to the formats for PAT command input and correction files.

Specify an explicit name, extension, and version number for each file. Use the following conventions for names:

File	Specification
Correction file source	name.PAT
Correction file object	name.POB
Input file object	name.OBJ;n
Patched object	name.OBJ;n+1

Specify checksums for all input and correction files.

The following example illustrates the command formats for updating an object module. First you must create the correction source file using a text editor, and then do the following:

```
>MAC NAME.POB=NAME.PAT
>LBR NAME.OBJ=LIBRARY.OLB;1/EX:NAME
>PAT NAME.OBJ;2=NAME.OBJ;1/CS:3471,NAME.POB/CS:1532
>LBR LIBRARY.OLB;2/RP=NAME.OBJ;2
```

The correction source input file is used to accumulate all patches for a given module.

1.5 NOTES TO RSX-11M V4.1 DOCUMENTATION

The following sections contain notes that correct or supplement the V4.1 documentation.

1.5.1 RSX-11M System Generation and Installation Guide

In Sections 2.3.1.2 and 2.3.2.1, Copying the Big Disk Magnetic Tape Kit Stand-Alone and Copying the Big Disk Magnetic Tape Kit On-Line respectively, the procedure BRU uses to copy the big disk 800 bpi magnetic tape kit (steps 4 through 7 for Section 2.3.1.2 and step 5 for Section 2.3.2.1) is not documented in the correct sequence. The correct procedure (either stand-alone or on-line) to copy the big disk 800 bpi magnetic tape kit is as follows:

1. Invoke the Backup and Restore Utility (BRU) and enter the command line to copy the tape to disk. For example, to copy the distribution tapes to the RM02 disk, enter the following command to BRU:

    ```
    >BRU
    BRU>/BAC:RSXM35/INI/REW/DEN:800/VER
    From:   MM0:
    To:     DR0:

    BRU - Starting Tape 1 on MM0:
    ```

2. BRU copies the first tape and then prompts you for the second tape.

    ```
    BRU - End of Tape 1 on MM0:
    BRU - Mount Tape 2 on MM0:
    BRU - Starting Tape 2 on MM0:
    ```

3. When the second tape finishes copying, BRU prompts you to remount the first tape on the tape drive so that it can do the verification of the first tape.

    ```
    BRU - End of Tape 2 on MM0:
    BRU - Mount Tape 1 on MM0:
    BRU - Starting verify pass Tape 1 on MM0:
    BRU - End of Tape 1 on MM0:
    ```

4. When BRU finishes verifying the first tape, it is ready to verify the second tape. BRU prompts you to remount the second tape on the tape drive.

    ```
    BRU - Mount Tape 2 on MM0:
    BRU - Starting verify pass Tape 2 on MM0:
    BRU - End of Tape 2 on MM0:
    BRU - Completed
    BRU><CTRL/Z>
    >
    ```

At this point, you have completed the tape-to-disk copy of the big disk 800 bpi tape kit and you are ready to generate your RSX-11M system.

1.5.2 RSX-11M/M-PLUS MCR Operations Manual

The /TERM keyword for the MCR SET command now supports the LA50. You should add this new support to page 3-210 of Chapter 3.

1.5.3 RSX-11M/M-PLUS Utilities Manual

1.5.3.1 Correction for PIP Utility – In Chapter 3, page 3-21, the maximum value of the byte location of EOF is 377(octal). Any value over this will result in an error message.

1.5.3.2 Correction for FLX Utility – In Chapter 4, Table 4-1, page 5, the following correction should be made:

/IM:n	Image Mode	The transfer is to be in image mode. Image mode forces fixed-length records. You can use the value n to indicate the desired record length (in octal bytes) for Files-11 output (1000 (8) bytes maximum). If you do not specify n, FLX assumes a record length of 512 (10) bytes.

The documentation currently indicates the value of n in decimal rather than in octal bytes.

1.5.3.3 Format Change the LBR /CR switch – In Section 10.5.2 of the Librarian Utility Program (LBR) chapter, the format for specifying the Create switch (/CR) has changed to the following:

For macro and object libraries –

        outfile/CR[:size:ept:mnt:infiletype=infile]

For universal libraries:

        outfile/CR[:size:ept:mnt:infiletype:libtype=infile]

1.5.3.4 New Switch for File Dump Utility (DMP) – In Chapter 11, File Dump Utility Program (DMP), the following new switch should be added to Table 11-1.

Switch	Description
/LIM:n:m	Specifies the range of bytes n through m of each record or block to be dumped. /OCT is still the default if no format switches are specified.

1.5.3.5 New Error Message for PAT Utility – In Chapter 14, Object Module Patch Utility (PAT), the following new error message should be added to PAT MESSAGES:

UNABLE TO OPEN FILE filename

> Explanation: There is insufficient work space in the internal File Storage Region (FSR) of the PAT utility.

> User Action: Install or run the PAT utility with an increment.

1.5.4 RSX-11M/M-PLUS I/O Drivers Reference Manual

A complete RD51 Fixed Disk/RX50 Flexible Disk Subsystem consists of the fixed and flexible disks and an RQDX1 hardware controller (RQDX1/RD51/RX50). This information should be included on page 5-4.1.

Updated I/O error codes for this manual are listed in the RSX-11M Mini-Reference.

1.5.5 IAS/RSX-11 I/O Operations Reference Manual

1.5.5.1 MTAACP Enhancements - The following sections summarize the enhancements made to the RSX-11 V4.0 and V4.1 Magnetic Tape Ancillary Control Processor (MTAACP) in compliance with the ANSI standard. Detailed descriptions of the enhancements are included in Appendix G of the IAS/RSX I/O Operations Reference Manual.

The RSX-11M V4.0 MTAACP has been enhanced for the following:

- compliance with ANSI and FIP standards

- processing of unlabeled tapes

- Mount, Dismount, and Volume Switch Control

The RSX-11M V4.1 MTAACP has been enhanced to support RMS V2.0. MTAACP supports new file attribute codes (creation, revision, expiration, and backup dates), which read and write the data in the file header using a 64-bit date format. MTAACP performs a computation to convert the date between ASCII and 64-bit binary form.

1.5.5.1.1 Compliance With ANSI and FIPS - DIGITAL now provides tape labeling systems that conform to ANSI Standard X3.27 -- 1978, according to the Federal Information Processing Standard. The MTAACP has been enhanced to comply fully at levels 1 and 2, and to comply at level 3 with the exception of support for user-supplied labels. User-supplied labels may appear on a tape; however, they are only accessible to application programs through the unlabeled tape facility.

The following new features are a result of the implementation of ANSI compliance:

- Seventeen-character file name support

  If file names appear in quotes ("), they may be up to 17 characters long and may contain any characters in the ANSI "a" character set.

  Note that as a result of this change, PIP directory listings of magnetic tapes have a new format. The most noticeable difference is the presence of several spaces preceding the version number. The spaces are included to clearly delimit the version number from the file identifier, which, in accordance with the standard, may contain embedded semicolons.

- Volume Set Identification support

  Previously, the MTAACP only supported Volume Set Identifiers that were the same as the Volume Identifier of the first volume in the set. However, the ANSI standard requires that Volume Identifiers be unique within an installation (thus

preventing a user from choosing names of his or her own).
Full support of the Volume Set Identifier allows the user to
choose a name for this field, which becomes the primary means
of access to a volume using the MOUNT command. This change
was implemented with full compatibility with the previous
releases and requires no changes in procedures for users who
do not wish to use the new feature.

● Control of attributes

MOUNT and the MAG utility allow the specification of block and
record sizes for tapes written on systems that do not write
this information on the tape. They also allow control of the
carriage control attribute for tapes created on systems whose
system code is not recognized by the MTAACP.

Note that the default block size for files written to tapes
from an RSX system may also be modified with MOUNT or the MAG
utility. This value is normally 512 bytes; if it is
modified, additional buffer space must be allocated in the
programs writing to tape.

● Magnetic tape file block size

The PIP block size switch is no longer necessary when reading
files from magtape; FCS will adapt automatically to input
files of varying block size.

PIP directory listing of files on magtape show the number of
magtape blocks. Thus, the file size on a magnetic tape
(represented in terms of variable size blocks) may be
different from the size of the file on disk.

● Volume and file accessibility

In accordance with the ANSI standard, volumes that have a
nonspace character in the volume accessibility byte may be
mounted only by a user who has the privilege to override
accessibility.

The Initialize Volume Utility allows the specification of the
character to be placed in the volume accessibility byte in the
Volume Header Label.

Also in accordance with the standard, files that have a
nonspace character in the file accessibility byte may only be
accessed by a user who has mounted the volume with "override
accessibility" specified. A user who has mounted a volume in
this manner may also write a nonspace character in the file
accessibility byte of the file header label by specifying a
valid "a" character in the high byte of the file protection
word and a low byte of all ones. The file protection word may
be specified at MOUNT time (for example, a file protection
specification of [,,WED,RWD] represents the character "A") or
as an attribute to the CREATE QIO.

1.5.5.1.2 **Support of Unlabeled Tapes** – The MTAACP now permits tapes
to be mounted with the /NOLABEL switch. This allows utilities such as
PIP and higher-level language applications programs (in FORTRAN, for
example) to read and write unlabeled tapes using the standard READ and
WRITE statements in the language.

Files on unlabeled tapes may contain fixed-length records that may be
blocked or unblocked.

In addition, when processing unlabeled tapes, you may request character set translation. An EBCDIC translate table is provided with the ACP; you may add up to three additional installation-defined translate tables. Control of this feature is provided by the MOUNT command and by the MAG utility, as well as by attributes specified by applications programs when you create files.

The file name used when you access an unlabeled tape may specify positioning information, if desired. If no positioning information is desired, the file name has no meaning. However, some utilities, such as PIP, may require a dummy file name to prevent the use of wildcard mode.

Note that there are now three ways tapes may be mounted:

- Mounted for ANSI

- Mounted for Unlabeled Processing

- Mounted foreign

The ability to mount a tape "foreign" is provided for compatibility with other operating systems and has no effect in an RSX-11M system other than reserving the drive for exclusive use. Access to a tape mounted "foreign" is handled by logical QIO directives. Access to a tape mounted "unlabeled" (/NOLABEL) or "ANSI" is through virtual QIOs, primarily those issued by FCS, RMS, the READ and WRITE statements of higher-level languages, and file-oriented utilities such as PIP.

In order to obtain support for unlabeled tapes, you must select ANSI Magtape support at system generation.

1.5.5.1.3 **Mount, Dismount, and Volume Switch Control** – When the operator dismounts an ANSI tape, the default action recommended in the standard is to unload the tape. Depending on which model tape drive is in use, this will either take the drive off line or actually unload the tape.

The DCL /[NO]UNLOAD switch and the MCR /LOCK=[NO]UNLOAD switch specify a different action to be taken. This switch can be applied to either the MOUNT or DISMOUNT command.

When a volume switch is requested, you may, by using the MAG utility, cause error codes to be returned to the program that is reading or writing. (Previously, it was necessary either to mount a tape or to abort the program.) The MAG utility also allows a tape to be initialized if a new tape is needed for output and no previously initialized tape is available.

1.5.5.1.4 **RMS-11 Support** – MTAACP supports new file attribute codes (creation, revision, expiration, and backup dates), which read and write the date in the file header using a 64-bit date format. MTAACP performs a computation to convert the date between ASCII and 64-bit binary form.

1.5.5.2 **FDB Offset Definitions** - The following information should be
added to the FDB Offset definitions listed in Appendix A of the
IAS/RSX-11 I/O Operations Manual:

Offset	Size (in Bytes)	Contents
F.RATT	1	Bit 2 = 1 to indicate the "print file format" (FD.PRN). FCS allows this attribute but does not interpret the format word.
F.NRBD	4	Contains the next record buffer descriptor. The format of the record buffer descriptor is the size word followed by the address of the buffer.
F.RCNM	4	Contains the number of the record for random access operations. The format of the record number is the high-order word followed by the low-order word.
F.VBN	4	Contains the virtual block number. The format of the virtual block number is the high-order word followed by the low-order word.

1.5.5.3 **Shared Access and Block Locking** - Include the following
information in Chapter 2, Section 2.2.1.5 of the IAS/RSX-11 I/O
Operations Reference Manual:

facc

- FA.SHR - Shared access is also a precondition for block locking.

actl

- FA.LKL!FA.EXL - FCS permits limited block locking to coordinate two or more tasks' access to the same file. All tasks accessing the file must open the file for shared access by setting bit FA.SHR in FDB field F.FACC (the field access byte).

1.5.5.4 **Opening a File by File ID Number** - Include the following
information in Chapter 3, Section 3.5 of the IAS/RSX-11 I/O Operations
Reference Manual. This information follows the bulleted list at the
end of the section:

The OFID$W macro call is equivalent to the OFID$U macro call;
invoking either OFID$W or OFID$U opens an existing file by file ID
number for update and extension.

1.5.5.5 **Buffer Flush Routine (.FLUSH)** - Include the following
information in Chapter 4 of the IAS/RSX-11 I/O Operations Reference
Manual as Section 4.17:

When using PUT$S to a disk or ANSI tape file, FCS does not normally write the contents of the block buffer to a file until the block buffer is full or the file is closed. The .FLUSH routine forces the block buffer to be written to the file. You may need this function to protect data from a system crash or if the data needs to be written so that another task can read it from the file.

The following register must be preset before calling this routine:

R0 Must contain the address of the associated FDB.

During output, all registers are preserved, the carry bit is clear or set to indicate success or failure, and the FDB F.ERR byte contains the success or failure code.

1.5.5.6 I/O Error Codes - Updated I/O error codes for this manual are listed in the RSX-11M Mini-Reference.

1.5.6 IAS/RSX-11 SYSTEM LIBRARY ROUTINES REFERENCE MANUAL

The following program section names and SYSLIB routines should be added to Table 1-1 of the IAS/RSX-11 System Library Routines Reference Manual:

Program Section Name	SYSLIB Routines Module Name	Routine Name(s)
PUR$D	CAT5B (data)	$CAT5B
	EDTMG (data)	$EDTMG
PUR$I	CAT5B (instruction)	$CAT5B
	EDTMG (instruction)	$EDTMG

1.5.7 RSX-11M/M-PLUS ODT Reference Manual

The format for the relocation registers in some sections of this manual is incorrect. The correct format for the registers is $nx, where n is a number and x is a letter. The manual, however, incorrectly reverses the letter and number, using an $xn format. For example, where the text references relocation register $R0, the reference should be $0R.

1.5.8 RSX-11M/M-PLUS System Management Guide

1.5.8.1 Virtual Monitor Routine (VMR) - VMR, documented in Chapter 10, now has the following new error messages:

- VMR -- Saved system truncated to n. blocks

  Explanation: (RSX-11S systems only.) VMR tried to save a
  system greater than n. blocks. This results in VMR
  truncating the saved image file to n. (112 decimal
  blocks for 28.K word systems saved on paper tape, 492
  decimal blocks for 123.K word systems saved on RX01
  floppy diskettes, and 496 decimal blocks for 124.K word
  systems saved on all other bootable devices).

- VMR -- Reference to memory beyond end of system image file

  Explanation: VMR attempted to fix a task, load a driver,
  or install a common beyond the space limits of the system
  image file. This results in the command aborting and not
  executing.

The VMR SET /TERM keyword now supports the LA50.

1.5.9 RSX-11M Guide to Writing an I/O Driver Manual

1.5.9.1 Changes to Data Structure Macros - Appendix C

1.5.9.1.1 HWDDF$ - The symbol for the previous mode field of the
program status word, PMODE, is no longer conditional on memory
management (M$$MGE).

The following symbol has been added to the third feature mask symbol
definitions:

```
    F3.SHF=40000            ;MEMORY SHUFFLER SUPPORTED
```

1.5.9.1.2 PKTDF$ - In the kernel AST control block definitions, the
following kernel AST types have been removed:

```
    AK.TBT=203             ;TASK FORCED T-BIT TRAP AST
    AK.DIO=204             ;DELAYED I/O (M-PLUS COMPATIBLE)
```

and the following kernel AST type has been added:

```
    AK.GGF=303             ;GROUP GLOBAL RUNDOWN AST
```

In the Executive pool monitor control flags, the following $POLST
flag:

```
    PC.ALF=4               ;POOL ALLOCATION FAILURE (1=YES)
```

has been expanded into two flags:

```
    PC.ALF=4               ;FAILED TO ALLOCATE LARGE BLOCK (1=YES)
    PC.XAF=10              ;FAILED TO ALLOCATE SMALL BLOCK (1=YES)
```

and the following flag has been added:

```
    PC.XIT=200             ;FORCE POOL MONITOR TASK TO EXIT
```

1.5.9.1.3 **TCBDF$** - In the Task Control Block offset definitions, the conditionalization on the group global event flag use count field, T.GGF, has been changed from group global event flag support or the Send Data directive:

```
        $$$=.
        T.GGF:  .BLKB  1        ;GROUP GLOBAL USE COUNT FOR TASK

                .IF NDF G$$EFN ! R$$SND
        .=$$$
                .ENDC  ;G$$EFN ! R$$SND
```

to group global event flag support and either the Send Data directive or alternate CLI support:

```
        $$$=.
        T.GGF:  .BLKB  1        ;GROUP GLOBAL USE COUNT FOR TASK

                .IF NDF R$$SND & G$$EFN ! A$$CLI & G$$EFN
        .=$$$
                .ENDC  ;R$$SND & G$$EFN ! A$$CLI & G$$EFN
```

1.5.9.1.4 **UCBDF$** - In the Unit Control Block offset definitions, the following MSCP disk driver offsets have been added:

```
                ;
                ; CHARACTERISTICS STORED FROM "GET COMMAND STATUS" END PACKETS
                ;
        000124  U.OTRF=U.VCB+70        ;OUTSTANDING COMMAND STATUS REQUEST REG
        000126  U.CMST=U.VCB+72        ;COMMAND STATUS PROGRESS REGISTER
```

The console driver offsets have moved from U.CNT to U.BUF+2:

```
                .=U.BUF+2
        000026  U.CTCB:  .BLKW  1      ;ADDRESS OF CONSOLE LOGGER TCB
        000030  U.COTQ:  .BLKW  2      ;I/O PACKET LIST QUEUE
        000034  U.RED2:  .BLKW  1      ;REDIRECT UCB ADDRESS
```

A new terminal-dependent characteristic word bit definition has been added:

```
        ;
        ; TERMINAL DEPENDENT CHARACTERISTICS WORD 4 (U.CW4) BIT DEFINITIONS
        ;
        U4.CR=100              ; LOOK FOR CARRIAGE RETURN
```

1.5.9.2 **Asynchronous Buffered I/O** - The following information on buffered input and output should be included in Chapter 2.

Typically, data for input and output requests are transferred directly to and from task memory. To allow the successful transfer of data, the task cannot be checkpointed until the transfer is complete. For most high-speed devices, the transfer occurs quickly enough so that a task does not occupy memory for too long a time. For slow-speed devices, however, some mechanism must be available to avoid binding memory to a task for too long a time while the task is performing I/O.

Using the routines $TSTBF, $INIBF, and $QUEBF in the Executive module IOSUB, a driver can execute an I/O request for a slow-speed device and allow the task to be checkpointed while the request is in progress.

To perform the I/O request, the driver buffers the data in memory allocated to the driver while the task is checkpointed and the I/O request is in progress.

To test whether a task is in a proper state to initiate I/O buffering, the driver calls the $TSTBF routine and passes it the address of the I/O packet. By extracting the address of the task control block (TCB) from the I/O packet, $TSTBF can examine various task attributes. For example, if the task is checkpointable, buffered I/O can be performed. $TSTBF returns to the driver and indicates whether buffered I/O can be performed.

If buffered I/O can be performed, the driver performs two operations. First, it establishes the buffering conditions. For an output request, it copies the task buffers to dynamically allocated pool space. For an input request, it allocates sufficient pool space to receive the incoming data. Second, the driver calls the $INIBF routine to initiate the I/O buffering. $INIBF decrements the task I/O count, increments the task's buffered I/O count in T.TIO, and releases the task for checkpointing and shuffling. If the task is currently blocked, the task state is transformed into a "stopfor" state until the task is unblocked, buffered I/O completes, or both. Checkpointing the task is subject to the normal requirements of an active or "stopfor" state as described in the RSX-11M-PLUS Executive Reference Manual.

After the driver transfers the data, it calls the $QUEBF routine to queue the buffered I/O for completion. $QUEBF sets up a KERNEL asynchronous system trap (KERNEL AST) for the buffered I/O request and, if necessary, unstops the task. When the task is active again, a routine in the Executive module SYSXT notices the outstanding AST and processes it. (If the request is for input, the routine copies the buffered data to task memory.) This mechanism occurs transparently to the task; thus the name KERNEL AST. The routine then calls the driver to deallocate the buffer from pool. $IOFIN completes the processing. These routines are not always present in RSX-11M. If you selected the full duplex terminal driver or networking support during system generation, they are automatically included. To include them in systems without these options, add the symbol A$$BIO= 0 to RSXMC.MAC during Phase I of system generation.

1.5.10 RSX-11M/M-PLUS Task Builder Manual

1.5.10.1 Task-building an F4PRES, FORRES, or FMSRES with/without ECSRES - The following section describes how and why you might want to link an application task to one or more languages, FMS, or a FCS resident library. Trade-offs or memory, speed, flexibility, and ease of use are discussed. For the sake of simplicity, the example of a FORTRAN-IV-PLUS OTS resident library (F4PRES) is used in the rest of this text to represent FORTRAN-IV-PLUS, FORTRAN-IV (FORRES), and FMS (FMSRES) resident libraries.

On RSX-11M systems, resident libraries must reside in dedicated partitions. In general, the presence of a permanent resident library is justified when it is used frequently enough that it saves physical memory, compared to having the FCS or OTS code in the task images of frequently used tasks. For more information on resident libraries, refer to the RSX-11M/M-PLUS Task Builder Manual Section 4.5, Use and Size of Overlay Run-Time Routines, and Section 5.2, Cluster Libraries.

In all cases, the application task need only be linked to properly built resident libraries using the LIBR, COMMON, RESLIB, RESCOM, or CLSTR options.

1.5.10.1.1 **FCSRES** - The types of FCS resident libraries are as follows:

1. Memory-resident overlaid FCSRES

    SYSGEN can automatically generate a memory-resident overlaid FCS resident library that uses one APR of task address space. This FCSRES makes available all of FCS (except little-used routines .CTRL, .PRSDI, and .PPR50), .CSI1, .CSI2, and many other system library routines. See [1,20]FCSRS1BLD.BLD for a list of routines. When FCSRES is built from LB:[1,1]SYSLIB.OLB, it uses 5128 decimal words of physical memory; and when it is built from LBP:[1,1]ANSLIB.OLB, it uses 6624 decimal words. FCSRES uses one APR in either case because it is composed of two memory-resident overlays and a null root segment.

    SYSGEN Phase II and SYSGEN Phase III can link utility tasks to this FCSRES, frequently improving task execution speed and virtual address space, and lessening task image size. This feature, plus the fact that no editing of a source file is required, makes the memory-resident FCSRES easier to use than the non-memory-resident FCSRES.

    This memory-resident FCSRES obviously requires memory-resident support.

    In order to task-build an application task to both the memory-resident FCSRES and F4PRES, it is necessary to take special measures when building the F4PRES. Such a resident library is said to have revectored FCS. Once the F4PRES is built, application tasks may be linked to it with or without linking to FCSRES as well. The TKB cluster library facility, using the CLSTR option, may be used to task-build the application task to two or more resident libraries at a savings in virtual address space.

2. Non-Memory-resident FCSRES

    You can manually assemble and build a non-memory-resident FCSRES from [200,200]FCSRES.MAC. If you build the resident library using LB:[1,1]SYSLIB.OLB, you can produce a 3744 decimal-word, 1-APR FCSRES without editing FCSRES.MAC. If you build the resident library using LB:[1,1]ANSLIB.OLB, then you must edit FCSRES.MAC to remove enough FCS routines from FCSRES to bring it below 4K words. Unedited, FCSRES.MAC builds into a 2-APR, 4448 decimal-word FCSRES using ANSLIB.OLB.

    Unedited, this FCSRES contains all of FCS. Note that .CSI1 and .CSI2, used by FORTRAN-IV and FORTRAN-IV-PLUS ASSIGN and OPEN statements, are not present in the non-memory-resident FCSRES. Their 1426 decimal total bytes will be present either in your application task images if you use ASSIGN or OPEN or they can be included in F4PRES.

    If you want to link RSX utilities to a non-memory-resident FCSRES, you must create and edit the TKB .CMD and .ODL files and task-build the utilities manually.

    No memory-resident support is required.

1.5.10.1.2 **F4PRES** - Building an optimal F4PRES depends on the specific F4P OTS routines your application tasks use, their need for virtual address space, and the available physical memory on your system.

You must decide which F4P OTS routines are used frequently enough by your applications to warrant their presence in F4PRES. Routines are included or excluded by editing F4PRES.MAC.

The gating factor is often the number of APRs used to map to the resident libraries. For example, if you construct an F4PRES that uses two APRs, clustered with FCSRES, for a total of two APRs, you might have an important privileged application task that only has 1 APR available. In this case, it may be best to edit more routines out of F4PRES, to trim it to 1 APR.

The key to building an F4PRES so that it can be used with the memory-resident FCSRES is that no FCS code is present in the F4PRES, but all subroutine calls to FCS in the F4PRES are resolved when the F4PRES is built. This scheme involves revectoring the FCS calls through the application task image. The scheme is described in detail (including TKB command files) in the RSX-11M/M-PLUS Task Builder Manual, Section 5.2.

1.5.10.1.3 **Options and Tradeoffs** - There are a number of ways of building application tasks with resident libraries.

The following examples assume a minimally sized F4PRES: 4096 decimal words (1 APR) if FCS is not contained in it, and 8192 decimal words (2 APRs) if FCS is contained in it. These numbers will vary according to the F4P OTS routines that you include in F4PRES. It may not be possible for you to construct a useful F4PRES of one or both of these sizes; yours may have to be two APRs or three APRs, respectively.

In the following cases, the "Virtual and Physical Memory" discussions are always relative to a task with no overlays or resident libraries. Your own disk or memory-resident overlays may add overlay run-time routines, autoload vectors, and segment and region descriptors to your task. Section 4.5 discusses this in more detail.

- Case 1

    F4PRES with revectored FCS calls and memory-resident FCSRES used as a cluster library. Link the application task with this TKB option:

        CLSTR=F4PRES,FCSRES:RO

    Uses a total of one APR, making available maximal virtual address space in the application task. This is most appropriate for tasks that can take advantage of the increased virtual address space. On a system with the memory-resident FCSRES, F4P application tasks that do not profit from the increased address space should be built according to case 2, with two LIBR= TKB options. Macro application tasks, of course, can be built with one LIBR=FCSRES:RO TKB option (see case 3 for memory characteristics in this case).

    If FCS routines are called from the task image, the calls are resolved to entry points in FCSRES. (FCS routines might be called by either OTS code in the task image, or your macro subroutines.)

Virtual and physical memory: Uses one APR.   The application
task root incurs a load of 1250 decimal bytes, 32 decimal from
FCSJMP.OBJ, 650 decimal from FCSRES.STB (autoload vectors,
segment and region descriptors), and 568 decimal from the
overlay run-time routines.

Execution speed: Some execution time is consumed when the
overlay run-time code in the task image must change the APRs
from one resident library to another.

- Case 2

    F4PRES with revectored FCS calls and memory-resident FCSRES,
    though not used as a cluster library. Link the application
    task with these TKB options:

        LIBR=FCSRES:RO LIBR=F4PRES:RO

    Uses a total of two APRs for the resident libraries, but there
    is less overhead than with a cluster. This is best for tasks
    that cannot profit by using the extra APR that a cluster could
    make available. If FCSRES is predominantly being used in this
    way (little use of FCSRES linked to RSX utilities or macro
    application tasks, and no clustered FCSRES and F4PRES), then
    you should also consider case 4, where a 2- APR F4PRES
    contains FCS with no cost in autoload vectors, FCSJMP, or
    overlay run-time routines for FCSRES.   Other tasks   can
    reference the resident libraries with one or more TKB LIBR
    options, or with CLSTR.

    If FCS routines are called from the task image, the calls are
    resolved to entry points in FCSRES. (FCS routines might be
    called by either the OTS code in the task image or your macro
    subroutines.)

    Virtual and physical memory: Uses two APRs. The application
    task root incurs a load of 1164 decimal bytes, 32 decimal from
    FCSJMP.OBJ, 650 decimal from FCSRES.STB (autoload vectors,
    segment and region descriptors), and 482 decimal from the
    overlay run-time routines.

    Execution speed: Some execution time is consumed when the
    overlay run-time code in the task image must change the
    mapping of the FCSRES's APR from one overlay to another, but
    less time is used than with a cluster.

- Case 3

    Link to the memory-resident FCSRES and have the OTS code
    present in your task image. Link the application task with
    this TKB option:

        LIBR=FCSRES:RO

    This is appropriate when FCSRES is necessary, but you cannot
    justify having a permanent F4PRES on your system.

    Virtual and physical memory characteristics: one APR is used
    for FCSRES. The OTS code is included in the application task
    image, likely thousands of bytes.   (This may need to be
    overlaid.) The task root also incurs a load of 1132 decimal
    bytes: 650 decimal from FCSRES.STB (autoload vectors, segment
    and region descriptors), and 482 decimal from the overlay
    run-time routines.

Execution speed: Some execution time is consumed when the overlay run-time code in the task image must change the mapping of resident libraries' APRs from one library to another. More execution time is used if you overlay the OTS code in the task image.

● Case 4

Link to F4PRES with revectored FCS, so that FCS code is present in your task image. This combination is never the best choice because F4PRES and your application tasks will include FCSJMP and FCSVEC with no benefit. However, tasks will link and execute correctly. Link the application task with this TKB option:

    LIBR=F4PRES:RO

● Case 5

Link to an F4PRES containing FCS. Link the application task with this TKB option:

    LIBR=F4PRES:RO

This is appropriate when F4PRES is necessary; you do not need a permanent FCSRES on your system, and no critical application tasks would profit from the increased address space of a clustered FCSRES and F4PRES.

This combination tends to contain more OTS code in the same number of APRs than case 6 because only the FCS used by F4PRES is present, leaving more room for OTS code.

If your task contains macro subroutines that use FCS, try to use the FCS routines already contained in F4PRES, like OFNB$, OFID$,or DELET$. Otherwise, the task will contain large amounts of FCS code.

Virtual and physical memory characteristics: two APRs are used. There is no overlay overhead due to the resident library.

Execution speed: There is no overlay overhead due to the resident library.

● Case 6

Link F4PRES to a non-memory-resident FCSRES (LIBR=FCSRES:RO in the F4PRES TKB command file). Link the application task to F4PRES with this TKB option:

    LIBR=F4PRES:RO

If you have a non-memory-resident FCSRES on your system, this may be appropriate. Note that case 5 tends to include more OTS code in the same number of APRs.

If FCS routines are called from the task image, a space problem can occur. (FCS routines might be called by either OTS code in the task image, or your macro subroutines.) FCSRES entry points are available only to a task or resident library linked directly to FCSRES; they are not available to a task linked to F4PRES in this case. Thus, any FCS routine called in your task will bring a number of FCS modules into the task image.

Virtual and physical memory characteristics: two APRs are used. There is no overlay overhead due to the resident library.

Execution speed: There is no overlay overhead due to the resident library.

- Case 7

    Link to a non-memory-resident FCSRES and have the OTS code present in your task image. Link the application task with this TKB option:

        LIBR=FCSRES:RO

    This is appropriate when the non-memory-resident FCSRES is necessary, but you cannot justify having a permanent F4PRES on your system.

    Virtual and physical memory characteristics: one APR is used for FCSRES. The OTS code is included in the application task image, likely thousands of bytes. (This may need to be overlaid.)

    Execution speed: There is no overlay overhead due to the resident library.

1.5.11 RSX-11M/M-PLUS Executive Reference Manual

1.5.11.1 Change to Executive Directives - For the Attach Region (ATRG$) and Create Region (CRRG$) directives, the Executive will return the directive error status IE.PNS if the region specified in the directive is not a common or if the common is not installed.

1.5.11.2 Corrections to Executive Reference Manual - The following changes should be made to the RSX-11M/M-PLUS Executive Reference Manual. All the changes apply to Chapter 5.

- On page 5-7, the first sentence under "Macro Expansion:" should read "Most of the directive descriptions expand the $ form of the macro."

- ALTP$

    On page 5-11, add the following return code and definition to the list of DSW return codes:

    IE.RSU -- Resource (the task's header) unavailable because task is checkpointed with outstanding I/O.

- CINT$

    1. On page 5-18, add the following text to the last paragraph:

        However, be aware that including the symbol definition file can cause references to system subroutines to be resolved from that file instead of from the system library. To avoid this problem, explicitly include the required library modules before specifying the

Executive symbol definition file. Specify the /SS switch with the file so the Task Builder will resolve any symbols that are still undefined. (Specifying the /SS switch prevents the Task Builder from trying to use multiply defined symbols.)

2. For corrections to the CINT$ example, beginning on page 5-25, see the June 1980 issue of the Software Dispatch, Sequence 7.1.1.1 N, pages 79 through 84.

● GCCI$

On page 5-76, the Command Buffer Format list should be in the following order:

```
G.CCDV -- . . .
G.CCUN -- Octal unit number of issuing terminal (1)
G.CCCT -- Number of characters (1)
G.CCCL -- . . .
G.CCTC -- Terminator (1)
G.CCFL -- Flags (1)
              .
              .
              .
G.CCBF -- . . .
```

● GCII$

On page 5-80, add the following Local Symbol Definitions:

```
G.CIBF -- Buffer address
G.CIBL -- Buffer length
G.CICN -- Radix-50 CLI name
G.CIDV -- ASCII terminal name
G.CIUN -- Terminal unit number
```

● QIOW$

On page 5-116, the parameter efn (event flag number) is mandatory in the FORTRAN call.

● RDAF$

On page 5-125, the Macro Call should define buf as the address of a 4-word buffer.

● RPOI$

1. On pages 5-130 and 5-131, the second paragraph in the definitions of itask and task should be changed to the following:

On RSX-11M-PLUS systems, any task may specify a new name for the requested task as long as the requested task is not a CLI task.

For both systems, the requested (specified in the tname parameter) task must be installed in the ...tsk format.

2. On page 5-131, in the Macro Expansion, the line

.ASCII /dev/ ;ASCII NAME OF TI: OF REQUESTED TASK

should be the last line in the expansion.

● RUN$

On pages 5-140 and 5-141, the smg parameter in the FORTRAN Call and the Macro Call is mandatory.

● SCLI$

On page 5-148, in the Macro Expansion, the line

```
.RAD50  /cli/   ;CLI NAME
```

should be the last line in the expansion.

● SDAT$

On page 5-150, the text of the NOTE should be replaced with the following text:

When a local event flag is specified, the flag is set for the sending task. When a common event flag is specified, the flag is set for all tasks. When a group-global event flag is specified, the flag is set for all tasks within the specified group. For all event flags, a significant event is always declared.

● SDRC$

On page 5-153, in the Macro Expansion, change the line

```
.BYTE   2       ;EVENT FLAG NUMBER = 2
```

to

```
.WORD   2        . . .
```

and delete the line

```
.BYTE   16.     ;EXIT STATUS BLOCK CONSTANT
```

● SMSG$

1. On page 5-163, in the Macro Expansion, change the following lines:

```
Change SMSG$  SM.ERR  . . .  to SMSG$ SM.SER  . . .
Change .BYTE  DIC,8.  . . .  to .BYTE 171,8.  . . .
Change .WORD  SM.ERR  . . .  to .WORD SM.SER  . . .
```

and delete from the list of Local Symbol Definitions the line

```
S.MERR -- Error Log Target Identifier
```

2. On page 5-164, change the definition of IE.ULN to "Specified LUN is not assigned to a mass storage device."

● STIM$

On page 5-196, in the list of offsets, S.TICS should be S.TISC.

CHAPTER 2

RMS-11 VERSION 2.0 RELEASE NOTES

CHAPTER 2

RMS-11 VERSION 2.0 RELEASE NOTES

2.1 NEW FEATURES FOR RMS-11 VERSION 2.0

The following new features appear in the RMS-11 V2.0 access methods.

2.1.1 Support for Clustering and Task Independence

The memory-resident library RMSRES can be clustered with any other resident library that supports clustering; this capability allows RMS-11 to share task virtual memory with other libraries. See your FMS-11 and language documentation to determine whether a particular library supports clustering.

Task-independence means that once a program is linked with RMSRES, the library can be rebuilt or replaced without requiring that the task linked to it be rebuilt.

Linking with RMSRES requires use of the RMSRLX.ODL file. Include the element RMSROT in the .ROOT statement of your ODL file, and the indirect reference @LB:[1,1]RMSRLX in your ODL file. If you used RMSRLX.ODL in this manner with V1.8, no change should be required (if you made reference to the RMSALL factor, this is no longer necessary but will do no harm).

To use this library as a cluster library, you simply specify the CLSTR option when linking the task.

2.1.2 Subset Library Support

Two versions of the RMS-11 resident library are provided. RMSRES is a full-function library that supports all file organizations. RMSRESSUB.TSK is a subset library that supports access to only sequential and relative file organizations. (Attempts to access indexed files using this library will result in the error code ER$ENV.)

2.1.3 Disk-Overlaid RMS-11

New versions of the RMS-11 V1.8 ODL files are provided on the distribution kit. These ODL files are:

● RMS11S.ODL

  Designed to use minimal virtual address space for support of sequential and relative file organizations, this ODL file adds about 6.5KB to the task size.

- RMS11X.ODL

  Designed to use minimal virtual address space for full support
  of all file organizations, this ODL file is structured to add
  about 10KB to the task size.

- RMS12X.ODL

  This ODL file adds about 12KB to the task size, supports all
  file organizations, and offers a good compromise between
  performance for record operations on indexed files and use of
  task virtual address space.

- RMS11.ODL.

  This is a prototype ODL file that you can tailor to your own
  needs.

The V1.8 ODL files will still work with RMS-11 V2.0 but the new
versions of the ODL files will be more efficient, generally requiring
less memory and less disk space for the task image. For more
information on V1.8 comparisons, see Section 2.3.

In addition, two new ODL files are provided with V2.0:

- RMS12S.ODL

  Supporting only sequential and relative file organizations,
  this ODL file adds about 9KB to the task size and is designed
  to offer a good compromise between performance and use of task
  virtual address space.

- DAP11X.ODL

  Structured to add about 14KB to the task size, this ODL file
  is designed to use minimal virtual address space for support
  of all file organizations and (on systems with the required
  DECnet support) remote access facilities.

2.1.4 Remote File and Record Access via DECnet

If suitable DECnet facilities exist on your system and on the target
system, RMS-11 V2.0 will allow file and record access to files
residing on other network nodes, provided that such nodes include an
RMS-11-based file access listener (FAL).

For most purposes, remote access is indistinguishable from local
access, although performance may not be equivalent. The following
general limitations apply:

- RMS-11 generally does not support remote functions that are
  not supported locally.

- Certain RMS-11 functions (wildcard support, PARSE, SEARCH,
  ENTER, REMOVE, RENAME, and transmission of device, directory,
  and file identifiers) are not supported by the data access
  protocol (DAP) used, and hence cannot be executed remotely.

- Certain FALs do not support the full set of RMS-11 functions
  expressible via DAP, and hence further limit remote access.
  For example, the current RSTS/E RMS-11 FAL does not support
  record access to indexed files.

● High-level languages may not allow expression of the file specification necessary to establish contact with a remote node.

To operate upon a remote file, your program must include a node specification for the remote file and you must include the RMS-11 remote access code when you build your task.

2.1.4.1 Remote Node Specification - You must include a remote node specification at the start of the file name string or the default name string you provide to an OPEN, CREATE, or ERASE operation. In addition, your file name string and default name string must conform to the DIGITAL Command Language (DCL) file specification syntax rules, and the file specification resulting from the merging of these two strings must conform to the file specification conventions on the destination system as well.

In general, a file specification consists of the following elements (in the order and with the delimiters shown):

    node::device:[directory]filename.type;version

where the elements beyond the **node** specification conform to the conventions of the target system. Elements not present will be defaulted according to the conventions of the target system.

The **node** element is of the form:

    node"user password"::

where **node** is the (required) destination node name and **"user password"** is an (optional) access control string containing login information acceptable to the remote system.

If **"user password"** is provided, the device and directory defaults and access privileges of that remote account are acquired; if not, the device and directory defaults and access privileges of the default DECnet account on the remote system are acquired.

2.1.4.2 Linking Programs with RMSDAP Support - To link your program with remote RMSDAP modules, you can either use the disk-overlaid version or reference the DAPRES resident library:

● To link with the disk-overlaid version, reference the DAP11X.ODL file in your ODL file. As with other RMS-11 ODL files, you reference the RMS-11 root modules as RMSROT, and specify other RMS-11 modules as the cotree RMSALL.

● To link with the DAPRES resident library, use the ODL file DAPRLX, and specify RMSRES and DAPRES in your TKB command file. RMSRES and DAPRES can be specified as single resident libraries or as clustered resident libraries.

Note that DAPRES is a clusterable, task-independent library as defined previously for RMSRES.

2.1.4.3 RMSDAP Environments - RMS-11-based FALs are currently available on VAX/VMS, RSTS/E, RSX-11M, and RSX-11M-PLUS systems. The version of data access protocol that is used must be Version 5.6 or later; this means that you must have at least Version 2.0 of DECnet/E, Version 3.1 of DECnet for RSX-11M, Version 1.1 of DECnet for RSX-11M-PLUS, or Version 2.0 of DECnet/VAX.

For more information on remote access using RMS-11, see Appendix B of the RSX-11M/M-PLUS RMS-11 User's Guide.

2.1.5 New File and Directory Operations

RMS-11 V2.0 supports several new file and directory operations.

2.1.5.1 ENTER/REMOVE - Your program can use the ENTER operation to create a directory entry for a file. You can delete the directory entry for a file by using the REMOVE operation; this operation does not affect either the existence of the file or the file contents.

2.1.5.2 RENAME - RENAME is a new RMS-11 operation that can be used to replace the directory entry for a file (that is, change the name of a file). The file specification for the new directory entry must not specify a new device or node for the file, but can specify a different directory, file name, file type, and file version.

2.1.5.3 PARSE - PARSE is a new RMS-11 operation that can be used to analyze a file specification.

The PARSE operation merges the file specification with the default file specification and with the system defaults, and analyzes the result. If you specify wildcarding, the PARSE operation constructs a match-pattern; this match-pattern can be used by later wildcard SEARCH operations.

2.1.5.4 SEARCH - SEARCH is a new RMS-11 operation that can be used to find the next (or first) file that matches a match-pattern (constructed as the expanded string by a previous PARSE operation).

2.1.6 Wildcarding

You specify wildcarding by including wildcard characters in the file specification (or the default file specification). Wildcards can be specified in the directory entry, file name, type, and version.

See the RSX-11M/M-PLUS RMS-11 Macro Programmer's Guide for a full explanation of how to use the new directory operations and wildcarding. You may also want to examine the demonstration programs that are provided.

2.1.7 Random Access to Fixed-Format Disk Sequential Files

For disk sequential files with records of fixed format, random access for GET, FIND, and PUT operations can be specified. This kind of random access is very similar to random access to relative files, with some exceptions.

Your program specifies key access by setting the RB$KEY code in the 1-byte RAC field of the RAB, and by specifying the address of a "relative record number" in the KBF field of the RAB. Key size must be specified as 4 in the KSZ field of the RAB.

Some differences from relative file handling are the following:

- There is no initialization of the space in the file. If the first operation is a random PUT to the file at relative record number 5, the "cells" preceding record number 5 may contain "garbage": that is, whatever data happened to be there.

  (This is similar to VAX-11 RMS.)

- There is also no maximum record number (except as limited by the 4-byte specification of the key). If your program specifies a random PUT operation at relative record number 999, the file will be extended to the appropriate size (but not initialized).

- A PUT operation overwrites any existing record in the target location.

- There is no support for DELETE operations to a sequential file.

2.1.8 Increased Device Independence

Files with stream and VFC record formats can now be created on unit-record devices to avoid the need for special-case code in copy-type operations.

- In the case of VFC files, the record header is thrown away on output unless it is a "print format" file, which RMS-11 now supports as VAX-11 RMS does.

- For stream files, if none of the 3 carriage control bits is set (print format, carriage control, or FORTRAN carriage control), and if the last character is not a line feed, form feed, or vertical tab, then carriage-return/line feed (CR/LF) is appended at the end of the record.

- For stream files, where either the carriage control or FORTRAN carriage control attribute is set, if the last two characters of the record are CR/LF, the trailing CR/LF is stripped off, and then definition of the carriage control (CR or FTN) attribute is applied.

For similar ease-of-copying reasons, RMS-11 now allows creation of relative and indexed files for output to nondisk devices (although for magnetic tape, record format must be variable length or fixed length).

2.1.9 More Usable Block I/O Facilities

The following enhancements have been made for block I/O:

- Block I/O can now be used to copy files without the need to "fix up" the file header attributes.

  Block I/O READ operations now generally respect the logical EOF header attribute (exceptions are documented in the RSX-11M/M-PLUS RMS-11 User's Guide and in the RSX-11M/M-PLUS RMS-11 Macro Programmer's Guide).

- Block I/O can also be used for sequential access, by zeroing the RAB BKT field, allowing sequential copy and update operations.

- Extends are now performed automatically, when necessary, during block I/O access. The only exception is for sequential files that are accessed write-shared. In this case, automatic extends are not allowed because there is no mechanism for coordinating the updating of the file header attributes.

2.1.10 Sharing for Relative and Indexed Files

For increased symmetry, flexibility, and compatibility with VAX-11 RMS, RMS-11 V2.0 incorporates a change to previous file sharing semantics for relative and indexed files on RSX-11M/M-PLUS systems. This change affects only write-accessors who specify read sharing (the default) for record access to a relative or indexed file. Although it is largely transparent in functional terms, in some cases it will have a noticeable effect upon the performance of the writer.

In previous versions of RMS-11, the performance of the writer was dependent upon whether any write-sharing read-only accessors were already present when the writer requested access:

1. If such a reader was present, the writer obtained shared access to the file, although additional writers were prohibited. RMS-11 internal algorithms did not allow the writer to "cache" buckets unless the bucket remained locked; this limitation affected performance for certain types of operations.

2. If no such reader was present, the writer obtained exclusive access to the file. No subsequent RMS-11 readers could obtain access, regardless of their sharing specifications. RMS-11 internal algorithms allowed this writer to cache buckets when possible, thus improving performance by reducing disk I/O.

RMS-11 V2.0 has been changed to eliminate this access-order dependency:

1. When a write-accessor specifies read-sharing (the default), shared access is always granted as described in (1) above. Subsequent write-sharing readers can obtain access even if the writer was the first to access the file.

2. A writer desiring the performance advantages of exclusive access can specify "no sharing" (new for RMS-11 V2.0), in which case access is granted as described in (2) above.

The performance difference between case 1 and case 2 will be most evident for sequential PUT operations to a relative file with multiple records in each bucket, or for sequential GET/UPDATE sequences to a relative file with multiple records per bucket or to an indexed file by primary key. If extra RMS-11 internal buffers are provided to improve performance, they will be ineffective for a writer unless "no sharing" is specified.

2.1.11 RMS-11 Utilities

2.1.11.1 New File Design Utility - The RMS-11 File Design Utility (RMSDES) is a new utility that allows you to interactively design and create files. RMSDES supersedes the RMS-11 File Definition Utility (RMSDEF).

Some of the features of RMSDES are as follows:

- Full RMS-11 file structure capabilities -- You can design and create any type of RMS-11 file to contain your data records, tailoring it to meet your program's data processing requirements exactly.

- Error detection -- RMSDES can detect many kinds of errors and omissions in the attribute information you supply. Because RMSDES is interactive, you can correct the errors immediately, before you create and load records into the file.

- Saved file description -- You can save a description of a file design for future use, whether or not you have created a file based on the description. This is useful if you want to complete an unfinished file design at a later date, or modify a file design to accommodate a new use.

- Copied file description -- RMSDES can construct a file description based on the attributes of an existing data file. You can either create a new file like the existing file, or modify the file design before creating a new file.

- Default calculations -- RMSDES can calculate defaults for many attributes.

- Extensive help facilities are provided for RMSDES.

2.1.11.2 New Features for RMSDSP - RMSDSP has a completely reorganized and much more readable format. It also includes the following new features:

- The combination of options /BP/FU now displays more complete information about the contents of backup container files.

- The size field lists high block as well.

- The protection information is "decomposed" into system, owner, group, and world protection fields.

- RMSDSP lists the file owner.

2.1.11.3 New Features for RMSCNV - RMSCNV has been rewritten and is
smaller and generally faster than the 1.8 version. It has the
following new features:

- The /CA switch can be used to dynamically create an output
  file, by copying file attributes from another file. (That is,
  you do not need to run RMSDES or RMSDEF to create the output
  file before doing the conversion if you already have a
  prototype file whose attributes can be copied for the output
  file.)

- The /IM switch can be used to copy files in image (block)
  mode.

- The /EO switch can be used to change ASCII stream files that
  end with a CTRL/Z character to end with null characters. This
  switch is provided for help in converting certain non-RMS-11
  stream files into a format readable by RMS-11 V2.0. (See also
  Section 2.3.6, Other RMS-11 Version 1.8 Compatibility Notes.)

- The /KN:keyname switch can be used to convert files based on a
  key name (as opposed to a key number, as in the /KR switch).

- The /ML switch allows you to explicitly set the limit of
  memory allocation.

- The /SL switch, summary listing, has been significantly
  enhanced. It now produces several summaries: summaries of
  options used for the conversion, summaries of the file
  attributes for the files involved in the conversion, a message
  summary of errors and warnings, and a processing summary.

- Remote access capability is provided for suitable
  configurations. Your system and the target system must
  include DECnet support (at least V2.0 of DECnet/E). The
  target system must also include an RMS-11-based file access
  listener (FAL), and you must rebuild RMSCNV to include the
  remote access capability (see Section 2.8.3, Utility
  Configurations).

2.1.11.4 Additional Utility Features -

- RMSIFL has been rewritten and incorporates a number of bug
  fixes.

  In addition, the algorithms for allocating work file space for
  sorting have been changed; in most cases, the amount of space
  needed for the sort should be decreased.

  The /DE switch has been enhanced to allow specification of a
  different device for RMSIFL's temporary files.

- Command and ODL files are provided for rebuilding the RMS-11
  utilities to use different forms of the RMS-11 access methods.
  See Section 2.8.3, Utility Configurations.

- All utilities now use lowercase format for messages and
  displays.

- The error message format has been changed for all utilities
  except RMSDEF and RMSIFL.

2.2 SUMMARIES OF TECHNICAL CHANGES

The following sections summarize the new software features for RMS-11.
The summaries of technical changes also appear in the front matter of
each manual.

The title of each manual is followed by a letter indicating whether
the manual is a new (N) or revised (R) version.

In addition to the new and revised manuals listed in the following
sections, the RMS-11 V2.0 documentation set includes a new manual,
RSX-11M/M-PLUS RMS-11: An Introduction, which introduces RMS-11
concepts, terminology, and operations. The RSX-11M Mini-Reference
also contains a new section on RMS-11 utilities and error codes.

                               NOTE

               All new RMS-11 features are fully
               accessible only to MACRO-11 programmers.
               See   your   high-level   language
               documentation for supported features.

2.2.1 RSX-11M/M-PLUS RMS-11 User's Guide (R)

This revision of the RSX-11M/M-PLUS RMS-11 User's Guide documents the
following technical changes:

- RMS-11 Version 2.0 supports random access to fixed-format disk
  sequential files and sequential block access to disk files of
  any format and organization.

- The RMS-11 Version 2.0 resident libraries are task
  independent. This means that once a program is linked with
  this library, the library can be rebuilt or replaced without
  requiring that the task linked to it be rebuilt.

- RMS-11 Version 2.0 contains no library equivalent to the
  RMSSEQ memory-resident library included with RMS-11 Version
  1.8. The RMSRES resident library or the disk-resident ODL
  files can be used to obtain equivalent functionality and
  performance.

- New versions of the RMS-11 Version 1.8 ODL files are provided.
  These ODL files are RMS11S.ODL, RMS11X.ODL, RMS12X.ODL, and
  RMS11.ODL. The Version 1.8 ODL files will still work with
  Version 2.0, but the new versions will be more efficient.
  RMS-11 V1.8 ODL structures other than RMS11S.ODL, RMS11X.ODL,
  and RMS12X.ODL may not work correctly with the RMS-11 V2.0
  code; when in doubt, verify them by comparison with the V2.0
  RMS11.ODL file. In addition, two new ODL files are provided
  with Version 2.0: RMS12S.ODL and DAP11X.ODL.

- Files with stream and VFC records can now be created on
  unit-record devices to avoid the need for special-case code in
  copy-type operations.

    - For VFC files, the record header is thrown away on output
      unless the file is a "print format" file.

- For stream files, if none of the 3 carriage control bits is set (print format, carriage control, or FORTRAN carriage control), and if the last character is not a linefeed, formfeed, or vertical tab, the carriage-return/linefeed (CR/LF) is appended at the end of the record.

- For stream files, if either the carriage control or FORTRAN carriage control attribute is set, and if the last 2 characters of the record are CR/LF, the trailing CR/LF is stripped off and then definition of the carriage control attribute (CR or FTN) is applied.

- For similar ease-of-copying reasons, RMS-11 now allows creation of relative and indexed files for output to nondisk devices (for magnetic tape, however, the record format must be variable length or fixed length).

- The RMS-11 File Design Utility (RMSDES) is a new utility that allows you to design and create files interactively. It is fully documented in the RSX-11M/M-PLUS RMS-11 Utilities manual.

- RMS-11 Version 2.0 supports five new directory operations: ENTER, PARSE, REMOVE, RENAME, and SEARCH. These operations are fully documented in the RSX-11M/M-PLUS RMS-11 Macro Programmer's Guide.

- RMS-11 Version 2.0 supports a new wildcard file specification facility and a new print-record output handling format. These are also fully documented in the macro programmer's guide.

- User-provided interlocks allow a special, limited form of sequential file sharing among a group of accessors that includes at most one read/write accessor and any number of read-only accessors.

- If suitable DECnet facilities exist on your system and on the target system, RMS-11 Version 2.0 will allow file and record access to files on remote network nodes, if those nodes include an RMS-11-based file access listener (FAL).

- For magnetic tape, RMS-11 now allows fixed-format records to be less than 18 bytes.

- Files with stream or VFC records can now be created on unit-record devices. In addition, RMS-11 now allows the creation of relative and indexed files for output to nondisk devices, although they will be treated as sequential files.

- <CTRL/Z> and <ESC> are no longer recognized as record terminators for stream files, and <CTRL/Z> is no longer recognized as a file terminator for stream files.

- RMS-11 Version 2.0 pads stream files with null characters, to the high block of the file (not just to the end of the current block).

- The memory-resident library RMSRES can be clustered with any other resident library that supports clustering.

- An optional subset library, containing support for sequential and relative files only, is available.

2.2.2 RSX-11M/M-PLUS RMS-11 Macro Programmer's Guide (R)

This revision of the RSX-11M/M-PLUS RMS-11 Macro Programmer's Guide documents the following technical changes:

- The new operation macros $ENTER, $PARSE, $REMOVE, $RENAME, and $SEARCH are documented, along with the related NAM block fields FNB, RSA, RSL, and RSS.

- The new facility for wildcard file specification is documented.

- The extension of access sharing is documented, along with the related masks FB$UPI and FB$NIL for the SHR field of the FAB.

- Random access to a sequential file with fixed-length records (similar to random access to a relative file) is documented.

- The new print-format record-output handling is documented, along with the related symbol FB$PRN for the RAT field of the FAB.

- The new sequential block access is documented; the previous block access (formerly called block I/O) is now called VBN access (virtual block number access).

- Block access can now be used to copy RMS-11 files without the need to modify the file's attributes manually.

- The addition of the success handler facility for file operation macros ($CLOSE, $CREATE, $DISPLAY, $ERASE, $EXTEND, and $OPEN) is documented.

- Increased device transparency for record access copy operations is supported. VFC and stream record formats are supported on unit-record devices. Relative and indexed files can be created for record access on nondisk devices, although they will appear as and be processed as sequential files there.

- The obsolete RMS-11 initialization macros $INIT and $INITIF are no longer documented. These macros are now defined as no-ops in the RMS-11 macro library RMSMAC.MLB; their previous functions are no longer needed because RMS-11 is now self-initializing. However, programs that use the $INIT and $INITIF macros in their previous senses remain valid under RMS-11 Version 2.0.

- Each XAB type now has a distinct name; the following are the new names:

      ALL block   Area allocation XAB
      DAT block   File date XAB
      KEY block   File key XAB
      PRO block   File protection XAB
      SUM block   File summary XAB

- The following symbol declaration macros are documented:

      FAB$BT   Declare FAB value and mask symbols
      NAM$BT   Declare NAM block value and mask symbols
      RAB$BT   Declare RAB value and mask symbols
      XAB$BT   Declare XAB value and mask symbols
      XBAOF$ Declare ALL block symbols
      XBDOF$ Declare DAT block symbols
      XBKOF$ Declare KEY block symbols

```
        XBPOF$      Declare PRO block symbols
        XBSOF$      Declare SUM block symbols
```

- The description of each operation macro includes the use and meaning of each associated control block field.

- The value of each RMS-11 user symbol is documented.

- The structure of each RMS-11 user control block is documented.

- The FAB has a new LRL (longest record length) field for sequential files (corresponding to the VAX-11 RMS XAB LRL field).

- The date/time XABs have changed in size from 36 to 46 octal bytes.

- For ANSI magnetic tape, RMS-11 allows fixed-format records to be less than 18 bytes.

- <CTRL/Z> and <ESC> are no longer recognized as record terminators for stream files; and <CTRL/Z> is no longer recognized as a file terminator for stream files.

- RMS-11 now pads stream files with null characters to the high block of the file (not just to the end of the current block). This means that RMS-11-created stream files can be read by programs that do not recognize the EOF value from the file header.

2.2.3 RSX-11M/M-PLUS RMS-11 Utilities (N)

The RSX-11M/M-PLUS RMS-11 Utilities manual is a new manual for this release, replacing Chapter 9 of the RMS-11 User's Guide for RMS-11 Version 1.8. It documents the following technical changes:

- The RMS-11 File Design Utility (RMSDES) is a new utility that allows you to interactively design and create files. RMSDES supersedes the RMS-11 File Definition Utility (RMSDEF).

- The RMS-11 File Display Utility (RMSDSP) provides more information about magnetic tape files and about backup files on disk and on magnetic tape (container files), and provides an optional detailed display for backup files. The displays also appear in a more readable format, and the size field shows both the allocated and used sizes of the file.

- The /DE switch to the RMS-11 Indexed File Load Utility (RMSIFL) has been enhanced to allow specification of a different device for RMSIFL's temporary files.

- The /SL switch to the RMS-11 File Conversion Utility (RMSCNV), the File Back-Up Utility (RMSBCK), and the File Restoration Utility (RMSRST) has been modified to append summary listings of processing to a single file if that file exists.

- Five switches have been added to the RMS-11 File Conversion Utility (RMSCNV): the /CA switch, the /EO switch, the /IM switch, the /KN switch, and the /ML switch. The /CA switch creates an output file with the attributes of an existing file; the /EO switch changes ASCII stream files that end

with a CTRL/Z character to end with null characters; the /IM switch reads and writes records in block mode; the /KN switch reads an indexed file using a key name; and the /ML switch explicitly sets the limit of memory allocation. The /PD switch no longer requires a number sign (#) before an octal padding character or quotation marks around an ASCII padding character unless it is a lowercase character, a space, or a tab.

- In addition, RMSCNV file conversion operations can now specify files on remote nodes. On systems with DECnet network capabilities and on which RMSCNV has Data Access Protocol (DAP) support, RMSCNV file conversions can take place not only locally, but over the network as well. You can convert files between a local and a remote node, or between two remote nodes.

- One switch has been added to the RMS-11 File Restoration Utility (RMSRST): /CV. The /CV switch converts file version numbers from octal to decimal for files created and backed up on RSX-11M/M-PLUS systems and restored to VAX/VMS systems; and from decimal to octal for files created and backed up on VAX/VMS systems and restored to RSX-11M/M-PLUS systems.

- All utilities now display information in lowercase letters, and the error message format for all utilities (except RMSDEF) has been changed.

2.3 COMPARISONS WITH RMS-11 VERSION 1.8

Conversion from RMS-11 V1.8 to RMS-11 V2.0 has been made transparent where possible. Structural changes required by feature additions, however, may be significant to some programs.

2.3.1 Memory-Resident Library (RMSRES)

RMSRES now contains all RMS-11 support code save for minor linkage code in the task. The library is also:

- "Task-independent," which means that it can be rebuilt or replaced and the tasks that use it need not be rebuilt

- "Clusterable," which means that it can share task virtual memory with other libraries that support "clustering" (see your FMS and language documentation)

These features required changes in the RMSRLX.ODL file, the file normally used to build tasks using RMSRES. Tasks that used RMSRLX.ODL with RMS-11 V1.8 should continue to build properly using the new RMSRLX.ODL and RMSRES. Build files that used RMSRES but did not use RMSRLX.ODL with RMS-11 V1.8 must be changed to use (or incorporate the contents of) the new RMSRLX.ODL file.

2.3.2 Memory-Resident Library (RMSSEQ)

RMS-11 V2.0 contains no library equivalent to the RMSSEQ memory-resident library included with RMS-11 V1.8. RMSSEQ offered no advantages in use of task virtual memory, and no advantages in use of system physical memory which are not equally attainable through use of

disk-overlaid RMS-11 at negligible performance cost. In addition, structural constraints made it impossible for RMSSEQ to be clusterable or task-independent.

The improved packaging of the full-function library RMSRES should make it a suitable replacement for RMSSEQ in most environments. Where this is inappropriate, disk-overlaid or non-overlaid RMS-11 can be used.

2.3.3 Memory-Resident Library (DAPRES)

If your system includes the necessary DECnet facilities, the new resident library DAPRES provides support for remote file and record access as well as local access. Like RMSRES, DAPRES is both task-independent and clusterable. In particular, a task using RMSRES may include remote-access support via DAPRES with negligible increase in task virtual address space requirement by clustering the two libraries: the file DAPRLX.ODL should be used to generate this configuration.

2.3.4 Disk-Overlaid RMS-11

Structural changes to the RMS-11 code were performed so that the RMS-11 V1.8 ODL files RMS11X, RMS12X, and RMS11S would remain valid. Most other ODL structures will probably continue to be valid, although validity cannot be guaranteed for all cases. Use of V1.8 ODL structures with V2.0 code may require more task virtual address space than with V1.8 code.

The new versions of RMS11X, RMS12X, and RMS11S use approximately the same (or slightly less) task virtual address space as their V1.8 counterparts used with the V1.8 code. If your system includes the necessary DECnet facilities, the new file DAP11X.ODL provides support for remote file and record access as well as local access.

The ODL MAKRMSODL.CMD is not provided and should not be used with V2.0 of RMS-11. However, a prototype ODL file, RMS11.ODL, is provided, and contains extensive comments. This file is not meant to be used directly, but may be helpful in designing your own ODL structures or in modifying the structure of the above ODL files to suit your special needs.

A restriction on ODL structures involves the RMS-11 entry point modules ROxxxx (where xxxx represents the leading characters of the RMS-11 operation name, for example, ROPUT). These modules, which were not explicitly referred to in previous RMS-11 ODL files, must not be referred to in RMS-11 ODL files for V2.0; if they are, a multiple definition for symbols of the form $RMxxx will occur.

2.3.5 Non-Overlaid RMS-11

While the total end-to-end size of RMS-11 has decreased despite addition of new features, the space savings and feature additions have not been spread evenly throughout. Most of the added features have occurred in the areas of file/directory operations and record operations for sequential files.

The result is that applications using non-overlaid RMS-11 for sequential-file-only access may experience an increase in virtual address space requirements for the RMS-11 code. If this presents a problem, it should normally be possible to select a suitable RMS-11 disk-overlay structure without noticeable performance impact.

2.3.6 Other RMS-11 Version 1.8 Compatibility Notes

* The utility RMSDFN is no longer supported.

* There is no longer a distinction between "standard" RMS-11 and "11K" RMS-11. All RSX-11M/M-PLUS systems automatically include the appropriate files needed for RMS-11, including indexed support.

* Programs that have been compiled under RMS-11 V1.8 should in virtually all cases still run under V2.0, without recompilation when relinked. Such programs, however, will not be able to take advantage of any new features in Version 2.0 of RMS-11.

* Tasks already built using RMS-11 V1.8 should continue to run correctly (without rebuilding) on RSX-11M V4.1. However, it is possible that RMS-11 V1.8 (or prior) tasks will not execute correctly on versions of RSX-11M after V4.1. RMS-11 V2.0 has been implemented to anticipate this possible change in RSX-11M: this allows you one full release cycle to convert any such tasks to RMS-11 V2.0.

* In general, RMS-11 V2.0 cannot be run on a previous version of RSX-11M.

  RMS-11 V2.0 depends on new features included in the F11ACP and on the new structure of TKB autoload vectors in V4.1 of RSX-11M; do not try to use old versions of TKB to link RMS-11 V2.0 programs.

  Note: The reverse situation can also cause certain problems. Because of the change in TKB autoload vector format, a task that was linked with new TKB but old RMS-11 (V1.8 or before), and that uses RMS-11 asynchronously, will behave unpredictably.

* With RMS-11 V2.0, some changes to the SYSLIB modules have been made; several modules have been placed in SYSLIB to maintain compatibility with previous releases.

  As a result, you must be careful to reference RMSLIB as a "search" library before referring to SYSLIB when building tasks that use non-overlaid RMS-11. (This is only necessary if you need to make an explicit reference to SYSLIB as a "search" library when task-building.)

* MACRO-11 programs using RMS-11 should always check for errors by checking the value of the appropriate STS field.

  With V1.8, the C-bit may have been returned on RMS-11 errors. However, this is not a defined error return and has not been maintained in RMS-11 V2.0.

* <CTRL/Z> and <ESC> are no longer recognized as record terminators for stream files, and <CTRL/Z> is no longer recognized as a file terminator for stream files.

Note that RMSCNV has a new switch, /EO, which can be used to convert stream files which were created by a non-RMS-11 program, and which used <CTRL/Z> as the file terminator.

- The expanded string is now an output from the OPEN and ERASE operations by file-id.

- RMS-11 now makes a check that at least 2 keys fit into a bucket. Therefore, with a bucket size of 1, the maximum key size is 243.

- When block I/O is used, OPEN and DISPLAY operations no longer return relative or indexed file key XAB, allocation XAB, or summary XAB information.

2.4 MISCELLANEOUS INFORMATION

- Note that all the new features provided in RMS-11 are fully accessible only to the MACRO-11 programmer; for high-level languages, check the language documentation for supported features.

- For magnetic tape, RMS-11 now allows ANSI fixed format tape records to be less than 18 bytes. In addition, RMS-11 supports the full set of ANSI tape names as quoted strings. This includes the full set of ANSI 'a' characters, up to 17 bytes.

- Several RMS-11 demonstration programs have been included on the distribution kit. These sample programs are provided as MACRO-11 sources and task images; they illustrate how to use the new PARSE/SEARCH/RENAME/ERASE operations and wildcarding facilities. A GSA source routine is also included to illustrate how to extend the task in the event of RMS-11 pool exhaustion.

  The names of the programs are PARSE, SEARCH, RENAME, ERASE, and GSA.

- The RMSIFL and RMSDES utilities and the remote access (DAP) support code in RMS-11 V2.0 will run only on CPUs that support the EIS instruction set.

2.5 PROBLEMS FIXED WITH RMS-11 VERSION 2.0

- The problem that was reported for RMS-11 V1.8 on GET after FIND on alternate keys (Dispatch article number 48.2.20) has been fixed for RMS-11 V2.0.

- RMSIFL V1.8 did not load indexed files correctly if the key type was integer. (Not all records could be accessed by key value.) This problem has been fixed in the new version of RMSIFL.

- RMS-11 now pads stream files with nulls, to the high block of the file (not just to the end of the current block). This will make RMS-11-created stream files readable by programs that do not recognize the EOF value from the file header.

2.6 KNOWN PROBLEMS WITH RMS-11 VERSION 2.0

- Under some circumstances, the deferred write option may be ignored.

- Use of the mass-insertion feature may occasionally cause the error ER$DME.

- When you use RMSDES and specify a GET command, RMSDES incorrectly requests write access on the GET file. In some cases, an ER$FLK or ER$PRV error may result if your task does not have write access to the file.

2.7 RMS-11 FILES AND PLACEMENT ON THE 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.
RMSBCK.TSK	LB:[1,54]	RMSBCK utility; does not use RMSRES.
RMSRST.TSK	LB:[1,54]	RMSRST utility; does not use RMSRES.
RMSDEF.TSK	LB:[1,54]	RMSDEF utility; does not use RMSRES.
RMSDSP.TSK	LB:[1,54]	RMSDSP utility; does not use RMSRES.
RMSCNV.TSK	LB:[1,54]	RMSCNV utility; does not use RMSRES.
RMSDES.TSK	LB:[1,54]	RMSDES utility; does not use RMSRES.
RMSIFL.TSK	LB:[1,54]	RMSIFL utility; does not use RMSRES.
RMSDES.IDX	LB:[1,2]	Indexed help file used by RMSDES.
RMS11.ODL	LB:[1,1]	Prototype ODL file.
RORMS1.MAC	LB:[1,1]	For use with the prototype ODL.
RMS11S.ODL	LB:[1,1]	ODL file for sequential.
RMS12S.ODL	LB:[1,1]	ODL file for sequential.
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.
DAP11X.ODL	LB:[1,1]	ODL for use with overlaid RMSDAP.
DAPRLX.ODL	LB:[1,1]	ODL for use with DAPRES.
RMSRES.TSK	LB:[1,1]	TSK image for linking against RMSRES.
RMSRES.STB	LB:[1,1]	STB file for RMSRES.
RMSRES.MAP	LB:[1,34]	Map file for RMSRES.

File Name	Destination	Comments
RMSRESSUB.TSK	LB:[1,1]	Subset library.
RMSRESSUB.MAP	LB:[1,34]	Map file for subset library.
DAPRES.TSK	LB:[1,1]	Task image for RMSDAP resident library.
DAPRES.STB	LB:[1,1]	STB file for DAPRES.
DAPRES.MAP	LB:[1,34]	
RMSRES.CMD, RMSRES.ODL	LB:[1,24]	For rebuilding RMSRES.
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.
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.
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.
DESNON.CMD, DESNON.ODL	LB:[1,24]	For rebuilding the overlaid version of RMSDES.
DESNRN.CMD, DESNRN.ODL, RMSDES.ODL	LB:[1,24]	For rebuilding the resident-library version of RMSDES.
DSPNON.CMD, DSPNON.ODL	LB:[1,24]	For rebuilding the overlaid version of RMSDSP.
DSPNRN.CMD, DSPNRN.ODL	LB:[1,24]	For rebuilding the resident-library 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.
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.
RMSUTL.OLB, RMSODL.ODL	LB:[1,24]	For rebuilding several utilities.

File Name	Destination	Comments
GSA.MAC	LB:[200,1]	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:[200,1]	Demonstration programs for the new directory and wildcarding facilities.

Note that all RMSDAP files can be deleted if you are not using RMS-11 to access files on remote nodes.

2.8 RMS-11 VERSION 2.0 INSTALLATION

On most RSX-11M systems, all RMS-11 files are automatically on your system after system generation.

However, for systems that were generated using distribution kits on RL01/02 or RK06/07 disks or the magnetic tape versions of these kits, the RMS-11 files must be transferred after system generation. If your system was not generated using one of these kits, you may want to skip this section, and continue with Section 2.8.2, Startup Command Procedures.

2.8.1 RL01/02 and RK06/07 Procedures

To transfer RMS-11 files to RL01/02 and RK06/07 disks, you must execute the command procedure [1,2]RMSINSTAL.CMD. This procedure is fully documented in the RSX-11M System Generation and Installation Guide.

2.8.2 Startup Command Procedures

If all the RMS-11 files have been transferred to your system, all that remains to be done is to install the resident libraries and RMS-11 utilities at system startup. To aid you in installing them, the file LB:[1,2]STARTUP.CMD contains sample comments that can be edited to become system startup commands.

You must allocate a partition for each resident library you are using, using the SET command(s). Note that you must allocate a partition called RMSRES, and install in that partition either the full-function resident library (RMSRES) or the subset library (RMSRESSUB.TSK). If you always plan to use the same library (no alternating between the full-function and the subset library), and you want to save disk space, you can delete the task image you are not using.

But, if you delete the full-function RMSRES.TSK and plan to use only RMSRESSUB.TSK, you should rename RMSRESSUB.TSK to RMSRES.TSK. Then user programs and RMS-11 utilities will always link against the task named RMSRES.

NOTE

If you are using the subset library and
you rebuild the RMS-11 utilities to use
the subset library, utility functions
involving indexed files will be
invalidated.

You can include the SET commands in your startup command file, or you
may want to include the RMSRES and DAPRES partitions permanently in
your system by doing the following:

● Using the SET commands to include the necessary partitions.

● Using the INSTALL commands to install the tasks in the running
  system.

● Using the SAVE command to save RMSRES and DAPRES into the
  system image.

See your RSX-11M/M-PLUS MCR Operations Manual for more details on
these commands.

2.8.3 Utility Configurations

The RMS-11 utilities that are provided on the distribution kit are
built to use disk-overlaid RMS-11.

2.8.3.1 Utility Command and ODL Files – For each utility, a command
and ODL file is provided which can be used to build the utility using
disk-overlaid RMS-11. The names of the files are the following:

```
For RMSBCK:   BCKNON.CMD,BCKNON.ODL,RMSODL.ODL
For RMSRST:   RSTNON.CMD,RSTNON.ODL,RMSODL.ODL
For RMSCNV:   CNVNON.CMD,CNVNON.ODL
For RMSDSP:   DSPNON.CMD,DSPNON.ODL,RMSODL.ODL
For RMSDES:   DESNON.CMD,DESNON.ODL,RMSDES.ODL
For RMSDEF:   DEFNON.CMD,DEFNON.ODL,RMSODL.ODL
For RMSIFL:   IFLNON.CMD,IFLNON.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, you can use

```
For RMSCNV:   CNVNRR.CMD,CNVNRR.ODL
```

To build RMSCNV using disk-overlaid RMS-11 and RMSDAP, you can use

```
For RMSCNV:   CNVNOO.CMD,CNVNOO.ODL
```

For each utility, a command and ODL file are provided which can be used to build the utility using the resident library RMSRES. The names of the files are the following:

        For RMSBCK: BCKNRN.CMD,BCKNRN.ODL
        For RMSRST: RSTNRN.CMD,RSTNRN.ODL
        For RMSCNV: CNVNRN.CMD,CNVNRN.ODL
        For RMSDSP: DSPNRN.CMD,DSPNRN.ODL
        For RMSDES: DESNRN.CMD,DESNRN.ODL
        For RMSDEF: DEFNRN.CMD,DEFNRN.ODL
        For RMSIFL: IFLNRN.CMD,IFLNRN.ODL

2.8.3.2 Rebuilding the Utilities – To rebuild the utilities, you must take the following steps:

- Log in to a privileged account.

- Set your default account to [1,24] on the system disk.

- Use 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 a system account ([1,54]).

Each command file for a particular utility creates a utility of the correct name. So, for example, CNVNRN.CMD and CNVNON.CMD both produce task images called RMSCNV.TSK; one is built against the resident library and one is built with disk-overlaid RMS-11.

Finally, note that the RMSIFL utility command files include an EXTTSK value that is used to increase the size of internal buffers, thereby allowing RMSIFL to load larger-sized files and files with many area and key definitions.

The default EXTTSK which is specified is fairly large. If you want a smaller version of RMSIFL, you can simply install RMSIFL with a smaller increment using the /INC switch. The /INC switch will override the EXTTSK directive.

The EXTTSK value however, must be greater than 2048 (decimal) words. If the value is too small, you may get the error "IFL memory exhausted," or a SORTs error code.

2.9 REPORTING PROBLEMS

Software Performance Reports (SPRs) allow you to report any problems directly to DIGITAL. The general procedure for submitting an SPR is described in Sections 1.4 and 1.4.1. The following additional information should be submitted with SPRs on RMS-11:

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

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

3. If RMS-11 aborts (or if some other task crashes and RMS-11 appears to be the cause), include a post-mortem dump and a map of the task involved.

4. Include copy(s) of the file(s) involved, in RMSBCK format.

5. If the errors are reproducible, include copies of the files that can cause the error. If the problem is not reproducible, include a copy of the corrupt file if possible.

6. For RMS-11 utilities, include also a description of the command line(s) or interactive session that led to the error.

   Include a listing of the actual error that occurred, if possible.

CHAPTER 3

RSX-11S VERSION 4.1 RELEASE NOTES

CHAPTER 3

RSX-11S VERSION 4.1 RELEASE NOTES

3.1 NEW FEATURES FOR RSX-11S V4.1

RSX-11S V4.1 software is an update of the memory-resident RSX-11S operating system. The major new functions of RSX-11S V4.1 are documented in the following sections.

3.1.1 New Devices

RSX-11S V4.1 supports the same new devices that RSX-11M V4.1 supports.

3.2 NOTES ON SOFTWARE

The following sections describe incompatibilities that exist between RSX-11S V4.1 and previous versions and provide miscellaneous information.

3.2.1 Host Hardware Configuration Requirements for RSX-11S SYSGEN

To do an RSX-11S SYSGEN, the host system must be mapped and have at least 64K words of memory.

3.2.2 Fixing Tasks on RSX-11S with PLAS Support

To correct problems caused by the logic in VMR that determines whether tasks are overlaid or nonoverlaid, VMR now has a new method for determining how tasks (regardless of whether they are overlaid or nonoverlaid) are fixed into memory.

Instead of using two different methods to fix tasks into memory as in previous releases, VMR now requires that all tasks in a RSX-11S system with PLAS support be fixed by issuing the following VMR command line:

    VMR>INS task/FIX=YES

Using the VMR INSTALL /FIX command provides access to the label block, thus allowing VMR to determine whether the task is overlaid or not. Note that this is a significant change from the way tasks were fixed in PLAS systems in previous releases. Only the memory-resident overlaid tasks had to be fixed using this command; now all tasks in the PLAS system must be fixed using the VMR INSTALL /FIX command.

3.2.3 VMR Support for Systems Larger than 124.K Words

VMR for RSX-11M V4.0 did not support fixing tasks above the 124.K word limit even though RSX-11S V4.0 contained support for systems larger than 124.K words. VMR for RSX-11M V4.1 now contains limited support for systems larger than 124.K words. This limited support will enable you to do the following:

- Fix tasks above the 124.K word limit (using FIX)

- Load drivers above the 124.K word limit (using LOAD)

- Install commons above the 124.K word limit (using INS)

However, the support is limited, and there are two restrictions which must be adhered to:

- 11SGEN still creates a maximum system image file of 498(decimal) blocks for a system of 124.K words. To fix tasks above 124.K words, you must first create another copy of the system image file large enough for the system size.

- The VMR SAV command still can only be used for systems smaller than 124.K words.

3.2.4 RSX-11S System Generation

3.2.4.1 Performing an RSX-11S V4.1 System Generation on an RSX-11M-PLUS V1.0 Host - If you intend to perform an RSX-11S V4.1 system generation while running under an RSX-11M-PLUS V1.0 system, you must update the V1.0 indirect object library ([1,24]IND.OLB), and rebuild indirect before performing the system generation.

A concatenated object file, INDFIX.OBJ, is in UFD [1,54] of the RSX-11S distribution kit. (This object file is on the first volume in multimedia kits.) Refer to Section 1.3.2.1 of these Release Notes for more information on this file.

The procedure for updating is as follows:

1. Set the UIC to [1,24].

2. Copy INDFIX.OBJ to the disk that has IND.OLB.

3. Issue the following command:

    LBR IND/RP=INDFIX

4. Assign the logical names IN: and MP: to their proper devices.

5. Verify that RSX11M.STB resides in [1,54].

6. Task-build Indirect: TKB @INDBLD.

7. Install the new indirect task image.

When you have completed this sequence of steps, you are ready to perform an RSX-11S V4.1 system generation.

NOTE

It is not necessary to update indirect
if your host is RSX-11M-PLUS V2.0.

3.3 NOTES TO THE RSX-11S SYSTEM GENERATION AND INSTALLATION GUIDE

The following sections contain information that modifies and
supplements the information in the RSX-11S System Generation and
Installation Guide.

3.3.1 Distribution Kits

RSX-11S V4.1 is now distributed on the following media:

1. One RK06 cartridge disk

2. One RK07 cartridge disk

3. One RL01 cartridge disk

4. One RL02 cartridge disk

5. One magnetic tape, 9-track, 800 bpi, 2400-foot

6. One magnetic tape, 9-track, 1600 bpi, 2400-foot

Note that RSX-11S is no longer distributed on RK05 disks. All
references to the RK05 distribution kit in the manual no longer apply.

3.3.2 Q-Bus Terminal Interfaces

Section 2.1 of the manual implies that you can only choose UNIBUS
terminal interfaces (for example, DH11 and DZ11) for your system. If
you have a Q-bus machine (for example, a PDP-11/23-PLUS), you can
choose a Q-bus terminal interface (for example, a DLV11 or DZV11) for
the Q-bus machine.

3.3.3 Transferring Task Image Files Using FLX

On page 2-7, the FLX command lines to transfer task image files from a
Files-11 format to either an RT-11 format or a DOS-11 format are
incorrect. To transfer an image file from Files-11 format to DOS-11
format, type the following command lines:

    FLX>CT:/ZE/DO
    FLX>/DO=[1,50]SIP.TSK/RS

To transfer an image file from Files-11 format to RT-11 format, type
the following command lines:

    FLX>DX:/ZE/RT
    FLX>DX:/RT=[1,50]SIP.TSK/RS

For a full description of FLX, see the RSX-11M/M-PLUS Utilities
Manual.

3.3.4 SETTIM

The sections of the manual that refer to SETTIM as a separate subroutine (Sections 2.5 and 4.5.1.5) do not apply. SETTIM is a FORTRAN callable subroutine that is stored in SYSLIB. See the RSX-11M/M-PLUS Executive Reference Manual for information on the SETTIM subroutine.

3.3.5 Building a Resident Subset FCS Library

The following information on how to build a resident subset FCS library for RSX-11S V4.1 supersedes the information in Section 4.7.1.3.

Create the resident library by assembling the file [2,200]11SRES.MAC and building it according to the directions given in the file itself. Two new entry points, .FCTYP and .FLUSH, have been added to make subset FCS conform to standard FCS.

The following descriptions of the new entry points supplement the information in 2.7.3.

.FCTYP   Returns with R1 clear to indicate this is the record I/O-only version of FCS.

.FLUSH   This routine is a NOP for record FCS and returns success.

For more information on building, defining, and loading resident libraries, see the RSX-11M/M-PLUS Task Builder Manual and the RSX-11M System Generation and Installation Guide.

3.4 SYSGEN DIALOGUE FROM AN RSX-11M HOST

3.4.1 Introduction

This section contains the actual terminal dialogue from an RSX-11S system generation on an RSX-11M host.

The dialogue and procedures in Phase I are the same for RSX-11M and RSX-11S; for detailed explanations of the Phase I system generation questions, see the RSX-11M System Generation and Installation Guide.

The system generated in the example in this section supports the following hardware configuration for a PDP-11/23-PLUS processor:

```
128K  memory
60    cycle line clock
1     RL02 cartridge disk controller with two drives
1     TU58 DECtape controller with two drives
1     LP-11 line printer controller
1     null device driver
1     DZV11 terminal interface
1     DLV11 terminal interface
```

The following software support is included:

- Large (20K) RSX-11S Executive

- Support for:

        Executive common
        Memory management
        Send/receive directives
        Get mapping context directive
        Address checking
        I/O rundown
        Loadable device drivers
        ASTs
        Powerfail recovery
        Logical device assignment
        Crash notification
        Group global event flags
        Memory parity

- TKTN

- XDT

- System-controlled partitions

- Basic MCR with support for OPEN, ATL/TAL, and full error-message text

- OTL with support for full error-message text and for all OTL devices generated into the RSX-11S system

- SIP with support for full error-message text and for all SIP devices generated into the RSX-11S system

- RSDV1H

- RSX-11S Subset File Control Services (FCS)

The system generation is being performed on an RSX-11M V4.1 RL02 release kit baseline system disk. The RSX-11S release kit consists of one RL02, labeled 11SKIT1.

Questions calling for a YES/NO answer can be answered by pressing the RETURN key for a NO answer.

You can also answer questions asking for a value and displaying a default value by pressing the RETURN key to specify the default answer.

Notes:

- You may want to do a preliminary system generation before you do your actual SYSGEN. To do a preliminary SYSGEN (PREPGEN), you invoke the command file for Phase I of the SYSGEN (@11SGEN) and answer YES to the question, "Do you want to inhibit execution of MCR commands (PREPGEN)?"

Phase I continues, but the answers you give to the questions are placed in a saved answer file instead of being used immediately. When Phase I finishes, you reinvoke 11SGEN.CMD to run the phase again. This time, you do not inhibit MCR command line execution and the SYSGEN procedure uses the answers in the saved answer file.

Doing a PREPGEN saves time and prevents mistakes. The sample dialogue in the following section shows a PREPGEN being done before the actual SYSGEN.

- During Phase I of the system generation procedure, 11SGEN.CMD asks for the device configuration of your system and also which system directives you would like in the Executive. In both cases, you can print out a table of the available options before you answer the question.

3.4.2 Actual Dialogue -- RSX-11S System Generation on an RSX-11M Host

```
   RSX-11M V4.1 BL35-A    124.K MAPPED (BASELINE)
>RED DL:=SY:
>RED DL:=LB:
>MOU DL:RSXM35
>@DL:[1,2]STARTUP
>* PLEASE ENTER TIME AND DATE (HR:MN DD-MMM-YY) [S]: 13:02 15-MAR-83
>TIM 13:02 15-MAR-83
>* ENTER LINE WIDTH OF THIS TERMINAL [D D:132.]:
>SET /BUF=TI:132.
>ACS SY:/BLKS=512.
>@ <EOF>
>MOU DL1:11SKIT1
>PIP DL0:[200,200]/NV=DL1:[2,20]11SGEN.CMD,11SGEN2.CMD
>DMO DL1:
DMO -- TTO:    dismounted from DL1:    *** Final dismount initiated ***
13:11:00   *** DL1: -- Dismount complete


>@11SGEN
>;
>; RSX-11S Version 4.1 BL35-A System Generation done on RSX-11M
>;
>; Copyright (C) 1979,1981,1983
>; Digital Equipment Corporation, Maynard, Mass.
>; All Rights Reserved.
>;
>;
>; RL01/RL02 distribution kit
>;
>*  3. Do you want to inhibit execution of MCR commands (PREPGEN)? [Y/N]: Y
>*  5. Are you generating an unmapped system? [Y/N]:
>*  6. Use an input saved answer file? [Y/N]:
>*  9. Name of output saved answer file [D: SYSSAVED.CMD] [S]:
>;
>; Phase I output saved answers created in file DL0:[200,200]SYSSAVED.CMD;1
>;
>* 16. Enter device for EXCPRV disk when it is ready (ddu:) [D: DL1:] [S]: DL1:
>MOU DL1:EXCPRV
>;
>; Target configuration
>;
>*  1. Processor Type   [D: 11/34]  [S]: 11/23
>*  2. Does processor have a switch register? [Y/N D:N]:
>*  3. Memory size (in K-word blocks) [D R:16.-1920. D:16.]: 128.
>*  4. Include support for K-series devices? [Y/N D:N]:
>*  6. Floating point processor (FP-11) present? [Y/N D:N]: Y
>*  8. Extended instruction set (EIS) present? [Y/N D:N]:
>* 10. Line frequency:    A- 60 Hz    B- 50 Hz   [D: A] [S]:
>* 11. KW11-Y Watchdog timer support? [Y/N D:N]:
>* 12. Memory parity support? [Y/N D:N]: Y
>;
>; The response to the following question specifies the highest interrupt
>; vector.  If you respond with a value less than or equal to 400, SYSGEN
>; will assign the value associated with the highest interrupt vector
>; specified during the Peripheral Section.  Therefore, if your system
>; will include devices that are not specified during the Peripheral
>; Section and which have vectors above 400 (devices such as K-series and
>; certain communication devices), specify that value in the next question.
>;
>* 14. Highest interrupt vector [O R:0-774 D:0]:
>;     For device configuration: '*' Prints device table, '.' Terminates inquiry
>;                                    '?' Prints current configuration
>;     Enter devices and number of controllers for devices which require drivers
>* 15. Devices [S]: *
>;
```

```
>;    Device codes for A/D and industrial/laboratory devices:
>;
>;      AD- AD01-D A/D Converter
>;      AF- AFC11 A/D Converter
>;      AR- AR11 Laboratory Subsystem
>;      IC- ICR11/ICS11 Industrial Control Subsystem
>;      IP- IP11/IP300 Industrial Control Subsystem
>;      IS- DRS11/DSS11 Industrial Control Subsystem
>;      LA- LPA11 Laboratory Subsystem
>;      LS- LPS11 Laboratory Subsystem
>;      UD- UDC11 Universal Digital Controller
>;
>;    Device codes for Interprocessor communication devices:
>;
>;      XB- DA11-B Interprocessor Link
>;      XL- DL11-E Interprocessor Link
>;      XM- DMC11-E/DMR11 Interprocessor Link
>;      XP- DP11 Synchronous Interface
>;      XQ- DQ11 Synchronous Interface
>;      XU- DU11 Synchronous Interface
>;      XW- DUP11 Synchronous Interface
>;
>;    Device codes for unit record devices:
>;
>;      CR- CR11/CM11-E Card Reader
>;      GR- VT11/VS60 Graphics Subsystem
>;      LP- LA11/LP11/LS11/LV11/LN01 Line Printer
>;      PP- PC11 Paper Tape Punch
>;      PR- PC11/PR11 Paper Tape Punch and Reader
>;
>;    Device codes for disk devices:
>;
>;      DB- RP04/RP05/RP06 Disk Drive
>;      DD- TU58 Cartridge DECtape II
>;      DF- RF11 Fixed Head Disk Drive
>;      DK- RK03/RK05 Disk Drive
>;      DL- RL01/RL02 Disk Drive
>;      DM- RK06/RK07 Disk Drive
>;      DP- RP02/RP03 Disk Drive
>;      DR- RM02/RM03/RM05/RM80/RP07 Disk Drive
>;      DS- RS03/RS04 Fixed Head Disk Drive
>;      DT- TU56 DECtape
>;      DU- RA60/RA80/RA81/RC25/RD51/RX50 Disk Drive
>;      DX- RX01 Floppy Disk Drive
>;      DY- RX02 Floppy Disk
>;      EM- ML11 Non-Rotating Electronic Memory
>;
>;    Device codes for Magnetic tape devices:
>;
>;      CT- TU60 Cassette Tape Drive
>;      MF- TU78 Magtape Drive
>;      MM- TU16/TU45/TU77/TE16 Magtape Drive
>;      MS- TS11/TU80/TSV05 Magtape Drive
>;      MT- TU10/TE10/TS03 Magtape Drive
>;
>;    Device codes for terminal interface devices:
>;
>;      YH- DH11/DHV11 Asynchronous Terminal Interface
>;      YJ- DJ11 Asynchronous Terminal Interface
>;      YL- DL11-A/-B/-C/-D/-E/-J/-W Asynchronous Terminal Interface
>;      YZ- DZ11/DZV11 Asynchronous Terminal Interface
>;
>;    Device codes for pseudo devices:
>;
>;      CO- console device
>;      NL- null device
>;
>;     Enter responses as: dev1=number controllers,dev2=number controllers, ...
>;     If a devices is specified without the number of controllers, the number
>;     of controllers defaults to 1.
>;
>;     Example: DK,DM=2,YL,NL.
>;
```

```
>* 15. Devices [S]: DD,DL,LP,NL,YL,YZ.
>;
>; Processor:11/23   Memory Size:128K,Mapped   System:RSX-11S
>;
>; Host configuration
>;
>*  1. Is a line printer available? [Y/N]: Y
>*  3. Does the listing/map device have at least 120 columns? [Y/N]: Y
>*  4. Assembly listings device (ddu:) [D: "NL:"] [S]:
>*  5. Map device for Executive and device drivers (ddu:) [D: DL1:] [S]:
>;
>* EOS * Do you want to: <CR>-continue R-repeat section E-exit P-pause [S]:
>;
>; Executive Options
>;
>;     Answer Y(ES) if the following support is desired
>;
>*  6. Executive Common? [Y/N]: Y
>*  7. Memory management (PLAS) directives? [Y/N]: Y
>*  8. Send/Receive by reference directives? [Y/N]: Y
>*  9. Get mapping context directive? [Y/N]: Y
>;     Address checking support will be included
>;     I/O rundown support will be included
>* 15. Loadable device drivers? [Y/N]: Y
>* 16. Should the default for all applicable drivers be loadable? [Y/N]: Y
>;     AST support will be included
>* 18. Include support for Cancel Selective Marktime? [Y/N]: Y
>* 19. Task termination and device not ready messages (TKTN)? [Y/N]: Y
>* 20. Power fail recovery? [Y/N]: Y
>* 22. Large (20K) Executive? [Y/N]: Y
>* 23. Logical device assignment? [Y/N]: Y
>* 25. Do you intend to include a user-written driver? [Y/N]:
>* 30. Executive Debugging Tool (XDT)? [Y/N]: Y
>* 32. Include support for communications products (such as DECnet)? [Y/N]:
>* 33. System controlled partitions? [Y/N]: Y
>* 36. Group global event flags? [Y/N]: Y
>;
>;     For directives:
>;
>;     * - Prints the table
>;     . - Terminates inquiry
>;     % - Gets all valid directives and terminates inquiry
>;
>* 37. Enter code for system directives [S]: *
>;
>;     A- Get partition parameters    B- Get task parameters
>;     C- Send/Receive                D- Alter priority
>;     E- Extend task                 F- Connect to interrupt vector
>;     G- Get sense switch            H- Set system time
>;     I- Stop bit                    J- Specify requested exit AST
>;     K- Parent offspring tasking    L- Parent offspring tasking with chaining
>;
>;     Enter responses as code1code2code3... (no commas) (e.g.: ABCDF.)
>;
>* 37. Enter code for system directives [S]: %
>;     SGN -- WARNING -- CPU does not have switch register
>;         -- Get sense switch directive not included
>* 38. Number of pre-allocated QIO packets [D R:0.-15. D:5.]:
>* 39. Size of data transfer vector in words [D R:0.-33. D:33.]:
>* 40. Round-robin scheduling interval in ticks [D R:0.-1000. D:5.]:
>* 41. Highest priority class for consideration [D R:1.-249. D:150.]:
>* 42. Lowest priority class for consideration [D R:1.-150. D:1.]:
>* 46. Seconds between printer not ready messages [D R:0.-255. D:30.]:
>* 48. System crash register and stack dump device CSR [O R:0-177700 D:177564]:
>* 51. Crash dump analysis: A- PANIC  B- Crash Dump Analysis (CDA) [D:  B] [S]:
>* 52. CDA output notification device CSR [O R:160000-177700 D:177564]:
>* 53. Enter CDA memory dump device mnemonic (ddu:) [S R:3-4]: DL1:
>* 54. Enter CDA memory dump device CSR [O R:160000-177700 D:174400]:
>;
>* EOS *  Do you want to: <CR>-continue R-repeat section E-exit P-pause [S]:
>;
>;
```

```
>; Terminal driver options
>;
>* 1. Terminal driver desired (A/B/C/D, * prints table) [S]: *
>;
>;     Terminal drivers:  A- Tailorable half duplex
>;                        B- Tailored half duplex (user-oriented)
>;                        C- Full duplex
>;                        D- Baseline half duplex
>;
>* 1. Terminal driver desired (A/B/C/D, * prints table) [S]: C
>* 2. Unsolicited input timeout value in seconds [D R:0.-255. D:120.]:
>;
>;     Answer Y(ES) if the following support is desired
>;
>* 6. Unsolicited input character AST? [Y/N]: Y
>* 8. Breakthrough write? [Y/N]: Y
>;    CTRL/R support will be included
>* 10. Escape sequence handling? [Y/N]: Y
>* 11. Get multiple characteristics? [Y/N]: Y
>* 12. Set multiple characteristics? [Y/N]: Y
>* 13. Get terminal driver options? [Y/N]: Y
>* 17. Read after prompt? [Y/N]: Y
>* 20. CRT rubout support? [Y/N]: Y
>* 25. Hardware unrecoverable input error notification? [Y/N]: Y
>* 26. Device independent cursor positioning? [Y/N]: Y
>;
>* EOS * Do you want to: <CR>-continue R-repeat section E-exit P-pause [S]:
>;
>;
>; System Options
>;
>;
>;     Answer Y(ES) if the following support is desired
>;
>* 11. What name would you like to give your system [D: RSX11S] [S R:0-6]:
>;
>* EOS * Do you want to: <CR>-continue R-repeat section E-exit P-pause [S]:
>;
>;     Thinking ...
>;
>; End of Executive option generation at 13:40:10 on 15-MAR-83
>;
>; Peripheral configuration
>;
>;     Parameters appearing in square brackets "[...]" can only be
>;     specified for the first controller of a particular device.
>;     Parameters appearing in parentheses "(...)" only need be specified
>;     if the indicated option is present on the target system.
>;
>;     The  default  for  loadable  drivers  has  been  set  to  *TRUE*.
>;     To override this setting enter R (resident) as the first parameter
>;     for  the  first  controller  of  the  specified  device.
>;
>; A/D and Laboratory devices: None specified
>;
>; Interprocessor communication devices: None specified
>;
>; Unit record devices: LP
>;
>; Enter [L/R,] vector, CSR, printer option
>;            (,[KMC timeout count <0=no KMC>, KMC vector, KMC CSR]) for:
>;
>* 16. LP controller 0 [D: 200,177514,LA180,0,300,160130] [S]: ,,LP11-B
>;
>; Disks: DD, DL
>;
>; Enter [L/R,] vector, CSR, number of drives for:
>;
>* 3. DD controller 0 [D: 300,165500]                [S]: ,,2
>;
>* 3. DL controller 0 [D: 160,174400]                [S]: ,,2
>* 53. Contr 0 unit 0. is an RL01/RL02 [D: RL01]     [S]: RL02
>* 53. Contr 0 unit 1. is an RL01/RL02 [D: RL01]     [S]: RL02
```

```
>;
>; Tapes: None specified
>;
>; Non-physical (pseudo) devices: CO, NL, TI, CL, LB, SY
>;
>; Terminal interface devices: YL, YZ
>;
>; Enter vector, CSR, E=DL11-E modem support/NO=no for:
>;
>* 21. YL controller 0 [D: 60,177560,NO]              [S]:
>;
>; Enter vector, CSR, highest line number <0 to 7> [, answer speed] for:
>;
>* 24. YZ controller 0 [D: ,,,300]                     [S]: 310,160100,3,300
>;
>* EOS * Do you want to: <CR>-continue R-repeat section E-exit P-pause [S]:
>;
>;
>; End of interrupt vector area has been set to 400
>;
>;
>; Create Executive build files
>;
>;
>*  1. Do you wish to edit any of the Executive files? [Y/N]: N
>;
>;
>* EOS * Do you want to: <CR>-continue R-repeat section E-exit P-pause [S]:
>;
>;
>;
>; Prepare for task building
>;
>;
>; Clean-up extraneous object files
>;
>;
>* EOS * Do you want to: <CR>-continue R-repeat section P-pause [S]:
>;
>; End of 11SGEN phase I at 13:53:06 on 15-MAR-83
>;     -- To continue type: @[200,200]11SGEN2
>;
>@ <EOF>
>

>@11SGEN
>;
>; RSX-11S Version 4.1 BL35-A System Generation done on RSX-11M
>;
>; Copyright (C) 1979,1981,1983
>; Digital Equipment Corporation, Maynard, Mass.
>; All Rights Reserved.
>;
>;
>; RL01/RL02 distribution kit
>;
>*  3. Do you want to inhibit execution of MCR commands (PREPGEN)? [Y/N]:
>*  4. Have you made a copy of the distribution kit? [Y/N]: Y
>*  5. Are you generating an unmapped system? [Y/N]:
>*  6. Use an input saved answer file? [Y/N]: Y
>*  7. Name of input saved answer file [D: SYSSAVED.CMD] [S]:
>;
>; Phase I input saved answers
>; created on 15-MAR-83 at 13:16:20
>; by 11SGEN version 2.0
>; are from file DL0:[200,200]SYSSAVED.CMD;1
>;
>*  9. Name of output saved answer file [D: DL0:[200,200]SYSSAVED.CMD] [S]:
>;
>; Phase I output saved answers created in file DL0:[200,200]SYSSAVED.CMD;2
>;
```

```
>* 10.. Use saved host configuration? [Y/N]: Y
>* 11. Use saved peripheral configuration? [Y/N]: Y
>* 12. Skip end of section (EOS) breakpoints? [Y/N]: Y
>* 13. Skip end of execution (EOX) breakpoints? [Y/N]: Y
>* 14. Clean up files from previous GENs? [Y/N]:
>* 16. Enter device for EXCPRV disk when it is ready (ddu:) [D: DL1:] [S]:
>;
>MOU DL1:EXCPRV
>PIP /FR

DL0: has 14667. blocks free, 5813. blocks used out of 20480.
Largest contiguous space = 9578. blocks
1400. file headers are free, 136. headers used out of 1536.

>SET /UIC=[11,10]
>;
>; Target configuration
>;
>; Processor:11/23   Memory Size:128K,Mapped   System:RSX-11S
>;
>;            Floating Point Processor
>;            Parity Memory
>;
>; Host configuration
>;
>; Executive Options
>;
>; Terminal driver options
>;
>; System Options
>;
>;
>;      Thinking ...
>;
>; End of Executive option generation at 13:58:41 on 15-MAR-83
>;
>; Peripheral configuration
>;
>;      Parameters appearing in square brackets "[...]" can  only be
>;      specified  for  the  first controller of a particular device.
>;      Parameters appearing in parentheses "(...)" only need be specified
>;      if  the  indicated  option  is  present  on  the  target  system.
>;
>;      The   default  for  loadable  drivers  has  been  set  to *TRUE*.
>;      To override this setting enter R (resident) as the first parameter
>;      for   the   first   controller   of   the   specified   device.
>;
>; A/D and Laboratory devices: None specified
>;
>; Interprocessor communication devices: None specified
>;
>; Unit record devices: LP
>;
>; Disks: DD, DL
>;
>; Tapes: None specified
>;
>; Non-physical (pseudo) devices: CO, NL, TI, CL, LB, SY
>;
>; Terminal interface devices: YL, YZ
>;
>SET /UIC=[1,24] ! Creating TTDRVBLD.CMD
>SET /UIC=[11,10]
>; End of interrupt vector area has been set to 400
>;
>;
>; Create Executive build files
>;
>PIP RSXMC.MAC=DL1:RSXMC0.MAC/AP
>INS $MAC
>ASN SY:=LB:
>;
>; Start of Executive assembly at 14:04:55 on 15-MAR-83
```

```
>;
>SET /UIC=[11,24]
>MAC @DL1:RSXASM
>;
>; End of Executive assembly at 14:32:15 on 15-MAR-83
>;
>;
>;
>; Start of device drivers assembly at 14:32:16 on 15-MAR-83
>;
>MAC @DL1:RSXDRVASM
>;
>; End of Device drivers assembly at 14:49:56 on 15-MAR-83
>;
>ASN =LB:
>;
>; Prepare for task building
>;
>SET /UIC=[1,24]
>PIP RSX11S.OBS=[11,24]*.OBJ
>PIP TTDRV.OBJ=[11,24]*.TTY
>INS $LBR
>LBR RSX11S/CR:100.:1010.:128.=RSX11S.OBS
>LBR TTDRV/CR:30.:256.:64.=TTDRV
>PIP /NV=[11,10]RSXBLD.CMD
>PIP [11,10]RSXBLD.CMD;*/DE/NM
>REM LBR
>;
>; Clean-up extraneous object files
>;
>PIP RSX11S.OBS;*/DE/NM,TTDRV.OBJ;*,[11,24]*.OBJ;*,*.TTY;*
>REM MAC
>DMO DL1:EXCPRV
DMO -- TTO:    dismounted from DL1:
DMO -- TTO:    dismounted from DL1:    *** Final dismount initiated ***
14:55:58  *** DL1: -- Dismount complete

>;
>; End of 11SGEN phase I at 14:55:59 on 15-MAR-83
>;    -- To continue type: @[200,200]11SGEN2
>;
>SET /UIC=[200,200]
>@ <EOF>
>


>@[200,200]11SGEN2
>;
>; RSX-11S Version 4.1 BL35-A System Generation Phase II
>;
>; Copyright (C) 1975, 1981, 1983
>; Digital Equipment Corp., Maynard, MA.  01754
>; All Rights Reserved.
>;
>;      Build The Executive And All Required Tasks
>;
>; This command file will produce the target system on your SY: disk.
>; If it is an RK05 or an RL01, this command file will delete a number
>; of files from this disk to make room for those required to complete
>; this 11SGEN.  You must be running on an expendable copy of
>; your disk.  Note however, that the disk will be left in a
>; state such that RSX-11S sysgens may be performed at a later
>; date.  In addition, the sysgen itself may be terminated and
>; may be restarted at just about any point to rebuild particular
>; pieces of the system.
>;
>* In what UIC is SGNPARM.CMD if not [200,200] [S]:
>; RL01/RL02 distribution kit 11SGEN version 2.0 for RSX-11M BL35-A
>;
>; Continuation from 11SGEN PHASE I done on 15-MAR-83 at 14:55:48
>TIM
15:00:37 15-MAR-83
>;
```

```
>; The map device specified must already be mounted with UIC [1,34].
>; If it is not, respond to the next question with ^Z and do so.
>;
>* In which device is the EXCPRV disk [DDU:] [S]: DL1:
>MOU DL1:EXCPRV
>ASN SY:=LB:
>;
>; Expanded comments provide a complete description of every step in
>; this 11SGEN command file.  On the other hand, short comments
>; provide virtually no explanatory text.
>;
>* Do you want expanded comments? [Y/N]: Y
>;
>; Make room on the baseline disk to complete the 11SGEN
>;
>SET /UIC=[1,1]
>PIP [1,34]*.*/PU/NM
>PIP [1,54]CTDRV.TSK;*/DE/NM,DBDRV.TSK;*,DPDRV.TSK;*,DRDRV.TSK;*
>PIP [1,54]CTDRV.STB;*/DE/NM,DBDRV.STB;*,DPDRV.STB;*,DRDRV.STB;*
>PIP [200,200]SYSGEN2.CMD;*/DE/NM
>PIP /FR


DL0: has 14549. blocks free, 5931. blocks used out of 20480.
Largest contiguous space = 9578. blocks
1401. file headers are free, 135. headers used out of 1536.


>;
>; Create RSX-11S subset system library
>; if you have not already done so.
>;
>INS SY:$LBR
>* Have you already created 11SLIB.OLB? [Y/N]:
>LBR LB:11SLIB/CR:30.:896.:320.
>REM LBR
>* Have you already built the Exec? [Y/N]:
>SET /UIC=[1,24]
>;
>; Purge Exec build command files and assembly prefix files
>;
>PIP RSXBLD.CMD/PU/NM
>PIP [11,10]RSXMC.MAC/PU/NM
>;
>; Now we build the Executive
>; If you must make changes to the Executive build files
>; answer yes to the following question. Edit RSXBLD.CMD
>; by issuing the command EDI LB:RSXBLD.CMD. 11SGEN2 will
>; build the Executive after you resume from the pause.
>;
>* Do you wish to edit the Executive build file? [Y/N]:
>INS SY:$TKB
>TKB @LB:RSXBLD
>* Did the Executive build successfully? [Y/N]: Y
>;
>; You now need the Executive maps to layout
>; the partitions in your new system.
>;
>* Do you have the maps already? [Y/N]:
>LOA LP:
>PIP LP:=DL1:[1,34]RSX11S.MAP,EXCOM1.MAP,EXCOM2.MAP
>UNL LP:
>;
>; Look in the map of the Exec for the symbol "$SYTOP" which
>; indicates the last address allocated to the Exec.  User
>; partitions may start at any location beyond this address
>; as long as they fall on a 64 byte boundary. See the RSX-11M
>; Sysgen manual for a description of how to set up partitions.
>; You may wish to stop at this time to layout your partitions.
>;
>SET /UIC=[1,64]
>* Do you want to build any loadable drivers now? [Y/N]: Y
>;
>; Build Loadable drivers
>;
```

```
>*  3. Device for driver's maps [D: NL:] [S]:
>SET /UIC=[1,24]
>;
>; The following are mnemonics of the loadable device drivers selected.
>;
>;      DD      DL      LP
>;
>; When all drivers are built, strike carriage return.
>;
>*  5. Driver 2-character device mnemonic [S]: DD
>*  6. Drive partition name [D: DRVPAR] [S]:
>TKB @DL1:DDDRVBLD
>*  5. Driver 2-character device mnemonic [S]: DL
>*  6. Drive partition name [D: DRVPAR] [S]:
>TKB @DL1:DLDRVBLD
>*  5. Driver 2-character device mnemonic [S]: LP
>*  6. Drive partition name [D: DRVPAR] [S]:
>TKB @DL1:LPDRVBLD
>*  5. Driver 2-character device mnemonic [S]:
>SET /UIC=[1,64]
>REM TKB
>SET /UIC=[1,24]
>PIP RSX11S.OBJ;*/DE/NM
>* Have you already built the full duplex terminal driver? [Y/N]:
>;
>; Now we build the full duplex terminial driver
>;
>;
>; The  task build command file for the full duplex terminal drive
>; specifies that it run in partition TTPAR, having a length of
>; 40000 octal.
>;
>* Do you wish to edit the build command file for the terminal driver? [Y/N]:
>INS SY:$TKB
>TKB @LB:TTDRVBLD.CMD
>* Did the terminal driver build successfully? [Y/N]: Y
>REM TKB
>* List full duplex terminal driver map? [Y/N]: Y
>LOA LP:
>PIP LP:=DL1:[1,34]TTDRV.MAP
>UNL LP:
>* Do you want to continue and build the privileged tasks? [Y/N]: Y
>DMO DL1:
DMO -- TT0:     dismounted from DL1:     *** Final dismount initiated ***
15:15:28  *** DL1: -- Dismount complete

>;
>; Prepare to assemble and build the RSX-11S privileged
>; software components.
>;
>; The tasks TKTN, BASMCR, OTL, and SIP must be assembled, and the
>; task build command files possibly edited before they can be built.
>; The tasks RSDV05, RSDV52, and RSDV1H need only to be built, but
>; their command files may also be edited before they are built.
>; It is not necessary to save/print the assembly listings;
>; However, we recommend that you save/print the task build
>; maps.  This is especially important when generating
>; an unmapped system -- without maps it will be difficult
>; to determine partition limits.
>;
>* Assembly list device for privileged tasks (DDNN:) [D: NL:] [S]:
>* Map device for privileged tasks (DDNN:) [D: SY:] [S]:
>* Have you already built task TKTN? [Y/N]:
>;
>; Task TKTN will be included in your 11S system as a result
>; of your selection in 11SGEN Phase I.
>;
>;
>; The privileged tasks BASMCR, OTL, and SIP were specially designed
>; for use in the RSX-11S environment.  Their inclusion in the system
>; is optional.  If desired, they will be assembled and built under
>; control of this command file.
```

```
>;
>* Do you wish to include BASMCR, OTL, or SIP in your system? [Y/N]: Y
>;
>; The tasks RSDV05, RSDV52, and RSDV1H dynamically display the
>; the status of memory on a graphics terminal and can be built
>; for the VT05B terminal, VT52 terminal, and VT100 terminal
>; respectively.  These tasks display which tasks are in memory
>; and where in memory they reside, which task is currently
>; running, and the status of pool.
>;
>* Do you want to include RSDV05, RSDV52, or RSDV1H? [Y/N]: Y
>;
>; The RSX-11S subset file control services (FCS) are dist-
>; ributed as a single file containing the concatenated object
>; modules.  They cannot be placed in the RSX-11M system object
>; library (SYSLIB.OLB).  However, users who require this FCS
>; should link with the RSX-11S subset of SYSLIB (11SLIB.OLB)
>; created during this Sysgen.
>;
>* Do you wish to include the RSX-11S subset FCS? [Y/N]: Y
>;
>; If you have previously done an RSX-11S Sysgen on this disk, all
>; of the files were transferred to the UFD [2,20].  These were
>; optionally deleted by the Sysgen process.  Therefore, unless
>; it was specified to delete the files or they were later
>; deleted independently, there is no need to copy all of the
>; files from the release kit again.
>;
>* Do you have to copy the files from the release kit? [Y/N]: Y
>* Enter device and unit of RSX-11S distribution kit (DDNN:) [S]: DL1:
>;
>; Place the RSX-11S release kit cartridge disk in drive DL1:
>;
>* Is the disk ready in drive DL1:? [Y/N]: Y
>SET /UIC=[2,20]
>INS SY:$UFD
>UFD SY:[2,20]
>REM UFD
>PIP *.*;*/DE/NM
>MOU DL1:11SKIT1
>PIP SY:=DL1:
>DMO DL1:
DMO -- TTO:    dismounted from DL1:    *** Final dismount initiated ***
15:20:46  *** DL1: -- Dismount complete

>SET /UIC=[11,10]
>INS SY:$UFD
>UFD SY:[11,10]
UFD -- Directory already exists
>REM UFD
>* Do you wish to make corrections to RSX-11S source modules? [Y/N]:
>INS SY:$MAC
>SET /UIC=[1,24]
>MAC @TKNASM
>PIP TKNASM.CMD;*/DE
>REM MAC
>;
>; The task build command file for TKTN specifies that it run in
>; partition SYSPAR, having a length of 3700 octal.
>; If you must change the partition name or length,
>; you will have to edit its task build command file.
>;
>* Do you have to edit the task build command file for TKTN? [Y/N]:
>INS SY:$TKB
>TKB @TKN11SBLD.CMD
>REM TKB
>;
>; BASMCR is a privileged task which supplies a subset (or basic
>; set) of the RSX-11M MCR commands. It can not be included if
>; during Sysgen Phase I you indicated that you did not want any
>; terminal devices.  Its size, depending upon options, ranges
>; from 6400 to 14000 octal on mapped systems.
>;
```

```
>* Do you want to include task BASMCR? [Y/N]: Y
>INS SY:$UFD
>UFD SY:[16,10]
>UFD SY:[16,24]
>REM UFD
>SET /UIC=[16,10]
>PIP BMPRE.MAC;*/DE/NM
>PIP *.MAC;*/RM/NM
>PIP [16,*]*.*;*/DE/NM
>PIP [16,10]=[2,20]BMPRE.MAC
>PIP [16,10]/EN=[2,20]ABOBAS.MAC,ATLBAS,BRKBAS,ERRBAS,ERRMSG
>PIP [16,10]/EN=[2,20]FMTDV.MAC,FNDUCB,GNBLK,GETNUM,KEYWD
>PIP [16,10]/EN=[2,20]LKLST.MAC,MCRBAS,OPEBAS,PRSBAS,REDBAS
>PIP [16,10]/EN=[2,20]REMBAS.MAC,RPSBAS,RUNBAS,TIMBAS
>SET /UIC=[16,24]
>;
>; The size of BASIC MCR is dependent upon the inclusion of
>; conditional assembly code.  The inclusion of the code is
>; controlled by symbol definitions appearing in an assembly
>; prefix file.  The default for these conditionals is
>; exclusion.  The following options, along with the corres-
>; ponding control symbols, are available:
>;
>;       R$$ATL - ATL/TAL COMMANDS
>;       R$$ERM - ERROR MESSAGE TEXT
>;       R$$OPE - OPEN COMMAND
>;
>; To include an option locate the desired conditional symbol
>; and remove the semicolon at the beginning of the line.  For
>; example:
>;
>;       *PL ;R$$ATL
>;       ;R$$ATL = 0              ;INCLUDE ATL
>;       *C /;//
>;       R$$ATL = 0              ;INCLUDE ATL
>;       *
>;
>; Repeat this for all desired options.  Note that the symbols
>; appear on consecutive lines in alphabetical order in the file.
>;
>* Do you wish to edit the assembly prefix file for BASMCR? [Y/N]: Y
>INS SY:$EDI
>EDI [16,10]BMPRE.MAC
[00029 lines read in]
[Page    1]
*PL ;R$$ATL
;R$$ATL =        0              ; INCLUDE ATL
*C /;//
R$$ATL  =        0              ; INCLUDE ATL
*PL ;R$$ERM
;R$$ERM =        0              ; INCLUDE ERROR MESSAGES
*C /;//
R$$ERM  =        0              ; INCLUDE ERROR MESSAGES
*PL ;R$$OPE
;R$$OPE =        0              ; INCLUDE OPEN
*C /;//
R$$OPE  =        0              ; INCLUDE OPEN
*EX
[Exit]

>REM EDI
>INS SY:$MAC
>MAC @BASMCRASM.CMD
>REM MAC
>SET /UIC=[1,24]
>PIP [16,10]BMPRE.MAC;*/DE/NM
>PIP [16,10]*.MAC;*/RM/NM
>PIP BASMCR.OBJ=[16,24]*.OBJ
>PIP [16,24]*.*;*/DE/NM
>INS SY:$LBR
>LBR BASMCR/CR:21.:64.:64.=BASMCR
>REM LBR
>PIP BASMCR.OBJ;*/DE/NM
```

```
>;
>; The task build command file for BASMCR specifies that it
>; run in partition MCRPAR, length 14000 octal (mapped).
>; If you must change the partition name or length,
>; you will have to edit its task build command file.
>;
>* Do you wish to edit the task build command file for BASMCR? [Y/N]:
>INS SY:$TKB
>TKB @BASMCRBLD.CMD
>REM TKB
>;
>; OTL is a privileged task which is the RSX-11S on-line task loader.
>; It performs the install, load, and fix functions for tasks which
>; were created on a host development system and copied via FLX to a
>; load medium.  Depending on the number of load devices, etc, OTL
>; ranges from 8000 to 15000 octal on mapped systems.
>;
>* Do you want to include task OTL? [Y/N]: Y
>INS SY:$UFD
>UFD SY:[20,10]
>UFD SY:[20,24]
>REM UFD
>SET /UIC=[20,10]
>PIP SY:OTLPRE.MAC;*/DE/NM
>PIP SY:*.MAC;*/RM/NM
>PIP [20,*]*.*;*/DE/NM
>PIP [20,10]=[2,20]OTLPRE.MAC
>PIP [20,10]/EN=[2,20]OTL.MAC,OTLCD,OTLCM,OTLERR
>PIP [20,10]/EN=[2,20]OTLHD.MAC,OTLLB,OTLLD,OTLQIO,OTLRD
>SET /UIC=[20,24]
>;
>; The size of OTL is dependent upon the inclusion of conditional
>; assembly code.  The inclusion of the code is done in two ways,
>; by a single symbol definition and by macro calls appearing
>; in an assembly prefix file.  The first controls the inclusion
>; of error message text. To include this option, locate the
>; symbol R$$ERM, and remove the semicolon from the beginning
>; of the line.  The default for error message text is exclusion.
>; For example:
>;
>;      *PL ;R$$ERM
>;      ;R$$ERM=0
>;      *C /;//
>;      R$$ERM=0
>;
>; The other type of option concerns the exclusion of unwanted
>; device support.  For each of the OTL load devices which
>; were included in Sysgen Phase I, corresponding support will be
>; generated in OTL.  Naturally, support will not be generated for
>; those OTL load devices which were not included in Sysgen Phase I.
>; The following devices along with corresponding control symbols
>; may be supported by OTL:
>;
>;      PR11 - PAPER TAPE READER ------ PR
>;      RX11 - RX11 FLEXIBLE DISK ----- DX
>;      RX21 - RX211 FLEXIBLE DISK ---- DY
>;      TA11 - CASSETTE TAPE ---------- CT
>;      TC11 - DECTAPE ---------------- DT
>;      TM11 - TM/TMA/TMB11 MAGTAPE --- MT
>;      TS11 - TS04 MAGTAPE ----------- MS
>;      TU16 - TJ/TWU16-45 MAGTAPE ---- MM
>;      TU58 - TU58 CARTRIDGE --------- DD
>;
>; To exclude OTL support for a device which has been generated into
>; your RSX-11S Executive, locate the corresponding device symbol
>; and place a semicolon at the beginning of the line.  For example:
>;
>;      *PL PR11
>;              INCLUD  PR11,P$$R11     ;PC11/PR11 SUPPORT
>;      *C //;/
>;      ;       INCLUD  PR11,P$$R11     ;PC11/PR11 SUPPPRT
>;      *
>;
```

3-18

```
>; Perform this for all of the above devices which will be part of
>; your RSX-11S Executive, but for which you do not want OTL
>; support.  Note that the symbols appear on consecutive lines in
>; alphabetical order in the file.
>;
>* Do you wish to edit the assembly prefix file for OTL? [Y/N]: Y
>INS SY:$EDI
>EDI [20,10]OTLPRE.MAC
[00036 lines read in]
[Page    0]
*PL ;R$$ERM
[00036 lines read in]
;R$$ERM=0
*C /;//
R$$ERM=0
*EX
[Exit]


>REM EDI
>INS SY:$MAC
>MAC @OTLASM.CMD
>REM MAC
>SET /UIC=[1,24]
>PIP [20,10]OTLPRE.MAC;*/DE/NM
>PIP [20,10]*.MAC;*/RM/NM
>PIP OTL.OBJ=[20,24]*.OBJ
>PIP [20,24]*.*;*/DE/NM
>INS SY:$LBR
>LBR OTL/CR:21.:128.:64.=OTL
>LBR OTL=[2,20]BLXIOM
>REM LBR
>PIP OTL.OBJ;*/DE/NM
>;
>; The task build command file for OTL specifies that it
>; run in partition OTLPAR, length 15000 octal (mapped).
>; If you must change the partition name or length,
>; you will have to edit its task build command file.
>;
>* Do you wish to edit the task build command file for OTL? [Y/N]:
>INS SY:$TKB
>TKB @OTLBLD.CMD
>REM TKB
>;
>; SIP is a privileged task which performs the RSX-11S system
>; image preservation duties.  It replaces the RSX-11M SAV MCR
>; command.  Depending on the number of load devices, etc, SIP
>; ranges from 3700 to 7400 octal on mapped systems.
>;
>* Do you want to include task SIP? [Y/N]: Y
>INS SY:$UFD
>UFD SY:[21,10]
>UFD SY:[21,24]
>REM UFD
>SET /UIC=[21,10]
>PIP SY:SIPPRE.MAC;*/DE/NM
>PIP SY:*.MAC;*/RM/NM
>PIP [21,*]*.*;*/DE/NM
>PIP [21,10]=[2,20]SIPPRE.MAC
>PIP [21,10]/EN=[2,20]SIP.MAC,SIPBOO,SIPCD,SIPERR
>PIP [21,10]/EN=[2,20]SIPLD.MAC,SIPQIO
>SET /UIC=[21,24]
>;
>; The size of SIP is dependent upon conditional assembly code
>; in exactly the same manner as OTL.  There are only a few minor
>; differences.  OTL supports the paper tape reader using the
>; control symbol PR11;  SIP on the other hand supports the paper
>; tape punch using the control symbol PP11. In addition, OTL
>; supports DECtape II cartridges using control symbol TU58 and
>; TS11 magnetic tape drives using the control symbol TS11.
>; Edit SIPPRE.MAC in the same way described for OTL.  For example:
>;
```

```
>;        *PL PP11
>;                 INCLUD   PP11,P$$P11        ;PC11/PR11 SUPPORT
>;        *C //;/
>;        ;        INCLUD   PP11,P$$P11        ;PC11/PR11 SUPPPRT
>;        *
>;
>* Do you wish to edit the assembly prefix file for SIP? [Y/N]: Y
>INS SY:$EDI
>EDI [21,10]SIPPRE.MAC
[00036 lines read in]
[Page    0]
*PL ;R$$ERM
[00036 lines read in]
;R$$ERM=0
*C /;//
R$$ERM=0
*EX
[Exit]

>REM EDI
>INS SY:$MAC
>MAC @SIPASM.CMD
>REM MAC
>SET /UIC=[1,24]
>PIP [21,10]SIPPRE.MAC;*/DE/NM
>PIP [21,10]*.MAC;*/RM/NM
>PIP SIP.OBJ=[21,24]*.OBJ
>PIP [21,24]*.*;*/DE/NM
>INS SY:$LBR
>LBR SIP/CR:20.:64.:64.=SIP
>REM LBR
>PIP SIP.OBJ;*/DE/NM
>;
>; The task build command file for SIP specifies that it
>; run in partition SIPPAR, length 7400 octal (mapped).
>; If you must change the partition name or length,
>; you will have to edit its task build command file.
>;
>* Do you wish to edit the task build command file for SIP? [Y/N]:
>INS SY:$TKB
>TKB @SIPBLD.CMD
>REM TKB
>SET /UIC=[1,24]
>PIP RMD.OLB/NV=[2,20]RMDMAP.OLB
>* Do you want to include task RSDV05? [Y/N]:
>* Do you want to include task RSDV52? [Y/N]:
>* Do you want to include task RSDV1H? [Y/N]: Y
>;
>; The task build command file for RSDV1H specifies that it
>; run in partition RSDV1H, length 40000 (octal).
>; If you must change the partition name or length, you
>; will have to edit the task build command file.
>;
>* Do you wish to edit the task build command file for RSDV1H? [Y/N]:
>INS SY:$TKB
>TKB @RSDV1HBLD.CMD
TKB -- *DIAG*-Symbol HEADER not found for patch


>REM TKB
>;
>; Add subset FCS to the RSX-11S system object library
>;
>SET /UIC=[1,1]
>INS SY:$LBR
>LBR LB:11SLIB/RP=[2,20]FCS11S
>REM LBR
>PIP LB:11SLIB.OLB/PU/NM
>SET /UIC=[1,64]
>PIP *.*/PU/NM
>PIP [1,24]*.*/PU/NM
>;
```

```
>; The operation left to be performed is that of the final
>; configuration of the virgin executive using Virtual MCR.
>;
>* Do you want to configure the Exec at this time using VMR? [Y/N]: Y
>;
>; The contents of your RSX-11S release kit currently resides
>; under the UFD [2,20]. If your SY: disk is not an RK06/07 or an
>; RP04/05/06 you may want to delete these files. Leaving
>; them on the disk will save the time of re-copying them the next
>; time you perform an RSX-11S Sysgen. If your SY: disk is an RK05,
>; they will be automatically deleted.
>;
>* Do you want to delete the RSX-11S kit from this disk? [Y/N]:
>;
>; We will now make a copy of your target system
>;
>PIP RSX11S.SYS/NV/CO/BL:498.=RSX11S.TSK
>;
>; If you wish to make a copy of your task or system image file
>; on another device we will pause now to allow you to do so.
>;
>* Do you wish to copy your system to another device? [Y/N]:
>;
>; RSX11S.TSK is a backup copy of the 11S system you have just built.
>; If you do not wish to retain it, we will delete it now.
>;
>* Do you wish to delete the system backup file RSX11S.TSK? [Y/N]:
>;
>; Your target system is now ready to set up partitions and install
>; the required tasks. The procedure to follow is:
>;        1) Run Virtual MCR
>;        2) Extend POOL space to base of first partition
>;        3) Set up your partitions
>;        3B) Load needed drivers
>;        3C) Set up a partition for the Directive Common
>;        4) Install tasks (BASMCR, OTL, SIP, TKTN)
>;        5) Fix tasks into (virtual) memory
>;        6) Optionally set the time and save a copy of THE
>;            system on a bootable medium
>;        7) Exit from virtual MCR and boot in your target system
>;
>; VMR example:
>;
>;        >VMR                      ! RUN VIRTUAL MCR
>;        ENTER FILENAME:RSX11S.SYS  ! VMR PROMPTS FOR FILE NAME
>;        VMR>SET /POOL=XXX          ! EXTEND POOL TO FIRST PARTITION BASE
>;        VMR>SET /MAIN=TTPAR:XXX:XXX:TASK  !SET UP FULL DUPLEX TT PARTITION
>;        VMR>SET /MAIN=DRVPAR:XXX:XXX:TASK  ! SET UP DRIVER PARTITION
>;        VMR>SET /MAIN=EXCOM1:XXX:XXX:COM  ! SET 1ST EXEC. COMMON
>;        VMR>INS EXCOM1             !INSTALL 1ST EXECUTIVE COMMON
>;        VMR>SET /MAIN=EXCOM2:XXX:XXX:COM  ! SET 2ND EXEC. COMMON
>;        VMR>INS EXCOM2             !INSTALL 2ND EXECUTIVE COMMON
>;        VMR>SET /MAIN=SYSPAR:XXX:XXX:TASK  ! SET UP TKTN PARTITION
>;        VMR>SET /MAIN=MCRPAR:XXX:XXX:TASK  ! SET UP MCR PARTITION
>;        VMR>SET /MAIN=OTLPAR:XXX:XXX:TASK  ! SET UP OTL PARTITION
>;        VMR>SET /MAIN=SIPPAR:XXX:XXX:TASK  ! SET UP SIP PARTITION
>;        VMR>SET /MAIN=RSDV1H:XXX:XXX:TASK       !SET UP RSDV1H PAR
>;        VMR>SET /MAIN=GEN:XXX:XXX:SYS  ! IF SYSTEM CONTROLLED PAR.
>;        VMR>SET /MAIN=GEN:XXX:XXX:TASK ! IF USER CONTROLLED PARTITION
>;        VMR>LOA XX:                ! LOAD XXDRV
>;        VMR>INS/FIX TKN            ! INSTALL & FIX TKTN
>;        VMR>INS/FIX BASMCR         ! INSTALL & FIX BASIC MCR
>;        VMR>INS/FIX OTL            ! INSTALL & FIX OTL
>;        VMR>INS/FIX SIP            ! INSTALL & FIX SIP
>;        VMR>INS/FIX RSDV1H         ! INSTALL & FIX RSDV1H
>;        VMR>TIM 10:45 11/10/77     ! SET TIME AND DATE
>;        VMR>SAV DX:RSX11S          ! SAVE SYSTEM ON RX01 FLOPPY
>;        VMR>^Z                     ! EXIT FROM VIRTUAL MCR
>;
```

```
>; End of VMR example
>;
>;
>;
>; Now we will run Virtual MCR so that you may initialize
>; your system.  Be sure to define all required partitions
>; at this time, because the SET function is not supported
>; in RSX-11S BASIC MCR.  Furthermore, if you plan to
>; include BASIC MCR and OTL in your system, they must
>; be installed and fixed into memory.  Other tasks,
>; such as SIP and TKTN may be loaded into a running
>; RSX-11S system using OTL.
>;
>;        NOTE: Since LOAD and UNLOAD are not supported on
>;        a running RSX-11S system, all required loadable
>;        drivers must be loaded using VMR.
>;
>;        NOTE: You have requested to include the
>;        directive common.  You must install the
>;        common prior to exitting Virtual MCR
>;        or your RSX-11S system will not boot.
>INS SY:$VMR
>VMR
Enter filename: RSX11S
VMR>SET /POOL=*
VMR>SET /MAIN=TTPAR:*:400:TASK
VMR>LOA TT:
VMR -- *DIAG*-Loadable driver larger than 4K
VMR>SET /MAIN=DRVPAR:*:*:SYS
VMR>LOA DD:
VMR>LOA DL:
VMR>LOA LP:
VMR>SET /TOP=DRVPAR:-*
VMR -- *DIAG*-Installed tasks may no longer fit in partition
VMR>SET /MAIN=EXCOM1:*:200:COM
VMR>INS EXCOM1
VMR -- *DIAG*-Partition reduced to executive common size
VMR>SET /MAIN=EXCOM2:*:200:COM
VMR>INS EXCOM2
VMR -- *DIAG*-Partition reduced to executive common size
VMR>SET /MAIN=SYSPAR:*:101:TASK
VMR>SET /MAIN=MCRPAR:*:137:TASK
VMR>SET /MAIN=OTLPAR:*:150:TASK
VMR>SET /MAIN=SIPPAR:*:74:TASK
VMR>SET /MAIN=RSDV1H:*:400:TASK
VMR>SET /MAIN=GEN:*:*:SYS
VMR>INS TKN/FIX=YES
VMR>INS BASMCR/FIX=YES
VMR>INS OTL/FIX=YES
VMR>INS SIP/FIX=YES
VMR>INS RSDV1H/FIX=YES
VMR>REA RSDV1H 1 TT1:
VMR>REA RSDV1H 2 TT1:
VMR>TIM 16:29 15-MAR-83
VMR>SAV DD:11S
VMR>^Z

>;
>; When this command file completes, your RSX-11S system is
>; ready to run.
>;
>REM VMR
>SET /UIC=[200,200]
>TIM
16:28:00 15-MAR-83
>;
>@ <EOF>
>
```

```
TESTING MEMORY
0128.KW
START? DD

XDT: 35-A

XDT>G
RSX11S V4.1 BL35-A


>
 TIM
16:30:16 15-MAR-83
>
```

3.5 SYSGEN ON A RSX-11M-PLUS HOST SYSTEM

This section contains information on preparing for your RSX-11S SYSGEN on an RSX-11M-PLUS host.

For the most part, the actual dialogue and procedures for the RSX-11S SYSGEN are the same as those for an RSX-11M host. The procedure for performing an RSX-11S SYSGEN on an RSX-11M-PLUS host is the same as is currently documented. The only differences are in the numbering and wording of some of the questions. These changes are reflected in the RSX-11S SYSGEN on a RSX-11M host.

3.6 SYSGEN ON A VAX/VMS HOST SYSTEM

This section contains information on preparing for your RSX-11S SYSGEN on a VAX/VMS host.

For the most part, the actual dialogue and procedure for performing an RSX-11S SYSGEN on a VAX/VMS host is the same as is currently documented. The only difference is in the numbering and wording of some of the questions. These changes are reflected in the RSX-11S SYSGEN on a RSX-11M host.

Also, if you have disk quota checking on your SYS$LOGIN device (your work disk), it must be disabled before you begin your RSX-11S system generation. This is because the compatibility mode task, MAC, needs to create temporary work files in UICs that probably do not have an established disk quota on SYS$LOGIN. Otherwise, MAC cannot assemble any files during 11SGEN.

To disable disk quota checking, invoke the Disk Quota Utility (DISKQUOTA) by issuing the following command line:

    $ RUN SYS$SYSTEM:DISKQUOTA

When the DISKQUOTA prompt appears on your terminal, issue the following command lines:

    DISKQ>USE diskname:(RET)
    DISKQ>DISABLE(RET)
    DISKQ> (CTRL/Z)
    $

where diskname: is the name of the SYS$LOGIN device.

Now disk quota checking is disabled and you are ready to begin your RSX-11S SYSGEN.

READER'S COMMENTS

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

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

_____
_____
_____
_____
_____
_____
_____
_____
_____
_____

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

_____
_____
_____
_____
_____
_____
_____
_____
_____
_____
_____
_____

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

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

Name _____ Date _____

Organization _____

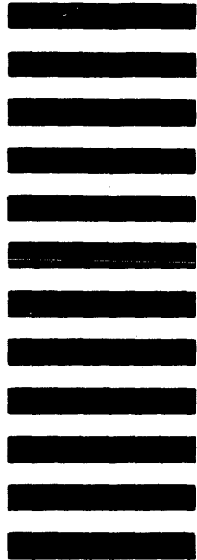Street _____

City _____ State _____ Zip Code _____

digital

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

POSTAGE WILL BE PAID BY ADDRESSEE

BSSG PUBLICATIONS ZK1-3/J35
DIGITAL EQUIPMENT CORPORATION
110 SPIT BROOK ROAD
NASHUA, NEW HAMPSHIRE 03061