Micro/RSX Base Kit Release Notes

Order No. AA-FY05A-TC

Micro/RSX Version 3.0

Contents

Chapter 2   Corrections to Documentation

Chapter 3   RMS–11 Version 2.0 Release Notes

Appendix A The DIGITAL Equipment Computer User's Society (DECUS)

Index

Tables

Preface

Manual Objectives

This manual contains important information for using the Micro/RSX Version 3.0 operating system. Read this document before installing and using your system.

The *Micro/RSX Release Notes* contain the following categories of information:

*   Descriptions of new software features and changes in existing software features

*   Descriptions of software restrictions and incompatibilities with previous versions

*   Supplementary information that has not yet been incorporated into the appropriate manuals

*   Corrections to documentation errors and references to new locations of information

Intended Audience

This manual is intended for all users of the Micro/RSX operating system.

Structure of This Manual

The following is a chapter-by-chapter description of the release notes.

Chapter 1—Micro/RSX Software Features and Restrictions

This chapter describes the new features and new device support for this release. Each section contains a complete description of new software features, restrictions, and supplementary release note information for a system component. If a particular system component has no new features, restrictions, or supplementary information, then a section describing that information is not included.

Chapter 2—Corrections to Micro/RSX Documentation

This chapter contains corrections to errors in your Micro/RSX Base Kit documentation set. Section titles in this chapter reflect manual titles.

Chapter 3—RMS-11 Release Notes

This chapter describes new features and restrictions for RMS-11 Version 2.0. Although the RMS-11 version number has not changed since the last release of Micro/RSX, new RMS-11 features have been added.

Appendix A—DIGITAL Equipment Computer User's Society (DECUS)

Appendix A is an introduction to the DIGITAL Equipment Computer User's Society (DECUS). The goals and activities of DECUS are described, and information about becoming a DECUS member is provided.

Conventions Used in This Manual

Convention	Meaning
red ink	In examples, what you type is printed in red. What the system types is printed in black.
RET	Any symbol with a 2- to 6-character abbreviation indicates that you press the corresponding key on your terminal. For example, RET indicates that you press the RETURN key.
CTRL/a CTRL/a	The symbol CTRL/a or CTRL/a means that you hold down the key labeled CTRL while pressing another key. For example, CTRL/Z indicates that you should press the CTRL key and the Z key together.

Chapter 1
Micro/RSX Software Features and Restrictions

The first two sections of this chapter provide a summary of the major new features and new device support for Micro/RSX Version 3.0. The remaining sections describe new software features, restrictions, and supplementary information for the various components of Micro/RSX.

1.1 Summary of New Features

Micro/RSX Version 3.0 supports the following new features:

- Logical names
- Vectored Executive
- Disk data caching
- Separate instruction and data space
- Supervisor mode
- User-defined default protection
- Password encryption
- LAT terminal server support
- New DCL commands and qualifiers

1.1.1 Increased Memory Requirement

There is an increased memory requirement for this release of Micro/RSX. Micro/RSX Version 3.0 requires a minimum of 512 Kb of memory.

1.2 New Device Support

This section describes new device support and device restrictions on Micro/RSX Version 3.0. It includes the following topics:

- Processor support

- Controller support

- Modem support

- Terminal and printer support

- Disk support

- Magnetic tape support

- Device support restrictions

- IP11 and IPV11 now a Layered Product

1.2.1 Processor Support

Micro/RSX Version 3.0 includes the following new processor support:

- J11-series

  This series consists of the LSI-11/73, MicroPDP–11/73, MicroPDP–11/83, and PDP–11/84.

Micro/RSX Version 3.0 supports the LSI-11/73 (KDJ11-A) processor module only in the following configuration:

  KDJ11-A (LSI-11/73) CPU module
  MXV11-BF multifunction module with MXV11-B2 ROM set

This configuration should be used when the LSI-11/73 CPU is to be used with the RQDX1 or RQDX2 disk controllers. The memory on the MXV11-BF multifunction module should be set to start above the MXV11 resident memory. Note that in this configuration the system Executive is running in memory which is not protected by parity checking.

Neither the RQDX1 nor RQDX2 controllers for the RD51/52/53 disks and RX50 diskettes will work with the DLVJ1 (formerly known as the DLV11-J) multi-line terminal controller. Configurations using the LSI-11/73 and DLVJ1 are not supported by Micro/RSX.

1.2.2 Controller Support

Micro/RSX Version 3.0 includes the following controller support:

- RQDX2 Controller
- RQDX3 Controller
- KDA50-Q Controller

The RQDX3 controller is a dual height Q-BUS controller module with two data channels designed to replace the RQDX1 and RQDX2 quad height disk controllers. The RQDX3 controller supports the RX50 diskette drive and the RD51, RD52, and RD53 disk drives.

The KDA50-Q controller is the Q-BUS version of the UDA50 controller. It supports the RA80, RA81, and RA60 disks.

1.2.3 Modem Support

The Data Terminal Emulator on Micro/RSX Version 3.0 supports the following modems:

- DF03
- DF112
- DF224

1.2.4 Terminal and Printer Support

Micro/RSX Version 3.0 includes support for the following terminals and printers:

- DTC01
- LA210
- LN03
- LQP02
- LQP03
- PRO-series

1.2.5 Disk Support

Micro/RSX Version 3.0 includes support for the following disks:

- RA60
- RA80
- RA81
- RC25
- RD52
- RD53

The RD52 and RD53 disks are described in the next two sections.

1.2.5.1 RD52 Fixed Disk

The RD52 disk is a 5.25-inch fixed disk. It is soft sectored and field formattable. The maximum capacity of the RD52 is 30.97 megabytes.

1.2.5.2 RD53 Fixed Disk

The RD53 5.25-inch fixed disk has 2.3 times the capacity and 33 percent faster average access time of the RD52, making it possible to support larger applications. The RD53 requires the RQDX2 or the RQDX3 controller. It has a capacity of 71 megabytes.

1.2.6 Magnetic Tape Support

Micro/RSX Version 3.0 includes support for the following magnetic tapes:

- TK25
- TK50

The following sections describe these tapes.

1.2.6.1 TK25 Subsystem

Micro/RSX Version 3.0 supports the TK25 tape drive. The TK25 consists of a TK25 controller for the Q-BUS and a TK25 streaming tape drive. The integrated subsystem consists of a tape drive and controller/formatter. The TK25 uses a DC600A-1/4 inch tape cartridge and stores data on serial data tracks in a serial serpentine recording method. It has a storage capacity of 60 megabytes for 8K-byte data records. Data recording is an 8000-bpi, modified GCR (group cyclical recording) method.

1.2.6.2 TK50 Tape Subsystem

Micro/RSX Version 3.0 supports the new TK50 tape drive. The TK50 (MU:)
is an integrated subsystem that consists of a controller for the Q-BUS and
a TK50 streaming tape drive. The controller handles all error recovery and
correction, and internally buffers multiple outstanding commands. The
TK50 driver (MUDRV) reads and writes data on a 1/2-inch tape cartridge
that is recorded at 6667 bpi on serial data tracks in a serial serpentine
recording (modified frequency modulation) method. The tape speed is 75
inches per second in streaming mode.

See Section 1.4, TK50 Tape Subsystem, for more information.

1.2.7 Device Support Restrictions

See Section 1.2.1, Processor Support, for information on processor
restrictions. There are restrictions to support for the following other
devices on Micro/RSX:

* DLVJ1

* RX50

* LK201 keyboard

* TK50

1.2.7.1 DLVJ1 Restriction

The DLVJ1 (formerly called the DLV11-J) is not supported on systems that
also include an RQDX1. Only one DLVJ1 per system is supported.

1.2.7.2 RX50 Restriction

The RX50 does not have a Bad Block Replacement Control Task (RCT)
area.

1.2.7.3 LK201 Keyboard Restriction

Some non-English LK201 keyboards do not have a backslash (\ ) character
key. To generate the CTERM enter-control-mode sequence on an LK201
keyboard, press [CTRL/4] [RET] . Do this as follows:

1. Press the CTRL key, and while holding it down, press the numeral 4
   key.

2. Press the RETURN key.

1.2.7.4 TK50 Restrictions

Using the TK50 with utilities other than BACKUP, or using the TK50 with the verify pass of BACKUP, can degrade TK50 performance and/or capacity.

The TK50 is only supported on the MicroPDP–11/23, MicroPDP–11/73, and MicroPDP–11/83.

1.2.8 IP11 and IPV11 Now a Layered Product

This release of Micro/RSX does *not* contain the IP11 driver software. It has been unbundled from the operating system and is now available only as a separate layered product.

With the introduction of the IP11 driver as a layered product, DIGITAL will be providing services for this product separately from the RSX family of operating systems. Software Product Services contract customers are entitled to

- Software Licenses and distribution kits for the IP11 driver for a nominal charge.

- Support for the IP11 driver, at no extra charge, for the current term of their RSX contract.

For further information, please contact your Software Product Services representative.

1.3 Micro/RSX Default Terminal Characteristics

Unlike RSX–11M and RSX–11M–PLUS systems, the default terminal setting on Micro/RSX terminals is SERIAL command execution. When a terminal is set to SERIAL command execution, you can execute only one command at a single time. If you wish to initiate multiple tasks from a terminal simultaneously, you can change the default to NOSERIAL by using the command SET TERMINAL/NOSERIAL.

Micro/RSX systems also differ from RSX–11M/M–PLUS systems in the way that terminals handle a CTRL/C character. On RSX–11M/M–PLUS systems, when you press CTRL/C, the default is for the system to display an explicit prompt from a command line interpreter (CLI), such as DCL> . If your terminal is set /NOSERIAL, you can use CTRL/C in this form and issue an ABORT command for one task but not for others.

When you press CTRL/C on a Micro/RSX terminal, however, the default behavior is for the system to abort the task running from your terminal. You can change this default by using the SET TERMINAL /CONTROL=C command.

1.4 TK50 Tape Subsystem

Micro/RSX Version 3.0 supports the new TK50 tape drive. The TK50 (MU:) is an integrated subsystem that consists of a controller for the Q-BUS and a TK50 streaming tape drive. The controller handles all error recovery and correction, and internally buffers multiple outstanding commands. The TK50 driver (MUDRV) reads and writes data on a 1/2-inch tape cartridge that is recorded at 6667 bpi on serial data tracks in a serial serpentine recording (modified frequency modulation) method. The tape speed is 75 inches per second in streaming mode.

The TK50 is a streaming tape subsystem. As with all streaming tape subsystems, if I/O commands are not issued to the tape unit quickly enough, the tape unit drops out of streaming mode and into start/stop mode. While in start/stop mode, each I/O command issued incurs a time penalty (about one second for the TK50).

Because the tape unit expects the next I/O command to be ready for processing when the current command finishes, it keeps the tape moving in the forward direction. It is this processing that allows the tape to stream; that is, to maintain constant forward motion. However, if the next I/O command is not ready, the tape unit moves past the position on the tape where the I/O operation was to be performed. As a result, the tape has to be repositioned; that is, backed up to the correct position on the tape.

It is this reposition operation that takes approximately one second to complete before the next I/O command can be processed. This reposition time can severely impact the performance of tape utilities if they are unable to issue commands to the controller fast enough.

In order to alleviate the performance impact of the reposition time, a padding feature is available in the TK50 firmware. This allows the controller to issue up to a preset number of padding blocks during output to the tape. Before each pad block is written, the controller checks to see if the next host I/O is ready for processing. If it is, then no more pad blocks are written and the host's I/O request is performed. If the next host command is not ready, the controller writes out another pad block.

It continues to do this until either a host I/O command is ready for processing or the number of consecutive pad blocks written reaches a controller specified maximum. In the latter case, the tape unit will drop out of streaming mode and into start/stop mode. When the next write command is issued, the controller will reposition back over the pad blocks that it wrote before actually doing the write operation. Because of this, tape capacity is not lost if the application cannot keep the unit in streaming mode. If the application does keep the unit partially in streaming mode, then some tape capacity will be lost due to the pad blocks. On input from the TK50, the controller does not return any of these pad blocks to the application program. Because of this, padding is transparent to the user's application.

The following section describes performance and capacity impact when using RSX utilities with the TK50. The performance indicated is to be used only as a guideline. The timings given will vary depending on system load, the type of disk used in disk/tape applications, and errors encountered on the tape.

1.4.1 TK50 Performance Using RSX Utilities

This section describes anticipated timings for various RSX utilities. These are only guidelines; the actual times will vary depending on system load, disk usage, and tape ECC errors encountered.

1.4.1.1 MOUNT

When mounting a TK50 tape, either foreign or as an ANSI tape, the time taken is dependent upon the position of the tape when the MOUNT command is issued. If the tape is at or near the beginning of a track, the expected time for the mount operation is approximately 20 seconds. During this time the controller performs calibration functions on the tape. If the tape is near the end of a track, the time to mount the tape can increase to as much as two minutes. This time is taken up by the actual rewind of the tape (approximately 1.6 minutes) followed by the calibration functions by the controller.

When a cartridge is inserted, it will be at the beginning of tape so the expected time, on the average, for the mount operation will be about 20 seconds. If the tape is left positioned near the end of a track by an abnormal event (for example, the system crashes during a tape operation), then this time can approach two minutes. However, this is a very rare instance.

1.4.1.2 DISMOUNT

When dismounting a tape, a REWIND command is generally issued to the tape unit as part of the dismount function. Depending on where the tape is positioned, this may take between one second (the tape is at the beginning of a track) and 1.6 minutes (the tape is at the end of a track).

1.4.1.3 BACKUP

This utility will generally keep the TK50 in streaming mode. The exceptions to this are:

- During the initial set-up phase—that is, when writing out the directory information to the tape—BACKUP will not be able to issue I/O commands to the TK50 quickly enough. As a result, the TK50 will run in start/stop mode during this phase of BACKUP's operation. The time taken for this operation is dependent on the number of directories contained on the input disk. This process can take from one to ten minutes, based on the number of directories written.

- BACKUP generally will not stream if there is a load on the system at the time BACKUP is run. This load will slow BACKUP down sufficiently so that it will not be able to issue commands to the TK50 drive fast enough to keep it streaming.

- Any use of the disk from which BACKUP is backing up or restoring will generally cause BACKUP to run the TK50 in the start/stop mode. The extra disk head movements will add sufficient time to BACKUP's processing time so that it will not be able to issue I/Os to the TK50 controller fast enough to keep it in streaming mode.

- BACKUP will not stream during its /VERIFY pass.

- BACKUP will not stream while it is sorting retrieval pointers. The number of such sort operations is dependent on the degree of disk fragmentation that exists on the input disk.

Once BACKUP begins copying data blocks out to the TK50, it will keep the TK50 unit in streaming mode provided that none of the five conditions above occur. A test case using a slow disk and 40MB of data was run. The save operation for this took 13 minutes, with the directory writes taking three minutes of this time. Estimating from the amount of tape used in this operation, and assuming that there is sufficient information on the disk to backup, BACKUP would have utilized approximately 83 percent of the total capacity (94MB) of the tape. Such a capacity rating would allow the backing up of a 75MB disk (0.83 x 94MB = 78MB). In this test

case, no verify operation was performed. However, the verify operation is estimated to take approximately one hour for this example.

1.4.1.4 FLX

The File Transfer Program (FLX) is supplied as part of the Advanced Programmer's Kit. This utility will not keep the TK50 streaming. FLX's internal design does not permit the issuing of I/O commands to the TK50 at the rate required in order to keep the tape streaming. Because it does not stream, the padding blocks added by the controller will be over-written during the reposition operation. Tape capacity will not be diminished. A test case using 180 blocks contained in nine files was run. FLX copied the files in 1.8 minutes.

1.4.1.5 COPY

This utility will not keep the TK50 streaming. Its internal design does not permit the issuing of I/O commands to the TK50 at the rate required in order to keep the tape streaming. Because it does not stream, the padding blocks added by the controller will be over-written during the reposition operation. Tape capacity will not be diminished. A test case was run using 180 blocks contained in nine files. COPY, using the TK50 as an ANSI tape, copied the information in 10.5 minutes. This time differs significantly from the FLX test case due to the overhead involved in ANSI tape processing. The default block size on ANSI tape was used for the test case.

There is an optimization that can be performed to increase throughput to an ANSI tape on the TK50 drive. This optimization does not result in streaming; it does result in fewer I/Os to the tape unit and therefore fewer repositions. The size of the block written to ANSI tape can be increased by using the /BLOCK_SIZE qualifier in COPY. A value up to 8192(decimal) bytes can be specified (the default is 512(decimal) bytes). COPY should be installed with an increment if this value is used. If it is not, COPY may issue "Not enough buffer space" error messages during certain copy operations to the tape.

1.4.2 TK50 Indicators

Because the actual TK50 tape recording surface is hidden from view, it is often difficult to determine what the tape unit is actually doing. There are two lights on the front of the TK50 tape unit that provide some information about what is happening. The two indicators lights are as follows:

1. Green indicator:

    This indicator serves a dual role. When the light is on consistently, it indicates that the tape cartridge can be safely removed from the unit. If this indicator is blinking or off, it is not safe to remove the cartridge. A blinking pattern indicates that the tape is in motion, either in a forward or reverse direction.

2. Red indicator:

    This indicator serves a multiple role. If the indicator is on consistently, a tape cartridge is loaded into the drive. Should the indicator blink slowly (for example, on for one second and then off for one second), this indicates that the tape is in use and is currently rewinding. If the indicator blinks more rapidly, it indicates a drive error has occurred. This latter condition can be cleared by depressing the LOAD/UNLOAD button twice in succession. Please note that this will cause loss of tape context.

1.4.3 TK50 Restrictions

Using the TK50 with utilities other than BACKUP, or using the TK50 with the verify pass of BACKUP, can degrade TK50 performance and/or capacity.

The TK50 is only supported on the MicroPDP-11/23, MicroPDP-11/73, and MicroPDP-11/83.

1.5 DCL

This section describes new features and a restriction for the DIGITAL Command Language (DCL) on Micro/RSX Version 3.0.

1.5.1 New DCL Features

DCL supports the following new features:

- New DCL commands
- New DCL qualifiers
- New arguments to DCL qualifiers

1.5.1.1 New DCL Commands

The following are new DCL commands for Micro/RSX Version 3.0:

DEFINE[/qualifier[s]]

        /FINAL
        /GLOBAL
        /GROUP[:g]
        /LOCAL
        /LOGIN
        /SYSTEM
        /TERMINAL:ttnn:
        /TRANSLATION_ATTRIBUTES=TERMINAL

THE DEFINE command equates a logical name to a physical device name, to all or part of a file specification, or to another logical name.

DELETE/DIRECTORY command

The DELETE/DIRECTORY command deletes a directory on a Files-11 volume and removes its name from the volume's Master File Directory (MFD).

SET HOST

The SET HOST command connects your terminal to another system.

If your system includes DECnet support, you have support for Network Command Terminals (NCT). Use the command SET HOST nodename to establish a network connection to a remote node and to establish your remote terminal as an RT: device on the host. The SET HOST command allows for transparent connections from one RSX system to another or between VAX/VMS and RSX systems.

SHOW CACHE[/qualifier]
> /DEVICE=ddnn:
> /REGION=name
> /RATE:n

The SHOW CACHE command allows you to display system information on disk data caching.

SHOW HOST

The SHOW HOST command displays the name of the processor to which your terminal is currently connected.

SHOW LOGICALS[/qualifier[s]]
> /ALL
> /GLOBAL
> /GROUP[:g]
> /LOCAL
> /LOGIN
> /SYSTEM
> /TERMINAL:ttnn

The SHOW LOGICALS command displays at your terminal all local and login logical name assignments.

SHOW PROTECTION

The SHOW PROTECTION command displays your personal default file protection code.

.5.1.2 New DCL Qualifiers

The following are new qualifiers to DCL commands for Micro/RSX Version 3.0:

ASSIGN[/qualifier[s]]
> /FINAL
> /GROUP:[g]
> /TRANSLATION_ATTRIBUTES:TERMINAL

COPY[/qualifier[s]]
> /ALLOCATION:n[.]
> /NONEW_VERSION

DEASSIGN[/qualifier[s]]
        /GROUP[:g]

INITIALIZE[/qualifier[s]]
        /BAD_BLOCKS:arg
                    NOAUTOMATIC
                    (AUTOMATIC,MANUAL)
                    (OVERRIDE,MANUAL)
        /DENSITY:arg
                HIGH
                LOW

INSTALL[/qualifier[s]]
        /FAST_MAP
        /[NO]RESIDENT_HEADER
        /[NO]TRANSLATION_ROUTINE:n
        /[NO]WRITE_BACK

LINK[/qualifier[s]]
        /CODE:(arg[s])
                CLI
                FAST_MAP
        /[NO]PRINT

MOUNT[/qualifier[s]]
        /[NO]CACHE[:(option[,option(s)])]
                    CREATE[:[regionname][:partitionname]][:regionsize]]
                    REGION:regionname
                    [NO]DIRECTORY[:n]
                    [NO]OVERLAY:[n]
                    [NO]VIRTUAL[:n]
                    [NO]READAHEAD[:n]
                    [NO]LOGICAL[:n]

SET PROTECTION[/qualifier]
                /[NO]DEFAULT

```
SET DEVICE ddnn:[/qualifier[s]]
            /[NO]CACHE[:(option[,option(s)])]
                            CREATE[:[region][:partition]][:regionsize]]
                            REGION:region
                            [NO]DIRECTORY[:n]
                            [NO]OVERLAY:[n]
                            [NO]VIRTUAL[:n]
                            [NO]READAHEAD[:n]
                            [NO]LOGICAL[:n]


SET TERMINAL[/qualifier[s]]
            /[NO]DTC01
            /[NO]HOSTSYNC
            /[NO]INQUIRE
            /[NO]LA210
            /[NO]LN03
            /[NO]LQP02
            /[NO]LQP03
            /[NO]PRO_SERIES
            /[NO]PRINTER_PORT
            /[NO]TTSYNC


SHOW ASSIGNMENTS[/qualifier[s]]
            /GROUP[:g]
            /TERMINAL:ttnn:


SHOW DEVICE[/qualifier[s]]
            /[NO]CACHE


SHOW TASKS[/qualifier[s]]
            /DEVICE:ddnn:
```

```
SHOW TERMINAL[/qualifier[s]]
             /[NO]DTC01
             /[NO]HOSTSYNC
             /[NO]LA210
             /[NO]LN03
             /[NO]LQP02
             /[NO]LQP03
             /[NO]PRO_SERIES
             /[NO]PRINTER_PORT
             /[NO]TTSYNC
```

1.5.1.3 New Arguments to DCL Qualifiers

The following are new arguments to the qualifiers for the DCL commands
INITIALIZE and MOUNT:

```
INITIALIZE[/qualifier[s]]
        /BAD_BLOCKS:arg
                    NOAUTOMATIC
                    (AUTOMATIC,MANUAL)
                    (OVERRIDE,MANUAL)
        /DENSITY:arg
                    HIGH
                    LOW


MOUNT[/qualifier[s]]
        /PROCESSOR:arg
                    UNIQUE
        /OVERRIDE:arg
                    IDENTIFICATION
```

1.5.2 DCL Restriction

This section describes a restriction for using DCL on Micro/RSX.

The following DCL commands only accept a logical name in the ddnn: format for a device name:

ALLOCATE

DEALLOCATE

DISMOUNT

MOUNT

SET DEVICE

SHOW DEVICE

SHOW TERMINAL

1.6 MCR

Monitor Console Routine (MCR) can be used on Micro/RSX systems, but DIGITAL does not provide documentation and takes no responsibility for its use.

1.7 Logical Name Support

Logical name support has been extended for Micro/RSX Version 3.0. This section contains information on the following features for logical names on Micro/RSX:

- Assigning logical strings

- Logical string translations

- Using logicals in the form ddnn:

- New directives

- FCS and RMS–11 support for logical names

1.7.1 Assigning Logical Strings

On Micro/RSX Version 3.0 systems, you can now make logical string assignments. Previously, both the logical name and its equivalence string were limited to the form ddnn:. Now you can use any alphanumeric characters for both the logical name and its equivalence string. No colon is required. For example, you can now assign the logical string SYS$LOGIN to DB3:[1,1].

The system maintains logical name and equivalence name pairs in four logical name tables. These are listed below.

Logical Name Table	Description
Task logical name table	Contains logical name entries that are local to a particular task
Session logical name table	Contains logical name entries that are local to logged-in user
Group logical name table	Contains logical name entries that can be used by users with the same group number
System logical name table	Contains entries that can be accessed by any task in the system

1.7.2 Logical Name Translations

A change has been made in the way that Micro/RSX creates and translates logical assignments. Previously, when you created a logical name, the system immediately translated the logical name to the actual physical device. No further translation was necessary when the logical was used. Now, when you create a logical name, the system assigns the logical name to the given equivalence name, but no translation is done until you use the logical. For example, if you make the following logical name assignments:

```
$ ASSIGN DB3: SY:
$ ASSIGN SY:[1,1] SYS$LOGIN
```

and issue the SHOW LOGICALS command, you see the following assignments displayed:

```
SY = DB3: (Local, Final, TT26:)
SYS$LOGIN = SY:[1,1] (Local, TT26:)
```

When you use SYS$LOGIN, it is fully translated to the physical device
DB3:. If you reassign SY: to DB2: and then use the logical string
SYS$LOGIN, SYS$LOGIN translates to DB2:[1,1].

Note
> When Micro/RSX creates, displays, or translates logical
> name assignments, the colons are dropped from the logical
> when used as part of a Files–11 file specification.

To disable recursive translation, you can make a logical assignment final
(no further translation occurs) by using the /FINAL qualifer.

1.7.3 Using Logicals in the Form ddnn:

To maintain compatiblility with existing applications, if you use the ddnn:
format for both the logical name and its equivalence name, the system
mimics the old behavior of the ASSIGN command. When the logical name
is assigned, the equivalence string is fully expanded. If the expanded string
is the specification of an existing system device, then the logical name is
assigned to that device string, and the translation is marked as final. If
the expanded string is a device specification but that device is not in the
system, then you receive the error message: "Device not in system."

If the equivalence string translates to anything other than a device, then
the system interprets the command using the new functionality of ASSIGN
and uses the string ddnn: as the logical equivalence name. In this case, if
you make the following assignments:

```
$ ASSIGN DR3: YY:
$ ASSIGN [1,1]SYSLIB.OLB SS:
$ ASSIGN SS: XX:
```

and then use the SHOW LOGICALS command, you see the following
assignments displayed:

```
YY = DR3: (Local, Final, TT26:)
SS = [1,1]SYSLIB.OLB (Local, TT26:)
XX = SS: (Local, TT26:)
```

1.7.4 New Directives

New directives have been added so that you can access logical name support in your own tasks. These directives are listed below.

Directive	Meaning
CLON$ and CLOG$	Create logical names
TLON$ and TLOG$	Translate logical names
RLON$ and RLOG$	Provide recursive translation
DLON$ and DLOG$	Delete logical names

The xxxN$ form of each directive performs exact match creation, translation, and deletion of logical names. The xxxG$ form of each directive has been provided to allow compatibility with the P/OS operating system.

On P/OS, if a logical name is used as a device, the colon on the end of the device string is considered a part of the logical name. On RSX, the colon is only a delimiter, and is not included in the logical name. Thus, the xxxG$ directives are included on both systems but have slightly different functionalities to allow a task to run on both systems.

If compatibility with P/OS is desired, use the xxxG$ form of the directives and include trailing colons in device logicals. This is the only case where the xxxG$ directives should be used. On P/OS, the xxxG$ directives will work the same way as the xxxN$ directives in that the exact string that is provided will be used. On RSX, the xxxG$ directives will remove one or two trailing colons from a logical name, and will eliminate any leading zeros in the number of the ddnn: string.

New directives have been added for file system access. These directives are listed below.

Directive	Meaning
ACHN$	Processes file specifications
PFCS$	Parses an FCS string
PRMS$	Parses an RMS–11 string

The ACHN$ directive performs the same function as the current ALUN$ directive. However, the ALUN directive only accepts input in the form ddnn:. The ACHN$ directive accepts any string as input. The PFCS$

and PRMS$ directives perform extensive logical expansion and merging and were included to minimize the code added to FCS and RMS–11. For further information on any of these directives, see the *RSX–11M/M–PLUS and Micro/RSX Executive Reference Manual*.

1.7.5 FCS and RMS–11 Support for Logical Names

File Control Services (FCS) and Record Management Services (RMS–11) have been modified to support logical names. For example, after issuing the following command:

```
$ ASSIGN DB3:[1,1] SYS:
```

you can access files in DB3:[1,1] using SYS:

```
$ DIR SYS:SYSLIB.OLB
```

For more information on using logical names for file access, see the *RSX–11M/M–PLUS I/O Operations Reference Manual*.

1.8 Disk Data Caching

Disk data caching enhances I/O performance by reducing the number of physical I/O requests directed to a disk. The disk data caching option is selected by default on Micro/RSX systems.

Disk data caching must be activated explicitly for a disk. You can activate, disable, override, monitor, or modify disk data caching using the DCL commands discussed in Chapter 19 of the *RSX–11M/M–PLUS System Management Guide*.

1.9 Indirect Command Processor

There are several new features and one restriction added to the Indirect Command Processor. The new features are as follows:

- Changes to <TITYPE> symbol

- New delimiter for symbols and expressions

- New and modified directives

- New special symbols

- New error message

The restriction involves the @/LB:module command line.

1.9.1 Changes to < TITYPE> Symbol

The <TITYPE> symbol supports a new terminal type. The type and its corresponding octal code are:

PC3xx series 35

The <TITYPE> symbol is now evaluated whenever the terminal type is requested, not just when Indirect is initialized.

1.9.2 New Delimiter for Symbols and Expressions

Pound signs (#) as well as quotation marks (") can be used as delimiters for string symbols and expressions.

1.9.3 New and Modified Directives

The following directive is new for this release:

.TRANSLATE   Translates logical names

The following directives accept logical name assignments as part of file or device specifications:

.CHAIN   .OPEN   .TESTDEVICE

.OPENA   .TESTFILE

.OPENR

1.9.4 New Special Symbols

There are two new special symbols in Indirect:

Special Symbol	Description
<SYTYP>	Contains a string describing the system type; for example, "Micro/RSX"
<VERSN>	Contains the current system version number; for example "3.0"

1.9.5 New Error Message

The following error message is new for this release:

```
AT.  --  Logical name translation error
```

This message indicates that a logical translation directive error has occurred. Use the SHOW LOGICALS command to determine the status of your logical assignments.

1.9.6 Indirect Restriction

The @/LB:module command line does not work on Micro/RSX Version 3.0 when it is issued to a CLI prompt or from within a command file. No problem occurs when it is issued from within a library.

If the module name referenced is greater than six characters, Indirect truncates the name to six characters. When a file name is truncated, Indirect returns the following error message:

```
AT.  --  File Not Found
```

Indirect assumes that the module name you wish to reference is only six characters long, since the module names are truncated to six characters by the librarian utility when the files are included in the command library.

This Indirect behavior will remain a restriction for Micro/RSX Version 3.0

To avoid this problem, always specify a module name of six characters or less, when issuing the @/LB:module command line. Also, do not use the @/LB:module command line unless you plan to include command files in a command library.

1.10 BACKUP

This section describes new features, restrictions, and supplementary information for BACKUP on Micro/RSX Version 3.0. BACKUP uses the Backup and Restore Utility, BRU.

1.10.1 New BACKUP Features

BACKUP supports the following new features:

- Skipping over a bootable system image
- New device support

1.10.1.1 Skipping over a Bootable System Image

BACKUP can now detect and automatically skip over a bootable system image (for example, BRUSYS) when appending to or restoring from a tape. This feature allows you to place a backup set onto a tape that contains a bootable system image at its beginning, and also eliminates the need for a separate tape containing the bootable system image.

When BACKUP is used to append to or restore from a tape, there will be a small delay for rewinding the tape and skipping over the bootable system image. Otherwise, this feature is transparent. See Section 1.8.2.2 for a restriction regarding the use of this feature.

1.10.1.2 New Device Support

BACKUP now supports the TK50 cartridge tape drive; the mnemonic for the TK50 is MU:. BACKUP now also supports the RD53, RC25, RA80, RA81, and RA60 disks; the mnemonic for all of these is DU:.

1.10.2 BACKUP Restrictions

There are restrictions to using the following qualifiers to BACKUP:

- /REW/APPEND

- /NEW_VERSION and /VERIFY

- /TAPELABEL

1.10.2.1 /REW/APPEND Restriction

When placing the first backup set on a tape that contains a bootable system image, you must use the /REW/APPEND qualifiers. You cannot just leave the tape positioned at the end of the bootable system image and then specify the /APPEND qualifier.

1.10.2.2 /NEW_VERSION and /VERIFY Restriction

You cannot specify the /NEW_VERSION and /VERIFY qualifiers together. If you use these qualifiers together, you will receive a "conflicting qualifiers" error message. BACKUP cannot verify a restore operation when the /NEW_VERSION qualifier is used, because the version numbers of the files on the output device may not correspond to the version numbers of the files on the input device.

1.10.2.3 /TAPE_LABEL Restriction

The /TAPE_LABEL qualifier is ignored during a restore operation if there is a bootable system image at the beginning of the tape.

1.10.3 Supplementary BACKUP Information

This section contains information about BACKUP that is not new for Version 3.0 but was not previously documented. This section includes the following topics:

- Multivolume backups for disks

- Modified BACKUP qualifiers

- Sort algorithm

- /MOUNTED qualifier error message

1.10.3.1 Multivolume Backups for Disks

BACKUP supports multivolume backup operations for disks. You can copy a large disk onto several smaller disks or several smaller disks onto a mounted large disk. However, the disks you copy to are in BACKUP (BRU) format, not Files-11. Therefore, before you can read what is saved on the disks, you must restore them with BACKUP. This process is similar to magnetic tape backup and restore operations.

The following qualifier has been added to BACKUP to perform multivolume disk backup operations:

/IMAGE:SAVE
/IMAGE:RESTORE

The /IMAGE qualifier 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.

1.10.3.2 Modified BACKUP Qualifiers

Because of the addition of multivolume disk backups, the meanings of the following BACKUP qualifiers have changed:

BACKUP Qualifier	Description
/APPEND	Directs BACKUP 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, which 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 those on the input volume, the file on the output volume is deleted and replaced with the file from the input volume.

1.10.3.3 Sort Algorithm

BACKUP has a new sort algorithm that increases the speed of the sorting procedure.

1.10.3.4 /MOUNTED Qualifier Error Message

If you use the /MOUNTED qualifier when the input device is a tape, BACKUP issues a syntax error message informing you that tapes cannot be mounted as Files–11 volumes. Previously, BACKUP ignored this qualifier if you issued it for a magnetic tape input device.

1.11 Executive

The following new Executive features are supported on Micro/RSX Version 3.0:

- New directives
- New high-level language interface

1.11.1 New Directives

The Executive supports the following new directives:

Directive	Description
ACHN$	Assign Channel
CLON$, CLOG$	Create Logical Name
DLON$, DLOG$	Delete Logical Name
FSS$	File Specification Scanner
PFCS$	Parse FCS
PRMS$	Parse RMS
RLON$, RLOG$	Recursive Translation of Logical
RRST$	Receive By Reference or Stop
TFEA$	Test for Specified Task Feature
TLON$, TLOG$	Translate Logical Name

1.11.2 New High-Level Language Interface

All of the Executive directives that can be called from a high-level language can now be used from FORTRAN and from other high-level languages. Some of these languages are BASIC-PLUS-2, PASCAL, DIBOL-83, and COBOL-81.

1.12 Task Builder

This section describes new features, restrictions, and supplementary information for the Task Builder on Micro/RSX Version 3.0.

1.12.1 New Task Builder Features

The following new features are supported for the Task Builder:

- FCSRES and FCSFSL change

- Separate instruction and data space

- Supervisor mode

- /CODE:CLI qualifier

- /CODE:FAST_MAP qualifier

- Change to behavior of an installed common

1.12.1.1 FCSRES and FCSFSL Change

FCSRES and FCSFSL are merged into a single vectored-entry memory image that can be used in either user or supervisor mode. This change reduces both disk and memory space requirements. The [1,1]FCSRES.TSK image file installs under the library name FCSRES and is used to satisfy both user and supervisor library requests.

To use the new library, you must rebuild existing tasks that reference FCSFSL, whether they reference FCSFSL directly or by referencing a library (such as F77RES) that has been built to reference FCSFSL. Existing tasks that reference FCSRES are unaffected by this change, and need not be rebuilt except to use new functionality (for example, the new logical names parser, CSI$4).

The FCS resident-library image ([1,1]FCSRES.TSK) has been built containing the supervisor-mode library completion routine. However, because the user-mode FCSRES library is a memory-resident overlaid library, it must be distinguished from the supervisor-mode library usage (because the completion routine cannot be overlaid). Thus, the files for FCSFSL must be used to build tasks using supervisor mode (the .TSK file is needed for information contained in the label block; the .STB file is needed for symbol values and overlay structure). To supply the two different descriptions but use the same library image, the Task Builder has been changed

to convert FCSFSL references to FCSRES references in the resulting executable task image. The build commands for both codes remain the same (LIBR=FCSRES or SUPLIB=FCSFSL), and files for both remain in [1,1].

Note that existing tasks referencing FCSFSL can continue to use the previous version of FCSFSL until they are rebuilt, but they cannot use the new image file supplied under the name [1,1]FCSFSL.TSK because this image does not contain executable library code. When tasks specifying FCSFSL are built with the modified Task Builder, they are forced to use the new, merged FCSRES image.

The new FCSRES image can generate a nonoverlaid user-mode reference to FCSRES for tasks having sufficient virtual address space available. This new mode results in smaller task root code (from reduced or eliminated overlay RTS support requirements) and faster task execution (from eliminating overlay RTS calls). To use the image in this new way, build the referencing task to specify the FCSFSL library files but through a user-mode reference (LIBR=FCSFSL).

The possible references and the results are summarized as follows.

Task Builder Option	Library Specification	Result
LIBR=	FCSRES	Same as existing user-mode FCSRES (PIC, memory-resident overlaid; uses one APR)
LIBR=	FCSFSL	New, user-mode nonoverlaid FCSRES requires two APRs
SUPLIB=	FCSFSL	Supervisor-mode FCSRES usage (syntax and functionality not changed; two APRs; was absolute—now PIC)
SUPLIB=	FCSRES	Invalid—completion routine cannot be overlaid

1.12.1.2 Separate Instruction and Data Space Support

Micro/RSX Version 3.0, when running on processors with appropriate hardware support, for example the DCJ11 microprocessor chip, now supports separate instruction and data space. All references in the *RSX–11M/M–PLUS and Micro/RSX Task Builder Manual* to exclusive RSX–11M–PLUS support for I- and D-space tasks now apply to Micro/RSX Version 3.0 running on these processors.

1.12.1.3 Supervisor-Mode Support

Micro/RSX Version 3.0, when running on processors with appropriate hardware support, for example the DCJ11 microprocessor chip, now supports supervisor-mode operation. All references in the *RSX–11M/M–PLUS and Micro/RSX Task Builder Manual* to exclusive RSX–11M–PLUS support for supervisor-mode tasks now apply to Micro/RSX Version 3.0 running on these processors.

1.12.1.4 I- and D-Space Task Autoload Vectors

For I- and D-space tasks (user mode), the Autoload vectors consist of two parts:

- An I-space part consisting of four words contained in the program section $$ALVI

- A D-space part consisting of two words contained in the program section, $$ALVD

1.12.1.5 I- and D-Space Task Disk Block Allocation

Disk space allocation for I- and D-space overlay segments is large enough to accommodate the I-space part and the D-space part in separate but adjacent disk blocks. The I-space part comes first on disk, and the D-space part comes second. When an overlay segment is read into memory, the I- and D-space parts are read separately. However, the speed with which I- and D-space parts are read into memory is increased over that which would be normal for two separate segments because these parts are adjacent on the disk.

1.12.1.6 Segment Descriptor Block Size

The size of the segment descriptor block that is internal to the Task Builder depends on whether an I- and D-space task is being built. For a conventional task, the size of the internal segment descriptor is 230(octal) bytes. For an I- and D-space task, the size is 262(octal) bytes.

1.12.1.7 /CODE:CLI Qualifier

The DCL qualifier /CODE:CLI indicates to the Task Builder that the task is a command line interpreter (CLI). You should use this qualifier when you task build the DCL task or any other CLI task. When you build a task with /CODE:CLI, you do not need to use the /CLI=YES qualifier when installing the task. However, if the task is built without using the /CODE:CLI qualifier, the task can still be installed as a CLI by specifying /CLI=YES.

The following command line shows the recommended placement of the /CODE:CLI qualifier:

```
$ LINK/TAS/CODE:CLI/MAP/SYM INPUTFILE
```

The default is that /CODE:CLI is not enabled.

1.12.1.8 /CODE:FAST_MAP Qualifier

The DCL qualifier /CODE:FAST_MAP indicates to the Task Builder that space must be allocated in memory between the task and the external header for use by the fast mapping feature (an Executive feature). The /CODE:FAST_MAP qualifier corresponds to the INSTALL qualifier FAST_MAP.

The following command line shows the recommended placement of the /CODE:FAST_MAP qualifier:

```
$ LINK/TAS/CODE:FAST_MAP/MAP/SYM INPUTFILE
```

The default is that /CODE:FAST_MAP is not enabled.

1.12.1.9 Change to Behavior of an Installed Common

Changes made in the memory image of a common can only be preserved when the common is removed by installing the common with the /WRITE_BACK=YES option. If the common has been installed with the /WRITE_BACK=YES option, the REMOVE command writes the common, with its changes, into its original task image file, replacing the original image. When the common is reinstalled, the new version of the common (as it existed in memory when it was removed) is placed in memory.

Note

The default is not to write back commons when they are removed. This is different from the previous version and

may require modification of applications that are dependent on the old behavior.

1.12.2 Task Builder Restrictions

The following restrictions apply to the Task Builder:

- Only referenced symbols result in Autoload vectors.

- First library specified must be first library entered.

- Base address displayed is incorrect.

- Do not cluster read-only and read-write libraries.

- You should use the /SELECTIVE_SEARCH qualifier with RSX11M.STB.

1.12.2.1 Only Referenced Symbols Result in Autoload Vectors

Autoload vectors supplied by the symbol table (.STB) files of the resident libraries are selectively included in the user's task image. Only referenced symbols result in Autoload vectors that are present in the task image.

1.12.2.2 First Library Specified Must Be First Library Entered

As stated in the *RSX–11M/M–PLUS and Micro/RSX Task Builder Manual*, the first library specified in the CLSTR option should be the first library entered by the task. If a memory-resident overlaid library has been built with a null root, the install process merely checks for the presence of the required libraries but does not map to them. When the task executes, the first library entered becomes the default library, regardless of the order in which the libraries are specified in the CLSTR option statement.

If the first cluster element specified in the CLSTR statement is not the first library entered and that library uses the stack for parameter passing, the information passed between the cluster library elements and the overlay runtime system becomes corrupted, resulting in unpredictable behavior of the executing task.

1.12.2.3 Base Address Displayed Is Incorrect

A non-PIC shared region's base address is displayed in the map as zero. The base address is not zero, but a true base address. This problem is a map-generation problem of the Task Builder.

1.12.2.4 Do Not Cluster Read-Only and Read-Write Libraries

Read-only and read-write libraries cannot be clustered together. This form of clustering is not supported.

1.12.2.5 The /SELECTIVE_SEARCH Qualifier with RSX11M.STB

When the RSX11M.STB file is used as an input file to the Task Builder, you should always append the /SELECTIVE_SEARCH qualifier to the file specification, because the .STB file contains so many symbols that, potentially, the Task Builder could exhaust its virtual memory tables.

Explicitly include on the command line the required library modules before specifying the Executive symbol definition file RSX11M.STB.

1.12.3 Supplementary Task Builder Information

This section contains information from Micro/RSX Version 1.0 that applies to the Task Builder. This section describes the following topic:

- Two forms of symbol definition in the .STB file

1.12.3.1 Two Forms of Symbol Definition in the .STB File

The symbol table (.STB) file 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, the .STB file contains a new internal symbol directory of autoloadable symbols for conventional tasks as well as the information needed to generate Autoload vectors for I- and D-space tasks.

1.13 System Library Routines

System library routines support the following new features on Micro/RSX Version 3.0:

- Date routine
- FCS parsing code

1.13.1 Date Routine

SYSLIB contains a new, alternate date routine that can be referenced by any user program. The new date routine allows users to print dates in a format that is not dependent upon the use of the DIGITAL standard date format (dd-mmm-yy). It allows you to use non-English abbreviations for dates.

To use the new routine, you must build the SYSLIB module INTDAT or INTDAS (the normal version or the supervisor mode version) into your program or library. See the *RSX–11M/M–PLUS and Micro/RSX Task Builder Manual* for the /LB: construct used to build these SYSLIB modules, which have been inserted into SYSLIB with entry points deleted.

The INTDAT and INTDAS modules contain the routines $DAT and $TIM. (These routines are described in the *IAS/RSX–11 System Library Routines Reference Manual*. The $TIM routine has not been modified; it is included simply because it is a part of the standard module. The $DAT routine's calling sequence remains the same, but the logical SYS$DATE_FORMAT now can contain the following attributes.

Attribute	Description
DD	Print two-digit day of month with leading zero
ZD	Print two-digit day of month with leading zero suppressed
MM	Print two-digit month number with leading zero
ZM	Print two-digit month number with leading zero
YY	Print two-digit year with leading zero
ZY	Print two-digit year with leading zero suppressed
MMM	Print alphabetic month (not necessarily three characters long)

You can use additional characters (other than the uppercase letters D, Z, M, and Y) in SYS$DATE_FORMAT as delimiters. Note that if SYS$DATE_FORMAT is not defined, you get the DIGITAL standard version (dd-mmm-yy) by default. SYS$DATE_FORMAT can have a maximum length of 16 characters.

The logical SYS$MONTH_nn (where nn is 01 through 12) provides the alphabetic month to be printed when the mmm attribute is used. If SYS$MONTHnn is not defined, you get the traditional 3-letter English month abbreviation (mmm) by default. SYS$MONTH_nn can have a maximum length of 12 characters.

Examples

SYS$DATEFORMAT = "MMM ZD, 19YY" SYS$MONTH06 = "June"

Output: "June 4, 1985"

SYS$DATEFORMAT = "DD.MMM.YY" c'/VSYS$MONTH06 = "VI"

Output: "04.VI.85"

Logical translation is done in standard order, which means that a local terminal assignment can override a systemwide assignment. This permits the same program to produce output in the individual user's own language or preferred format.

There are two limitations to the alternate date routine. First, using it necessitates more output buffer space because the output produced can be as long as 25 characters, whereas the standard $DAT routine always produces either 8 or 9 characters. Second, the new module can be linked with many, but not all, existing programs. An example of a program that cannot use this routine is one that performs operations on the resulting output string expecting it to be in the format produced by the standard routine.

1.13.2 FCS Directory Parsing Code Enhancement

A new feature has been added to the directory parsing code in File Control Services (FCS). You can obtain the directory string used in the parse. An extension has been created for the FDB, which has the following format:

.Byte   Extension length

.Byte   Reserved

.Byte   Length of buffer for directory string

.Byte   Length of directory string (filled in by .PRSDI)

.Word   Address of directory string buffer

The extension block and directory string buffer are allocated in your address space. To obtain the directory string used in the parse, do the following:

- Enter the address into the the proper locations in the extension block.

- Enter the length of the buffer into the proper locations in the extension block.

- Enter the length of the extension into the proper locations in the extension block.

- Insert the address of the extension block at the offset F.EXT in the FDB.

When the directory parsing code detects that F.EXT has a nonzero value, it uses the value as an address and moves the directory string into the buffer. It also puts the length of the actual directory string into the byte in the extension. This directory will always be filled in, except when the directory is obtained from the default name block because the default name block does not contain the directory string. In this case, the directory length will be set to zero.

1.14 I/O Drivers

This section describes new features, restrictions and supplementary information for I/O drivers on Micro/RSX Version 3.0.

1.14.1 New I/O Driver Features

The following new I/O driver features are supported:

- Stall I/O for the RC25

- Extended QIO function

- Modified TF.TMO subfunction

- Tape driver functional change

- New disk driver error code

- Network command terminals

1.14.1.1 Stall I/O for the RC25

Stall I/O support for the RC25 has been modified for this release. I/O is now stalled only to the fixed disk of the RC25 subsystem. The removable disk will lose context if the drive is spun down. Also, the VER task is no longer used in stall I/O processing.

Because context may be lost on the removable disk if the subsystem is spun down, all spindown requests are ignored for the fixed unit of the RC25. For the removable disk unit, you must be privileged in order to spin down the device when it is dismounted.

When you queue an I/O request to the DU: driver and it cannot be processed because of stall I/O, the following message is printed on the console:

```
DDnn: - I/O stalled
```

When the drive has been spun up again and I/O processing can continue, the following message is printed on the console:

```
DDnn: - I/O resumed
```

1.14.1.2 Extended I/O

There is a new I/O function, IO.EIO, which allows for new subfunction bits. See Section 1.14.2, I/O Driver Restriction, for more information. For a description of IO.EIO, see Chapter 2 of the *Micro/RSX I/O Drivers Reference Manual*.

1.14.1.3 Modified TF.TMO Subfunction

The TMO parameter of the TF.TMO subfunction has been modified so that you can specify the amount of time the driver waits between two input characters before the read is aborted. The amount of time TMO waits is called time-out. The maximum time-out value you can specify is 255(decimal) intervals. If 0 is specified, the read times out immediately after reading any data that may be in the type-ahead buffer. If you enter a 0, no time is allowed for you to enter characters, and all characters are read from the type-ahead buffer. For more information see the *Micro/RSX I/O Drivers Reference Manual*.

1.14.1.4 New and Modified Magnetic Tape Driver Information

The completion and meaning of certain functions have changed:

- Completion of IO.RWD for MUDRV (MU:) devices means that the rewind has been initiated; for MSDRV (MS:) devices it means that the rewind to BOT has been completed.

- IO.DSE (Data Security Erase) is valid only for the TK50. It takes approximately 35 minutes to complete. Once the operation is initiated, it is under the complete control of the TK50 controller. The IO.DSE command cannot be aborted by the host software; the controller ensures that the operation runs to completion.

- IO.SEC (Get Tape Characteristics) has specific changes for the TK25 and TK50.

- MUDRV devices do not issue select errors. If the drive is taken off line, the drive positions the tape at BOT so the off-line condition is treated as a tape position lost or powerfail.

  To access the tape again, the cartridge must be unloaded and loaded again. The tape unit will rewind to BOT. No recovery by the application is possible in such an event. If the tape was mounted as an ANSI tape, the tape context maintained by MTAACP is invalid. The tape must be dismounted and remounted in order to reinitialize the data structures used by MTAACP. If the tape was being accessed in write mode, then the file being written is incomplete, and the tape may no longer be in valid ANSI format.

- Read recovery for MSDRV devices is done as follows: Read errors are retried by rereading the block in error a predetermined number of times. For MS: devices (except for the TK25), on every eighth reread the block is passed by the tape cleaner blade. If the error persists after a predetermined number of retries, an error code, IE.VER, is returned.

- Under certain conditions, if a file is written to a tape mounted /NOLABEL, its block size will be even and one more than the value specified in the MOUNT command.

1.14.1.5 New Disk Driver Error Code

A new error code has been added:

```
IE.IFC - Invalid function for disk
```

1.14.1.6 Network Command Terminal

If your system includes DECnet support, you have support for Network Command Terminals (NCT). Use the command SET HOST nodename to establish a network connection to a remote node and to establish your remote terminal as an RT: device on the host. The SET HOST command allows for transparent connections from one RSX system to another or between VAX/VMS and RSX systems.

1.14.2 I/O Driver Restriction

The IO.EIO function will not work if your terminal has been set as a remote terminal (RT:) to another system. That is, after entering the following:

$ SET HOST xxxxx

and logging into an RT:, the terminal driver rejects a QIO issuing an extended I/O request from the RT:.

1.15 I/O Operations

This section describes a new feature for I/O operations on Micro/RSX Version 3.0.

1.15.1 File Specification Format Expanded

The format for specifying files on disk and magnetic tape has been expanded for logical name support. See the *RSX–11M/M–PLUS and Micro/RSX I/O Operations Reference Manual* for more information.

1.16 System Management

This section describes new features, a restriction, and supplementary information for system management on Micro/RSX Version 3.0.

1.16.1 New System Management Features

The following new system management features are supported:

- Password encryption

- I/O exercising for terminals

- New SHUTUP feature

- Enhancements to SYSPARAM.DAT and STARTUP.CMD

- Resource Monitoring Display (RMD) for Micro/RSX users

- New RMD features

- New Data Terminal Emulator (DTE) error message

1.16.1.1 Password Encryption

Version 3.0 of Micro/RSX supports password encryption, to provide increased system security. When a new account is created, the password for that account is automatically encrypted by the Account File Maintenance Program (ACNT).

Encryption means that once an account has been created the password cannot be displayed, even by privileged users. When a privileged user runs ACNT, (ENCRYPTED) is displayed in the password field of the ACNT record. If a user forgets a password, a new password has to be assigned.

Passwords can now be a maximum of 39 characters, although accounts used for access by DECnet must restrict password length to 8 characters.

1.16.1.2 I/O Exercising for Terminals

The Micro/RSX Version 3.0 I/O Exerciser (IOX) supports terminal I/O exercising. Terminal I/O is tested by using the LOOPBACK parameter to the SELECT command. The SELECT command is part of the IOX utility. See Chapter 4 of the *RSX-11M/M–PLUS System Management Guide* for more information.

1.16.1.3 New SHUTUP Feature

SHUTDOWN messages now include a "reason for shutdown" field in SHUTUP. You are prompted for the reason, and it is displayed in the system shutdown message.

1.16.1.4 Enhancements to SYSPARAM.DAT and STARTUP.CMD

Micro/RSX Version 3.0 contains the following enhancements to SYSPARAM.DAT and STARTUP.CMD:

- If you use the SET option to request that a terminal be slaved, the set will be performed before bringing the terminal on line.

- You can use the COMMEXEC=YES option to request that the DECnet communications Executive be kept in the Executive even if you don't have DECnet.

  This is useful for non-DECnet products that require the communications Executive, such as the 3271 protocol emulator.

- You can use the CACHE= option to enable data caching at boot time for disk devices.

The system parameter file LB:[1,2]SYSPARAM.DAT contains further information on the structure of these new options.

1.16.1.5 Resource Monitoring Display (RMD) for Micro/RSX Users

The Resource Monitor Display (RMD) is available on Micro/RSX as part of the Advanced Programmer's Kit. RMD is described in Chapters 6 and 19 of the *RSX–11M/M–PLUS System Management Guide*. However, in order to use RMD on Micro/RSX, you must use the following DCL commands:

SHOW MEMORY

SHOW TASK/DYNAMIC

SHOW TASKS/ACTIVE/DYNAMIC

SHOW CACHE/DEVICE

SHOW CACHE/REGION

1.16.1.6 New Resource Monitor Display (RMD) Features

Two new displays about cache regions have been added to RMD: the RMD C page, which displays general statistics about a cache region; and the RMD D page, which displays detailed statistics about a cached device. See Chapter 19 of the *RSX–11M/M–PLUS System Management Guide* for more information.

1.16.1.7 New Data Terminal Emulator (DTE) Error Message

The following is a new error message for the Data Terminal Emulator:

```
?DTE-F-MODERR, Unable to determine modem type
```

When you receive this message, DTE has either not received a response from the modem or is unable to interpret the response.

Check that the modem is properly connected, that the modem switches are set properly, and that the port characteristics are properly set.

1.16.2 Queue Manager Restriction

If you want to spool to the null device (NL:), you must specify the /SHAREABLE qualifier when you initialize the print processor.

1.16.3 Supplementary Queue Manager Information

This section contains information that was not included in the previous release of Micro/RSX.

The DCL /NOWARNINGS qualifier suppresses messages from the Queue Manager (QMG) such as "Queue already exists."

Once a queue has been established, it will remain until eliminated, in spite of system crashes. However, it is desirable to attempt to reestablish the standard print queues each time the system is restarted, in case a queue has been inadvertently eliminated. If you try to create a queue that already exists, you get the message "Queue already exists" from the Queue Manager. The /NOWARNINGS qualifier can be used in some cases, for example in a startup command file, to suppress the "Queue already exists" message.

1.17 Debugging

The Micro/RSX Version 3.0 Executive contains code that detects certain types of internal system corruption. If the Executive Debugging Tool (XDT) is included in the system, the Executive attempts to enter XDT as soon as the system corruption is detected. Loadable XDT uses Bugchecks to report this type of error. The *RSX–11M/M–PLUS and Micro/RSX Debugging Reference Manual* includes information on interpreting Bugchecks.

1.18 Error Logging

This section describes new features, a restriction, and supplementary information for error logging.

1.18.1 New Error Logging Features

The following new error logging features are supported:

- Internal I/O operation errors

- Error logging for non-DIGITAL devices

- Error logging control files

- Qualifiers to DCL command SHOW ERROR_LOG

1.18.1.1 Internal I/O Operation Errors

The error logging system supports error logging for internal I/O operations such as disk data caching.

1.18.1.2 Error Logging for Non-DIGITAL Devices

You no longer need to add a record to the DEVICE_INFO table in the DEVSM1 module for the error logging system to recognize a user-written device level module. After you write the device level module(s) for your devices, compile the module(s) with the DSP2P1.SYM file.

Next, insert your module(s) in ERRLOG.ULB. The name of a user-written module must be in the following form:

ExxUSR

The letters xx stand for the device mnemonic. Your device mnemonic cannot be the same as any DIGITAL-supplied device mnemonic.

Specify your module name in the MODULE statement as:

MODULE ExxUSR

In your module, you must set the variable INDICATE.TAPE_FLAG to FALSE if the device you specify is a disk.

If the device you specify is any device other than a disk, set the variable INDICATE.TAPE_FLAG to TRUE.

Remember to set your INTERMOD_DEVERR variables accordingly.

If there is a NOTES module, its name must be in the following form:

NxxUSR

1.18.1.3 Error Logging Control Files

The error logging control files now have device-specific information for the following devices:

- RX50

- RD51

- RD52

- RD53

- TK50

1.18.1.4 Qualifiers to DCL Command SHOW ERROR_LOG

The DCL command SHOW ERROR_LOG displays error logging information for the specified device or, if you do not specify a device, all devices. The SHOW ERROR_LOG command has the following qualifiers:

SHOW ERROR_LOG[/qualifier[s]]
> **/CU[RRENT]**

Displays errors that have occurred on all devices since the last time the error logging system was started using the /UPDATE qualifier.
> **/HI[STORY]**

Displays a summary of errors that have occurred on all devices since the last time the error logging system was started using the /ZERO qualifier.
> **/RE[CENT]**

Displays a brief description of errors (in chronological order) that have been recorded by the error logging system.

1.18.2 Error Logging Restriction

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 this 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.

1.18.3 Supplementary Error Logging Information

This section describes the relationship between hard and soft limits. This information was not documented in Micro/RSX Version 1.0.

Error logging 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 does not affect the logging of hard errors; the opposite is also true.

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

1.19 Layered Products Restriction

Because of support for named directories, the SET DEFAULT/NONAMED command should be used before running any command files that use the SET /UIC command.

Chapter 2

Corrections to Documentation

This chapter contains corrections to manuals included in the Micro/RSX Base Kit. The section headings in this chapter reflect the titles of manuals that require correction.

2.1 Micro/RSX User's Guide, Volumes 1 and 2

This section corrects errors in the *Micro/RSX User's Guide*, Volumes 1 and 2.

2.1.1 Description of the RUN/COMMAND

In Section 15.2.1, the description for the DCL command RUN/COMMAND should read as follows:

RUN/COMMAND:"taskcommand"

Passes the specified command line to the task you are running. The command must begin with the task's acronym and it must be valid for the task or a syntax error will occur. Also the command must be inside the quotation marks (") and not more than 40 characters long. For example, the following command:

    $ RUN/COMMAND:""MAMBO /LI) [RET]

runs MAMBO and then passes the /LI switch to it.

When the task has finished executing the command line, the task is removed.

2.1.2 < SYSTEM> Symbol Value

In Chapter 9, Section 9.4.1.2, add the following statement:

The special symbol <SYSTEM> will have a value of 5 if Indirect is running on a VAX/VMS host under VAX–11 RSX.

2.1.3 .ONERR Directive

In Chapter 9, Section 9.6.21, add the following statement:

For the .ONERR directive, the first item in the list of error conditions is not a valid condition. The item is "Task not installed in system (.XQT, .WAIT)".

2.1.4 Other Corrections

Please make the following corrections to the *Micro/RSX User's Guide*:

- Sections 5.8.2 and 12.8.2—DELETE: the /[NO]CONFIRM qualifier is missing. It is a synonym for /[NO]QUERY.

- Sections 5.9.2 and 12.10.1—PRINT: the /[NO]WIDE qualifier is not a valid qualifier.

- Section 13.7—MOUNT: the density specified with the /DENSITY qualifier cannot be altered; it can only be checked using INITIALIZE /UPDATE/DENSITY.

- Section 13.8—DISMOUNT: the /[NO]UNLOAD qualifier applies only to magnetic tape devices.

- Section 13.9.1—INITIALIZE/UPDATE: the correct arguments for the /DENSITY qualifier are HIGH and LOW. /DENSITY does not alter the density value in the Volume Home Block, but checks the density value established when the volume was initialized.

- Sections 6.5.1 and 13.10.3—SHOW DEVICES: the correct syntax for the SHOW DEVICES command is as follows:

  SHOW DEVICES[/attribute] [dd[nn:]]

  The parameter dd[nn:] is the specific device or the type of device that you want to examine. The /dd[nn:] qualifier is not a valid qualifier.

- Section 15.14.3.1—SHOW TASKS/ACTIVE: Table 15-2 should include the following two task status flags:

Status Flag	TCB Flag	Meaning
NRP	TS.NRP	Mapped to nonresident partition
RUN	TS.RUN	Running on another processor

Also, TS.CHK should be T2.CHK, and T3.RDN should be TS.RDN.

- Chapter 10—Quick Reference: the following qualifiers are either missing or incorrect in the Quick Reference:

  — DEFINE/FINAL: synonym for /TRANSLATION:FINAL

  — DELETE/[NO]CONFIRM: synonym for /[NO]QUERY

  — INITIALIZE/UPDATE/DENSITY:arg: arg is either HIGH or LOW

  — INSTALL/EXTENSION:n: n is decimal

  — HELP/LOCAL

  — MOUNT/WINDOW:arg: the three arguments are: n, (USER:n,INDEX:n), and FULL

  — PRINT/[NO]WIDE: /[NO]WIDE is not a valid qualifier

  — RUN/EXTENSION:n: n is decimal

- Chapter 16—Common Error Messages: the following message needs to be included:

`yyy -- Destination expected`

Explanation: You did not specify the destination parameter in the command line.

User Action: Reenter the command line after checking for proper syntax.

Chapter 3

RMS-11 Version 2.0 Release Notes

This chapter describes new RMS-11 features and restrictions for Micro/RSX
Version 3.0. Also included are corrections and additions to RMS-11
documentation.

RMS-11 Version 2.0 has not changed version numbers for this release of
Micro/RSX. Information in this chapter has not been incorporated into the
RMS-11 manuals.

3.1 New Features

RMS-11 Version 2.0 includes the following new features:

• New resident library configuration

• New interface to RSX logical names

• Improved remote access to RSTS/E systems

New features have also been added to several RMS-11 utilities.

3.1.1 New Resident Library Configuration

A new high performance/low system-overhead configuration has been developed for the RMS–11 resident library. The library has been partitioned into three separate segments, not all of which have to be physically resident in memory.

This partitioning allows "demand-paging" of the library segments. Only those segments containing code that is in use must be resident. The remainder of the library segments are eligible for replacement by tasks that can use the physical memory. Not more than two segments are ever required simultaneously by a given user task.

The functional breakdown for the new configuration is as follows:

- RMSRES.TSK is the "root" of the segmented library. The .TSK file and its associated .STB file are the only files needed to link user tasks against RMS–11. This segment supports $DISPLAY, $ERASE, $PARSE, $SEARCH, $RENAME, $FREE, the common internal EXTEND function, all operations for sequential files, all block I/O record operations, all magtape functions, and some common code for relative and indexed file $CREATE, $OPEN and $CLOSE operations.

- RMSLBL.TSK is the library segment that supports operations for Relative files, $EXTEND, Indexed file $OPEN, $CLOSE, $CONNECT, $DISCONNECT, $FIND and $GET operations.

- RMSLBM.TSK is the library segment that supports all remaining Indexed file operations; $DELETE, $PUT, $UPDATE, data record insertion and index update for $PUT and $UPDATE operations, bucket allocation routine and indexed $CREATE operation.

Note

All library segments do not need to be physically resident; only the root needs to be installed. However, if an operation requires a satellite that has not been installed, RMS–11 will return ER$ENV. To avoid complications, we recommend installing all segments.

Comparisons with old configuration of the RMS–11 segmented resident library:

- The minimum amount of physical memory required to use the RMS–11 resident library is approximately 22Kb instead of 16Kb. However, if relative or indexed file operations are required, then the minimum size has increased to approximately 36Kb instead of 16Kb. The old

configuration was optimized for impact on memory. It has since been determined that impact on system pool usage was a significant problem. Consequently the new configuration attempts to compromise both of these needs.

- The old library consisted of the root (RMSRES) and six satellites (RMSLBA...RMSLBF), each of approximately 8Kb. The new library consists of a root (RMSRES of approximately 22Kb), and two satellites: RMSLBL of approximately 8Kb, and RMSLBM of approximately 14Kb. The new library root combines the old RMSRES, RMSLBA, and RMSLBB. The RMSLBL satellite contains the old RMSLBC and RMSLBD. The RMSLBM satellite contains the old RMSLBE and RMSLBF.

- If most of your RMS–11 applications use only sequential file operations there will be no need for the system to load any satellites. Consequently, the overhead caused by swapping between satellites is eliminated. If your applications use relative or indexed files, a similar effect occurs. Once the appropriate satellite has been loaded, all of the support for that operation is in memory.

- The impact on system pool usage is reduced as a result of the following changes:

  - Two satellites are attached instead of six.

  - The satellites are only attached if they are required.

  - Three partitions are installed instead of seven.

- The new library root is still LB:[3,54]RMSRES.TSK. This will allow existing TKB command files to work as before. All tasks built against the old resident library will transparently use the new library. Incompatible roots and satellites will still result in the error ER$LIB.

3.1.2 New Interface to RSX Logical Names

On appropriate RMS-11 operations, the user-provided file specification will be sent to a system parse routine that will translate any logical names that are present. This support is available in the RMS-11 utilities and any layered applications for this release of RMS-11. On remote access the logical name is not parsed until it reaches the remote node.

See the following Micro/RSX Version 3.0 manuals for information on new RSX logical names:

- *Micro/RSX System Manager's Guide*

- *Micro/RSX User's Guide, Volume 1*

- *Micro/RSX User's Guide, Volume 2*

- *RSX-11M/M-PLUS and Micro/RSX Executive Reference Manual*

3.1.3 Remote Access to RSTS/E Systems

RMSDAP supports remote access to indexed files on RSTS/E systems. See Appendix B of the *RSX-11M/M-PLUS RMS-11 User's Guide* for more information on remote access.

3.1.4 New Utilities Features

This section describes the new enhancements to the following RMS-11 utilities:

- RMSCNV

- RMSBCK (part of the Advanced Programmer's Kit)

- RMSDSP

- RMSRST (part of the Advanced Programmer's Kit)

3.1.4.1 RMSCNV

The RMSCNV utility supports a new /ER switch. The purpose of this switch is to allow RMSCNV to continue processing a file after encountering an exception record. The new switch uses the following format:

/ER:filename

If no /ER switch is used, RMSCNV will terminate after encountering the first exception record. It will issue an error message indicating the type of exception record.

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

3.1.4.2 RMSBCK, RMSDSP, and RMSRST

The RMSBCK, RMSDSP, and RMSRST utilities include enhancements to increase performance and decrease task size. These utilities now process file specifications that include the wildcard characters asterisk (*) and percent sign (%). These wildcard characters can be used in any position within the directory, file name, file type, and file version number fields. The following new switches have been added to these utilities.

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

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

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

3.2 Software Restrictions

This section describes software restrictions as well as other known problems that apply to RMS–11 Version 2.0. Suggested measures for preventing or correcting problems are also included.

3.2.1 RMS–11 Access Methods

The following notes apply to RMS–11 access methods:

- RMS–11 allows the creation of an indexed file with a maximum record size that exceeds its bucket size. However, the file is not usable. You should, therefore, avoid creating an unusable file.

- Locate mode does not work when accessing a sequential fixed file if the target record ends on the block boundary. This problem impairs performance. To avoid it, do not use a fixed record size that allows records to fit perfectly in a block.

- Default Extension Quantity (DEQ) does not work correctly for sequential files. When RMS–11 extends a file, it uses the Multiblock Count (MBC) size instead of the pack default. Until this problem is fixed you should specify an explicit DEQ value instead of 0. Consult your programming language documentation for implementation details.

- Current RMS–11 tasks will return the error message "Directory Not Found" (ER$DNF) on certain file operations that are executed from an account set to nonamed directory mode (SET DEF /NONAMED). This will occur if the following conditions are true:

  1. The tasks were not built against the RMS–11 resident library.

  2. A directory is not provided in either the file specification or the default file specification.

To work around this, you should provide a directory in the file specification, or set your terminal to named directory mode (SET DEF /NAMED). If that is not possible, you should rebuild those tasks using the new version of RMS–11.

- Assigning one device to another by using logical names may cause undesirable results. If an application performs a $PARSE operation on a string, which results in a terminal device (for example DM0:), and there is another entry in the user, group, or system logical table for DM0: to be reassigned to DM1:, then on a subsequent $OPEN operation RMS–11 will try to access DM1:, not DM0:. To avoid this problem, do not use actual device names as logical names on your system.

Note

In the next major release of RMS–11, it may be possible to receive an underscore in the device name. This will correct the above problem. Consequently, it will be necessary for applications to consider this if they use the file specification returned by RMS–11 to form names for work files (since files cannot have names with an underscore in them).

3.2.2 RMS–11 Utilities

The following restrictions apply to RMS–11 utilities for Version 2.0.

RMSDES

- Using default areas—when the record size is greater than 1024 bytes, RMSDES selects area bucket sizes that are not large enough. In this case, the file is successfully created, but is unusable. To work around this problem, override the bucket size with the correct value before creating the file.

- Using default areas—when there is a larger number of duplicates than can fit in a bucket (see Chapter 6 of the *RSX–11M/M–PLUS RMS–11 User's Guide*, for calculation), RMSDES sets up larger than necessary area allocations. These can be overridden by RMSDES SET commands, or the number of duplicates can be reduced and RMSDES will allocate more conservatively.

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

- RMSDES uses the system device as the default device on a GET file DAT command. This default may be inconvenient if your login device is not the system device, or if you have set your default device to other than the system device by issuing the DCL SET DEF or MCR ASN commands. To work around this, you should supply the device name in the file specification.

RMSIFL

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

- RMSIFL uses a sort algorithm that, when called upon to sort on a key, does not preserve the FIFO ordering of duplicates. For alternate keys, it is necessary to work around this problem by using RMSCNV. If the only concern is ordering of duplicates in the primary key, then as long as your input file is an indexed file or a file sorted on primary key, you can use RMSIFL /NOSO.

RMSCNV and RMSIFL

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

3.3 Corrections to Problems with the Previous Release

This section describes corrections to software problems relating to the previous release of RMS–11 Version 2.0.

3.3.1 RMS-11 Access Methods

The following corrections apply to RMS-11 access methods:

- Records loaded into an indexed file by descending key value were not all found when either random access or sequential access was used. This problem has been corrected.

- The first word of a block in a file was sometimes corrupted with a −1 when an $UPDATE operation was performed on a sequential file. This occurred when a $FIND or an $UPDATE operation followed a $GET operation on the last record in a block, if that record ended at the end-of-file or within one byte of end-of-file. The −1 appeared in the first word of the block. This problem has been corrected.

- In some cases there were problems with Update operations to an indexed variable file containing alternate keys when record sizes were increased during $UPDATE operations. In cases where the new record no longer fit in the old bucket, the record was written to a new bucket with an incorrect RRV pointer. Later attempts to get that record by using the alternate key would return the ER$RRV condition. If you have experienced this problem, reload your indexed files with RMSIFL in order to fix the alternate key tree(s).

- A problem with the single-precision multiply routine in the access methods has been corrected. When calculations were performed that would produce a value larger than 131073, an incorrect value resulted. This value transferred the bits in the high word of the two-word value to the low word. For example, in a relative file with 3 block buckets, an attempt to insert relative record 43692 by issuing a $PUT operation, resulted in writing the record to block 3 rather than block 131076. This problem affected random $PUT and $GET operations in relative files and fixed sequential files. For relative files, the multiplication of the relative record number by the bucket size in blocks would be invalid. For sequential fixed files, the multiplication of the relative record number by the record size in bytes would be invalid. This problem has been corrected. You should reload your files if your applications were affected by this problem.

3.3.2 RMS-11 Utilities

The following corrections apply to RMS-11 utilities:

RMSCNV

- A problem with the /EO switch in RMSCNV has been fixed. This switch enables the conversion of the ASCII CTRL-Z EOF terminator to a null, and pads the rest of the file with nulls. This problem has been corrected.

RMSIFL

- Using the /DE switch allows the five RMSIFL sort files to be redirected to alternate devices rather than the device used in the input file specification. This switch failed in several ways. RMSIFL reported either "Failure to Create IFLA.TMP (ER$CRE)" or "Extraneous Data in File Specification (ER$XTR)" In some cases, it sent data output to the console device. This problem has been corrected.

- Using RMSIFL with an indirect command file (IFL @file.cmd) would return the message "?Output file must be empty" on any commands following the first command. This would abort a batch submission. This problem has been corrected.

- The RMSIFL utility was not functioning properly when it attempted to process a file with the /ER:filename switch. This problem has been corrected.

- RMSIFL was failing to display the correct exception record type in the exception record file if the exception record was detected on the alternate key. This problem has been corrected.

- RMSIFL was returning an incorrect output record count and exception record count after processing a file. Any exception detected in the alternate key caused these counts to be displayed as "????". This problem has been corrected.

- RMSIFL displayed an incorrect primary key value when exception records were detected on the primary key. Exceptions on alternate keys, however, worked fine. The problem only occurred when no /ER switch was specified. This problem has been corrected.

- There is a new form of IFLNON.CMD and IFLNRN.CMD that includes the PAR statement. This form of command file does not need to be modified when RMSIFL is rebuilt to incorporate new modules. This corrects the previous problem of needing to modify command files. See the *RSX–11M/M–PLUS and Micro/RSX Task Builder Manual* for further information on tailoring the command file to reduce the size of RMSIFL.

- RMSIFL did not correctly handle PACKED DECIMAL alternate keys. It rejected these as exception records during the alternate key loading phase. This problem has been corrected.

- RMSIFL did not return correct exit status when it encountered a corrupted index. This caused batch jobs to run incorrectly. This problem has been corrected.

- RMSIFL that is built to use supervisor mode RMSRES was aborting with a memory protection violation in certain cases. This problem has been corrected.

RMSRST (part of the Advanced Programmer's Kit)

- The /SE switch is used to selectively restore files from a container. RMSRST did not always find the files specified because of an incorrect order-based algorithm. This problem has been corrected.

- When an explicit file specification was given with the /SE switch, RMSRST found the file but continued "searching" through the tape. This problem has been corrected.

- The use of a wildcard directory in the output specification would not restore the file(s) to the original owner account. This problem has been corrected.

3.4 Documentation Corrections

The RMS–11 Version 2.0 manuals are not being revised for this release. This section describes documentation errors and directs you to the appropriate sections in the manuals to make corrections. New information to be added to the manuals is also included in this section.

3.4.1 RSX-11M/M-PLUS RMS-11 User's Guide

Please make the following corrections to the *RSX-11M/M-PLUS RMS-11 User's Guide*:

- In Section 2.2.3.3, please add the following information to the discussion of deadlock:

  An application should use multistream rather than multi-channel access to write to the same indexed file. When RMS-11 updates an RRV in a bucket that is currently locked, it must wait for that lock to be released. Control will not be returned to the program until this release occurs. Deadlock will occur when the lock is held on another channel within the same program; however, RMS-11 can update an RRV in a bucket that is locked on another stream within the same program. See your programming language documentation for details on the implementation of multistreaming.

- In Section 6.2.4, the discussion of writing a record, please add the following note:

  In the event that the record includes a partial alternate key but is not large enough to include space for the full alternate key field, RMS-11 will act as follows:

  RMS-11 will treat the alternate key as if it were not present in the record, making no entry in the alternate key index structure.

- In Section 6.3, the discussion of contiguity and areas, please add the following information:

  You will obtain a small benefit by setting areas to contiguous on a noncontiguous multi-area file, however, there is no means for RMS-11 to determine if those areas remain contiguous. Consequently, RMSDSP and RMSDES will display them as noncontiguous. As long as the areas are preallocated, they will behave like contiguous areas; as soon as they need to be extended, they will not behave like contiguous areas.

- In Section 8.1.2.1, the discussion of task building against the RMS-11 resident library, incorrect syntax is documented for the cluster option in the Task Builder command file. The correct syntax is as follows:

  ```
  CLSTR = RMSRES,DAPRES:RO
  ```

- In Appendix B, the discussion of remote file and record access using DECnet, the documentation states that the RSTS/E FAL does not support remote record access to indexed files. This is no longer true.

3.4.2 RSX-11M/M-PLUS RMS-11 Utilities Manual

Please make the following correction to the *RSX-11M/M-PLUS RMS-11 Utilities Manual*:

- In Section 3.3.2, the discussion of exception records, the exception record codes should be modified to read as follows:

  002: Record contains a duplicate key where not allowed

  This change indicates a duplicate in either the primary or alternate key fields when no-duplicate was specified.

3.4.3 RSX-11M/M-PLUS RMS-11 Macro Programmer's Guide

Please make the following changes to the *RSX-11M/M-PLUS RMS-11 Macro Programmer's Guide*:

- In Section 2.3, the argument for P$BUF is "bufcount". This is incorrect. The correct argument is "iopoolsize", as discussed in Section 2.3.4.

- In Section 5.19, the last paragraph incorrectly describes the use of the FID field in the NAM block. It should read as follows:

  "If this value is non-zero..."

- Chapter 7 includes example programs that demonstrate certain RMS-11 operations. The instructions provided for building these programs are used to build RMS-11 code directly into the task. In order to reduce the size of the Micro/RSX installation kit, the RMSLIB.OLB contains only those modules necessary for building tasks to use the RMS-11 resident library. Consequently, it is not possible for you to build from these instructions. Use the following command lines (which use the MCR extended command line format) instead:

  ```
  MACRO LB:[1,1]RMSMAC/LIBRARY,SY:[UIC]PARSE

  LINK/CHECKPOINT:SYSTEM/OPTIONS SY:PARSE,GSA,-
  ->LB:[1,1]RMSLIB/INCLUDE:ROAUTL:-
  ->ROIMPA:RMSSYM:ROEXSY,VMLIB/INIDM:EXTSK

  Option? LIBR=RMSRES:RO
  Option? CR
  ```

  See the *RSX-11M/M-PLUS RMS-11 User's Guide*, for more information about building RMS-11 tasks.

- In Appendix A, page A-8, please add the following sentence:

  An attempt to insert a record that is too small to contain the whole primary key field may also return the error ER$KEY.

- Also in Appendix A, please add the following error conditions:

  ```
  ER$BEQ    Bad logical equivalence string
            (Octal:  177510, Decimal:  -184)

  ER$FTB   File specification too big due to logical name
            (Octal:  176364, Decimal:  -780)

  ER$TRN   Too many logical name translations
            (Octal:  174550, Decimal:  -1688)
  ```

3.5 Files and Placement on the Distribution Kit

The following tables describe the contents of the RMS–11 Version 2.0 distribution kit for Micro/RSX Version 3.0. Files listed in Table 3-1 are distributed on both the Micro/RSX Base Kit and the Micro/RSX Advanced Programmer's Kit. Files listed in Table 3-2 are distributed on the Micro/RSX Advanced Programmer's Kit only.

Table 3–1: RMS–11 File Placement on Micro/RSX Distribution Kits

File Name	Destination	Comments
RMSRLX.ODL	LB:[1,1]	ODL for use with RMSRES
DAPRLX.ODL	LB:[1,1]	ODL for use with DAPRES
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
RMSDSP.TSK	LB:[3,54]	RMSDSP utility; uses RMSRES
RMSCNV.TSK	LB:[3,54]	RMSCNV utility; uses RMSRES
RMSDES.TSK	LB:[3,54]	RMSDES utility; uses RMSRES
RMSIFL.TSK	LB:[3,54]	RMSIFL utility; uses RMSRES
RMSDES.IDX	LB:[1,2]	Indexed help file used by RMSDES
RMSRES.TSK	LB:[1,1]	TSK image for linking against RMSRES, as a non-supervisor mode library

Table 3-1 (Cont.): RMS-11 File Placement on Micro/RSX Distribution Kits

File Name	Destination	Comments
RMSLBL.TSK	LB:[1,1]	Task image for RMSLBL segment
RMSLBM.TSK	LB:[1,1]	Task image for RMSLBM segment
RMSRES.STB	LB:[1,1]	STB file for RMSRES
DAPRES.TSK	LB:[1,1]	Task image for RMSDAP resident library
DAPRES.STB	LB:[1,1]	STB file for DAPRES
RMSRES.TSK	LB:[3,54]	Task image for linking in supervisor mode; also installed as the root of the library (supervisor mode and non-supervisor mode)
RMSRES.STB	LB:[3,54]	STB file for linking supervisor mode tasks

Table 3-2: RMS-11 File Placement on Advanced Programmer's Kit Only

File Name	Destination	Comments
MACRO-11 Option		
RMSMAC.MLB	LB:[1,1]	For MACRO-11 programs
RMS.HLP	LB:[1,2]	General HELP file
RMSCNV.HLP	LB:[1,2]	HELP for RMSCNV
RMSDES.HLP	LB:[1,2]	HELP for RMSDES
RMSDSP.HLP	LB:[1,2]	HELP for RMSDSP
RMSIFL.HLP	LB:[1,2]	HELP for RMSIFL
MAGTAPE Option		
RMSBCK.HLP	LB:[1,2]	HELP for RMSBCK
RMSRST.HLP	LB:[1,2]	HELP for RMSRST
RMSBCK.TSK	LB:[3,54]	MSBCK utility; uses RMSRES
RMSRST.TSK	LB:[3,54]	RMSRST utility; uses RMSRES

Note

All RMSDAP files can be deleted if you are not using RMS–11 to access files on remote nodes.

3.6 RMS–11 Version 2.0 Installation

On Micro/RSX systems, all RMS–11 files are automatically installed when you install your system.

RMS–11 resident libraries are built into your system. If you have DECnet on your system, see the DECnet kit documentation for details on installing the RMS-DECnet interface library (DAPRES).

Appendix A

The DIGITAL Equipment Computer User's Society (DECUS)

DECUS, the DIGITAL Equipment Computer User's Society, is one of the largest and most active user groups in the computer industry. It is a not-for-profit association, supported and administered by DIGITAL, but actively controlled by members. DECUS headquarters, located in Marlborough, Massachusetts, administers all international policies and activities. Members include individuals who have purchased, leased, ordered, or used a DIGITAL computer, or anyone who has a genuine interest in DECUS. Membership is free and voluntary.

A.1 DECUS Goals

The primary goals of DECUS are as follow:

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

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

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

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

To further these goals, DECUS serves its members by holding symposia; maintaining a program library; publishing an association newsletter, technical newsletters, and books; and supporting a number of Special Interest Groups (SIGs).

A.2 DECUS Activities

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

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

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

DECUS has a Program Library, which contains over 1,700 software packages that are written and submitted by users. A wide range of software is offered, including languages, editors, numerical functions, utilities, display routines, games, and other types of application software. Library catalogs are available that contain program descriptions and ordering information.   The programs are available for a nominal service charge that covers the cost of reproduction and media.

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

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

Index

RMSDSP (cont'd.)
  switches, 3-6
RMSIFL, 3-8, 3-10
RMSRST, 3-5, 3-11
  switches, 3-6
RQDX3 controller, 1-3
RUN/COMMAND command
  correction, 2-1

S

Separate instruction and data
    space, 1-29
  Autoload vectors, 1-30
  disk block allocation, 1-30
  J11 processor, 1-29
  segment descriptor block size,
    1-30
SET command
  SET DEVICE command, 1-15
  SET HOST command
    restriction, 1-39
SHOW command
  SHOW DEVICES
    correction, 2-2
  SHOW TASKS/ACTIVE
    correction, 2-3
SHUTDOWN message, 1-40
SHUTUP
  new SHUTDOWN message,
    1-40
Stall I/O
  RC25, 1-37
STARTUP.CMD
  enhancements, 1-41
Supervisor mode, 1-30
  J11 processor, 1-30
SYSPARAM.DAT
  enhancements, 1-41
System library routines
  date routine, 1-34
  FCS directory parsing code,
    1-35

T

Task Builder, 1-28
  autoload vector, 1-32
  FCSRES and FCSFSL change,
    1-28
  restriction on clustering
    libraries, 1-33
  restrictions, 1-32
  /SS switch with RSX11M.STB,
    1-33
  symbol definition in the .STB
    file, 1-33
Terminal characteristics
  CTRL/C, 1-6
  /SERIAL, 1-6
Terminal I/O exercising, 1-40
TF.TMO subfunction
  modification, 1-37
TK25 tape subsystem, 1-4
TK50 tape subsystem, 1-5, 1-7,
    1-8
  BACKUP, 1-9
  COPY, 1-10
  DISMOUNT, 1-9
  FLX, 1-10
  indicator lights, 1-11
  modifications to tape driver,
    1-38
  MOUNT, 1-8
  restriction, 1-6, 1-11

X

XDT
  Bugchecks, 1-42

USER'S COMMENTS

Your comments and suggestions are welcome and will help us in our continuous effort to improve the quality and usefulness of our documentation and software.

Remember, the system includes information that you read on your terminal: help files, error messages, prompts, and so on. Please let us know if you have comments about this information, too.

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.

_____

_____

_____

_____

_____

_____

What kind of user are you?   ___ Programmer   ___ Nonprogrammer

What do you use the system for? _____

Years of experience as a computer programmer/user: _____

Name _____ Date _____

Organization _____

Street _____

City _____ State _____ Zip Code _____
                                               or Country