RSX-11M/RSX-11S
Release Notes

Order No. AA-2573E-TC

digital

READ RSX-11M SOFTWARE DISPATCH REVIEW
FOR ADDITIONAL INFORMATION

RSX-11M/RSX-11S
Release Notes

Order No. AA-2573E-TC

RSX-11M Version 3.1
RSX-11S Version 2.1

To order additional copies of this document, contact the Software Distribution Center, Digital Equipment Corporation, Maynard, Massachusetts 01754

digital equipment corporation · maynard, massachusetts

First Printing, November 1974
Revised: September 1975
January 1977
April 1977
December 1977

The information in this document is subject to change without notice and should not be construed as a commitment by Digital Equipment Corporation. Digital Equipment Corporation assumes no responsibility for any errors that may appear in this document.

The software described in this document is furnished under a license and may only be used or copied in accordance with the terms of such license.

No responsibility is assumed for the use or reliability of software on equipment that is not supplied by DIGITAL or its affiliated companies.

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

The following are trademarks of Digital Equipment Corporation:

DIGITAL	DECsystem-10	MASSBUS
DEC	DECtape	OMNIBUS
PDP	DIBOL	OS/8
DECUS	EDUSYSTEM	PHA
UNIBUS	FLIP CHIP	RSTS
COMPUTER LABS	FOCAL	RSX
COMTEX	INDAC	TYPESET-8
DDT	LAB-8	TYPESET-11
DECCOMM	DECSYSTEM-20	TMS-11
ASSIST-11	RTS-8	ITPS-10

RSX-11M/RSX-11S RELEASE NOTES

1.0 GENERAL NOTES OF IMPORTANCE

1.1 RSX-11M V3.1 Features

RSX-11M V3.1 offers a modest set of enhancements to Release 3 as well as improved performance and superior reliability. Naturally, the size constraints of the previous releases have been upheld; systems running under RSX-11M V3.1 should be about the same size as their predecessors.

Non-privileged tasks that ran under RSX-1M V03 will run as is with RSX-11M V3.1. Privileged tasks and most drivers need only be relinked, because internal modifications between releases were minimal.

Following is a summary of noteworthy enhancements in RSX-11M V3.1. Consult the documentation set for more complete descriptions.

Device Support:

RM03 - 67Mb Disk for the PDP-11/70

LPA11-K - Laboratory Peripheral Accelerator

Laboratory K-Series Modules
- Modular software implementation that bypasses need for a formal device driver
- Enhanced data throughput
- Callable from FORTRAN, BASIC-PLUS-2, and MACRO-11 programs
- Designed to match LPA-11K support routines.

Patient Powerfail Support
- Disk drivers will wait for drives to resume spinning before retrying the operation
- Magnetic tape drivers detect loss of vacuum and prevent accidental loss of tape volume due to improper tape position.

Write-check extended to all supported disks except RX01.
Error Logging extended to the RX01 and the new RM03.

Executive Enhancements:

The Connect to Interrupt directive,

Dynamic UMR allocation for PDP-11/70 systems,

Conversion of Null Task into Kernel state to reduce context switching,

Unlimited outstanding I/O requests for a checkpointable task,

IAS/RSX-11D compatibility features to
- Support all forms of directive macros,
- Receive Data by specific task name,
- Return default protection UIC from Get Task parameters directive.

Support for RMS record locking,

Crash Dump Analyzer (CDA) support.

MCR Enhancements:

Buffered display output to eliminate system lockouts,

Indirect command processor for
- Improved string and numeric capabilities,
- Call, Return and Chain capability,
- New .IF command and symbols to return system information,
- Handling up to four open data files.

ATL/TAL for a specific task,

ACT for a specific terminal,

DEV for a specific device type,

HELP supports the selective retrieval of information.

Multiuser Support:

HEL and BYE log to the system console terminal,

ACNT enhancements
- Maintains file in sorted order,
- Allows account file access by a privileged user,
- Selective account listings with printer attach/detach logic.

Utility Enhancements:

EDT - New DEC EDITOR,

DSC support for transfer verification (/VE) or volume comparison (/CMP),

Standalone version of BAD,

CDA - Crash Dump Analyzer,

DMP support for 1600 bits/in magnetic tapes.

New Pseudo-device LB: used by Macro, TKB, INS, LOA, UNL, etc.

Support for 132-character MAC command input with continuation lines,

New switch in PIP (/SB) for copying files from ANSI magtape to disk,

Print Spooler Task (PRT) is tolerant of records bigger than 132 characters.

Miscellaneous:

SYSGEN

- New kit for dual RK05 systems provides a better SYSGEN procedure,
- SYSGEN3 reworked to manage disk space better and to build RMS-11 utilities,
- Done totally by indirect command files making process easier to understand, maintain, and extend.

New Documentation:

DEC EDITOR Reference Manual
RSX-11M Crash Dump Analyzer Reference Manual
IAS/RSX-11 System Library Routines Reference Manual

GCML supports command input greater than 80 characters and allows continuation lines.

1.2 Reporting Problems and Errors

When completing a Software Performance Report (SPR) form, please describe only one problem at a time to simplify record-keeping and to facilitate a speedy response.

User problems are often difficult to reproduce in a different configuration, so please help us by defining as precisely and as simply as possible the state of the system when the problem occurred. Illustrate a general problem with one or more examples. When an SPR contains concise information about a problem's context, we are more likely to reproduce it accurately, and therefore more likely to correct it. In addition, please ask simple and direct questions so that we can answer them clearly and directly. When you refer to a manual, specify the full title and the appropriate section, table, or page number.

When reporting problems that involve DIGITAL-supplied software, please supply the following information (on a machine-readable medium if possible):

- A copy of the Executive task-build map if the problem involves a system crash,
- A copy of the system's Executive configuration file RSXMC.MAC,
- The console output from CRASH and a Panic Dump listing.
- The version numbers of relevant software and information about patches or fixes made to the task image in question,
- The task termination console output (if appropriate) if the problem involves a system utility, as well as a map if the task image differs from that distributed.
- Any available information about the reproducibility of the problem.

If a system failure occurs while running privileged, add-on software, such as DECNET, XYDRV, etc., you should try to reproduce the failure with the additional software not installed. Then, on your SPR, be sure to indicate clearly the behavior of the system with and without the add-on software. Without this information, the maintainers probably will not be able to reproduce your problem.

1.3 Unsupported Software

As a courtesy to our customers, DIGITAL is supplying the object libraries and command files to build three tasks that are normally available only through DECUS. These are:

1. TECO - a character-oriented editor,
2. RUNOFF - a text-processing program, and
3. SRD - a program that produces sorted directory listings.

This software is not supported by DIGITAL but is supplied to provide additional useful capabilities. Documentation of these three programs is available only through DECUS.

These utilities may be built by the SYSGEN3 indirect command file, which is fully documented in the RSX-11M System Generation Manual.

1.4 Applying Corrections to Released Software

There are several ways to make corrections to released software:

- At the source level, using the Source Language Input Program (SLP),
- At the object level, using the Object Module Patch Utility (PAT), and
- At the task image level, using the Task/File Patch Utility (ZAP).

All three methods require the user to adhere to the following rules:

- Never delete the unmodified, distributed version of the file unless you have a copy. The distributed version is always the base to which corrections are applied.
- Never delete the correction file; this file contains an accumulation of all corrections made to the distributed file, but if disk space is at a premium, store all correction files on separate media.

1.4.1 Applying Corrections to Files on the Source Disk - Interim changes to the Executive, MCR, and the file system (F11ACP) are made by creating correction files to be processed by the Source Language Input Program (SLP). SLP uses these files to generate a new copy of the modules that contained the errors by applying the corrections to the distributed source file. SLP is fully described in the RSX-11 Utilities Procedures Manual.

Once the corrections have been applied and a new version of the file has been obtained, DO NOT delete the original source file. All interim changes are cumulative and are based on the contents of the distributed sources.

The diverse distribution media for RSX-11M make it impractical to provide a comprehensive set of examples for applying source corrections. The examples provided are intended to illustrate the general techniques and files to be used. They assume a privileged

user executing on a multiuser, mapped, dual RK05-based system with the dual RK05 distribution kit.

If you have a single RK05 system, disk-to-disk transfers using PIP must be made using FLX in two steps: (1) transfer from RK05 to secondary storage medium, then (2) transfer from secondary storage medium to the target RK05. If you are not using a multiuser system, you do not allocate and deallocate devices. If your system disk is large (RK06, RP02, RP03, RP04, RP05, or RP06), it contains all the necessary files, and disk-to-disk file transfers are not required.

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

```
>TIM mm/dd/yy hh:mm
>SET /UIC=[11,40]
>ALL DK1:
>MOU DK1:EXECSRC
>ASN DK1:=SY:
>EDI REQSB.COR
[CREATING NEW FILE]
INPUT
.
.
.
*EX
[EXIT]

>SET /UIC=[11,10]
>SLP @[11,40]REQSB.COR
>
```

If the updated Executive module in your system is not a loadable driver, you will probably have to rebuild the Executive and privileged tasks in order to incorporate the modification.

To rebuild, you need to repeat most of SYSGEN2 after making a few preparations. In brief:

- Assemble the new module using the RSXMC.MAC file for the target system.
- Use LBR to replace the old version of the module in the target system's RSX11M.OLB.
- Rebuild the Executive using the target system's RSXBLD.CMD.
- Use SYSGEN2.CMD to rebuild the privileged tasks and to establish your system with VMR. (Answer "yes" to HAVE YOU ALREADY BUILT THE EXEC.) Note that before initiating VMR, you must assign devices SY: and LB: to the disk containing your target system.

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

Boot the target disk.

```
>TIM mm/dd/yy/ hh:mm
>ALL DK1:
>MOU DK1:EXECSRC
>SET /UIC=[1,24]  ![1,20] for unmapped systems
>MAC ZZDRV=[1,1]EXEMC/ML,[200,200]RSXMC/PA:1,DK1:[11,10]ZZDRV
>DMO DK1:
>DEA DK1:
>LBR RSX11M/RP=ZZDRV
>PIP ZZDRV.OBJ;*/DE
>
```

Use the indirect command file [200,200]BLDDRV.CMD to produce the files ZZDRV.TSK and ZZDRV.STB. Note that if you do not want the task-build map for the driver, BLDDRV throws it away by outputting it to the null device (NL:). If your system does not have an NL:, TKB will output a diagnostic message.

If necessary, copy ZZDRV.TSK and ZZDRV.STB into the UFD corresponding to the system UIC.

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

Use VMR to unload the old device driver and to load the new one. If the new driver is larger than the old one, it may not fit into the same locations as the old one. It may be necessary to unload and reload all the loadable drivers in that partition to create enough room. If the system is unmapped, some other drivers may have to be rebuilt.

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

Hardware boot the modified system.

Updating an MCR source file is similar to updating an Executive source file, except that the MCR source files are on the disk labeled MCRFCP in UFD [12,10]. The steps to update a source file for MCR (single-user and multiuser) and external MCR tasks are the same. The procedures for rebuilding the components differ slightly. An example is provided for each procedure.

To update the MCR source file SETOV.MAC, mount the RK05 disk labeled MCRFCP.

Create the SLP correction file SETOV.COR and use it to update SETOV.MAC. Assemble SETOV for the target system.

```
>TIM mm/dd/yy hh:mm
>SET /UIC=[12,40]
>ALL DK1:
>MOU DK1:MCRFCP
>ASN DK1:=SY:
>EDI SETOV.COR
[CREATING NEW FILE]
INPUT
.
.
.
*EX
[EXIT]


>SET /UIC=[12,10]
>SLP @[12,40]SETOV.COR
>
```

Boot the target system, if necessary.

```
>TIM mm/dd/yy hh:mm
>ALL DK1:
>MOU DK1:MCRFCP
>SET /UIC=[1,24]     !or [1,20] for unmapped systems
>MAC SETOV=[200,200]RSXMC/PA:1,[1,1]EXEMC/ML,DK1:[12,10]SETOV
>DMO DK1:
>DEA DK1:
>
```

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

- Copy the necessary indirect command files and libraries from the RK05 containing the privileged object files to SY:.
- Incorporate the updated module into MCR's object library.
- Rebuild MCR, and install it into the system using VMR, ensuring that SY: and LB: are first assigned to the disk containing the target system.

All of the task-build command files output a map to logical device MP:. MP: must be assigned to NL: or other device to avoid a diagnostic error message from TKB.

Example for a single-user MCR:

```
>ALL DK1:
>MOU DK1:SYSTEM
>SET /UIC=[1,24]   !or [1,20] for unmapped systems
>PIP /NV=DK1:MCRBLD.CMD,MCRBLD.ODL,MCR.OLB
>DMO DK1:
>DEA DK1:
>LBR MCR/RP=SETOV
>PIP SETOV.OBJ;*/DE
```

If it was necessary to modify the MCR task-build command file (MCRBLD.CMD) during the last system generation, it may be necessary at this time to replicate those changes.

```
>ASN SY:=MP:
>TKB @MCRBLD
>SET /SYSUIC
SYSUIC=[g,m]
>SET /UIC=[g,m]
>PIP /NV=[1,54]MCR.TSK   !or [1,50] for unmapped systems
>VMR
ENTER FILENAME: RSX11M
VMR>REM ...MCR
VMR>INS MCR
VMR>^Z
>RUN $SHUTUP
```

Hardware boot the system.

Example for a multiuser MCR:

```
>ALL DK1:
>MOU DK1:SYSTEM
>SET /UIC=[1,24]   !or [1,20] for unmapped systems
>PIP =DK1:MCRMUBLD.CMD,MCRMUBLD.ODL,SYSBLD,SYSROT.OBJ,MCR.OLB
>DMO DK1:
>DEA DK1:
>LBR MCR/RP=SETOV
>PIP SETOV.OBJ;*/DE
```

If it was necessary to modify the MCR task-build command file (MCRMUBLD.CMD) during the last system generation, it may be necessary at this time to replicate those changes.

```
>ASN SY:=MP:    !if your system supports ASN
>TKB @MCRMUBLD
>SET /SYSUIC
SYSUIC=[g,m]
>SET UIC=[g,m]
>PIP /NV=[1,54]MCRMU.TSK,SYS    !or [1,50] for unmapped systems
>VMR
ENTER FILENAME: RSX11M
VMR>REM ...SYS
VMR>REM ...MCR
VMR>INS MCRMU
VMR>INS SYS
VMR>^Z
>RUN $SHUTUP
```

Hardware boot the system.

There is only one procedure for replacing an external MCR task. It involves the following steps:

- Copy the necessary command files and libraries from the disk containing the privileged object files to SY:.
- Incorporate the updated module into the task's object library.
- Rebuild the task and install it into the system using MCR or VMR. Before using VMR, you must assign SY: and LB: to the disk containing the target system.

Example of the module INSLB of the external MCR task Install:

```
>ALL DK1:
>MOU DK1:SYSTEM
>SET /UIC=[1,24]  !or [1,20]for unmapped systems
>PIP /NV=DK1:INSBLD.*,INS.OLB,MCR.OLB
>DMO DK1:
>DEA DK1:
>LBR INS/RP=INSLB
>PIP INSLB.OBJ;*/DE
```

If it was necessary to modify the external task's task-build command file during the last system generation, it may be necessary at this time to replicate those changes.

```
>ASN SY:=MP:       !if your system supports ASN
>TKB @INSBLD
>SET /SYSUIC
SYSUIC=[g,m]
>SET /UIC=[g,m]
>PIP /NV=[1,54]INS.TSK    !or [1,50]for unmapped systems
>VMR
ENTER FILENAME: RSX11M
VMR>REM ...INS
VMR>INS INS
VMR>^Z
>RUN $SHUTUP
```

Hardware boot the system.

Updating F11ACP can be confusing since the system includes three different versions of F11ACP. Understanding how these versions relate to one another is necessary to avoid confusion.

FCP and BIGFCP contain the same modules and therefore build from the same library (FCP.OLB). The two versions have the same capablities, but different overlay structures. BIGFCP is faster and bigger (5.5K versus 2.5K).

FCPNMH, which has fewer capablities than FCP or BIGFCP (it is a Non-MultiHeader F11ACP), is built from two libraries: FCP.OLB and FCPNMH.OLB. The modules taken from FCP.OLB are the same modules used to build FCP and BIGFCP. The modules taken from FCPNMH.OLB have the same names as modules in FCP.OLB; however, because of conditional assembly, the modules from FCPNMH.OLB contain different object code.

All F11ACP updates start by creating a SLP correction file in the UFD [13,40] (for RK05 distributions, the UFDs [13,40] and [13,10] are on the disk labeled MCRFCP). In the following example, WTRN1 replaces a dummy name.

Mount the disk labeled MCRFCP

```
>TIM mm/dd/yy hh:mm
>SET /UIC=[13,40]
>ALL DK1:
>MOU DK1:MCRFCP
>ASN DK1:=SY:
>EDI WTRN1.COR
[CREATING NEW FILE]
```

```
INPUT
.
.
.
*EX
[EXIT]

>SET /UIC=[13,10]
>SLP @[13,40]WTRN1.COR
>
```

The updated source module is then assembled and the resulting object module is used to update a library. A module can be classified into one of three categories depending on the library into which the corrected module must be inserted. SPR responses will identify which category applies to the module in question. Note that a particular module's category can change from one SPR to another. The deciding factor is in which section of the conditionally assembled code the error occurs.

- FCP only -- The source module is assembled with the prefix file F11PRE.MAC, and the resulting object module is placed only in FCP.OLB. This occurs when the error is in a module that is only in FCP.OLB, or when the error is in conditionally assembled code that is enabled only for FCP and BIGFCP.

- FCPNMH only -- The source module is assembled with the prefix file PRE.MAC, and the resulting object module is placed only in FCPNMH.OLB. This occurs when the error is in conditionally assembled code that is enabled only for FCPNMH.

- FCP and FCPNMH -- The source module is assembled twice. The first time with the prefix file F11PRE.MAC. The resulting object file is placed in FCP.OLB. The second assembly is done with the prefix file PRE.MAC, and the resulting object file is placed in FCPNMH.OLB. This occurs when the error is in unconditionally assembled code in a module that is in both FCP.OLB and FCPNMH.OLB.

Example for an FCP only module (WTRN1 replaces a dummy name):

```
Boot the target system

>TIM mm/dd/yy hh:mm
>ALL DK1:
>MOU DK1:MCRFCP
>SET /UIC=[1,24]  !or [1,20] for unmapped systems
>MAC WTRN1=[1,1]EXEMC/ML,[200,200]RSXMC/PA:1,DK1:[13,10]F11PRE,WTRN1
>DMO DK1:
>DEA DK1:
>
```

Example for an FCPNMH module (WITRN replaces a dummy name):

```
Boot the target system

>TIM mm/dd/yy hh:mm
>ALL DK1:
>MOU DK1:MCRFCP
>SET /UIC=[1,24]   !or [1,20] for unmapped systems
>MAC WITRN.NMH=[1,1]EXEMC/ML,[200,200]RSXMC/PA:1,DK1:[13,10]PRE,WITRN
>DMO DK1:
>DEA DK1:
>
```

Example for an FCP and FCPNMH module (CLACC replaces dummy name):

Boot the target system

```
>TIM mm/dd/yy hh:mm
>ALL DK1:
>MOU DK1:MCRFCP
>SET /UIC=[1,24]   !or [1,20] for unmapped systems
>MAC CLACC=[1,1]EXEMC/ML,[200,200]RSXMC/PA:1,DK1:[13,10]F11PRE,CLACC
>MAC CLACC.NMH=[1,1]EXEMC/ML,[200,200]RSXMC/PA:1,DK1:[13,10]PRE,CLACC
>DMO DK1:
>DEA DK1:
>
```

FCP, BIGFCP and FCPNMH require similar but not identical procedures for incorporating the modified module into the system. All procedures involve the following steps:

- Copy the necessary command files and libraries from the RK05 disk containing the privileged object files to SY:.
- Incorporate the updated module into an object library.
- Rebuild F11ACP, and install it into the system using VMR, ensuring that SY: and LB: are first assigned to the disk containing the target system.

All of the task-build command files output a map to logical device MP:. MP: must be assigned to NL: or other device to avoid a diagnostic error message from TKB.

Example for FCPNMH (2.0K):

```
>ALL DK1:
>MOU DK1:SYSTEM
>SET /UIC=[1,24]   !or [1,20] for unmapped systems
>PIP /NV=DK1:FCPNMHBLD.CMD,FCPNMHBLD.ODL,FCPNMH.OLB,FCP.OLB
>DMO DK1:
>DEA DK1:
>LBR FCPNMH/RP=WITRN.NMH
>PIP WITRN.NMH;*/DE
```

If it was necessary to modify the task-build command file (FCPNMHBLD.CMD) during the last system generation, it may be necessary at this time to replicate those changes.

```
>ASN SY:=MP:    !if your system supports ASN
>TKB @FCPNMHBLD
>SET /SYSUIC
SYSUIC=[g,m]
>SET /UIC=[g,m]
>PIP /NV=[1,54]FCPNMH.TSK  !or [1,50] for unmapped systems
>VMR
ENTER FILENAME: RSX11M
VMR>REM F11ACP
VMR>INS FCPNMH
VMR>^Z
>RUN $SHUTUP
```

Hardware boot the system.

The following example serves for both the 2.5K file system (FCP) and the 5.5K file system (BIGFCP) by substituting FCP or BIGFCP for every occurrence of XXX in the example.

```
>ALL DK1:
>MOU DK1:SYSTEM
>SET /UIC=[1,24]   !or [1,20] for unmapped systems
>PIP /NV=DK1:XXXBLD.CMD,XXXBLD.ODL,FCP.OLB
>DMO DK1:
>DEA DK1:
>LBR FCP/RP=WTRN1
>PIP WTRN1.OBJ;*/DE
```

If it was necessary to modify the task-build command file (XXXBLD.CMD) during the last system generation, it may be necessary at this time to replicate those changes.

```
>ASN SY:=MP:    !if your system supports ASN
>TKB @XXXBLD
>SET /SYSUIC
SYSUIC=[g,m]
>SET /UIC=[g,m]
>PIP /NV=[1,54]XXX.TSK   !or [1,50] for unmapped systems
>VMR
ENTER FILENAME: RSX11M
VMR>REM F11ACP
VMR>INS XXX
VMR>^Z
>RUN $SHUTUP
```

Hardware boot the system.

1.4.2 **Making Patches to Object Modules** - To make interim changes to object modules, use the Object Module Patch Utility (PAT) to incorporate an object patch into an existing module. To correct an error, DIGITAL publishes a patch file written in MACRO-11 assembly code. Create and assemble the published file, which you then input to PAT together with the relevant object file.

You must include the published checksum values in the specifications of the input file and the correction file. If a checksum value does not agree with the computed result, PAT reports an error. If a correction file causes the error, check the source against the original published source to verify that it has been copied correctly. If the input file causes the checksum error, verify that the correct version of the file is being patched.

Note that it may be necessary to use LBR to extract the original object module from a library. PAT produces a new object module file containing the corrected object code. When patching modules from a library, you should always take the object module from the distributed library and put it in a copy that will accumulate all patches to that library.

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

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

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

- Always specify checksums for the input and correction files.

The following example illustrates the command formats for creating the correction file and the PAT output file:

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

The correction file source input is used to accumulate all patches to a given module. This module should be prepared in the format described in Appendix E of the IAS/RSX-11 MACRO-11 Reference Manual, with the following additions:

- The .TITLE statement must contain the name of the module to be patched.
- the preface to the module should contain a summary of all patches made.
- The module must include a list of equated symbols that defines the initial location counter values for each p-section that contains a patch.
- The module should be subdivided into one or more patch blocks that contain the following information:
  1. An .IDENT statement in the format described in Section E.10 of the IAS/RSX-11 MACRO-11 Reference Manual,
  2. Any equated symbols, local macros, or local data associated with the patch,
  3. A description of the patch code,
  4. The patch code, and
  5. A set of equalities that reset the location counter for each p-section to the initial value.

The following example illustrates this format:

```
.TITLE  NAME
   .
   .

copyright statement
```

```
;
; VERSION 00
;
; MODIFICATIONS:
;
;       03-MAY-77
;
;       00A -- CORRECT SYNCHRONOUS TRAP
;              PROCESSING
;
;       30-JUN-77
;
;       00B -- PROVIDE INTERLOCKING MECHANISM
;              FOR DATA BASE ACCESS
;
        .PSECT
.BLK.=
         PSECT  A
A=.
        .PSECT  B
B=.
        .IDENT  /00A/

           .
           .
        module code

           .
           .
        .PSECT
.=.BLK.
        .PSECT  A
.=A                             ; RESET LOCATION COUNTER
        .PSECT  B
.=B                             ; RESET LOCATION COUNTER
        .IDENT  /00B/

           .
        module code
           .
           .
           .
        .PSECT
.=.BLK.
        .PSECT  A
.=A
        .PSECT  B
.=B

;
; ADD NEXT PATCH HERE
;
        .END
```

See the RSX-11 Utilities Procedures Manual for a complete description of PAT.

1.4.3 **Making Patches to Task Image Files** - Interim changes to task image files may be made by executing an interactive command sequence with the Task/File Patch Utility Program (ZAP). These changes modify the appropriate task image files to make the corrections. Two command sequences are specified, one for each of the distributed task image files.

Virtual disk block 2 is assumed to be the beginning of all task images. In addition, for unmapped task images, 40000 (octal) is assumed to be the base address of the task. If a task to be patched has been rebuilt with checkpoint space allocation or with a nonstandard base address, the ZAP relocation commands must be modified to reflect the differences.

See the RSX-11 Utilities Procedures Manual for a complete description of ZAP.

2.0 EXECUTIVE

2.1 Dynamic UMR Allocation

Dynamic UMR allocation has been implemented for PDP-11/70 systems running in 22-bit mode. Most NPR device drivers that ran on RSX-11M V03 should not be affected by this feature. In particular, the 6-word block still exists at the end of the SCB but it does not require initialization. In addition, the driver calls to the Executive to map the UMRs are identical. However, a driver cannot use the stack to store data prior to calling $STMAP because $STMAP returns as a co-routine. For a detailed description, read Appendix B of the RSX-11M Guide to Writing an I/O Driver.

2.2 User-Written Drivers

Chapter 4 of the RSX-11M Guide to Writing an I/O Driver describes the I/O packet field I.PRM as having read-only access by a device driver. In fact, several DIGITAL-supplied drivers store temporary data into unused portions of this eight-word field. However, you should not use the I/O packet as scratch storage. For example, if a driver completes an I/O request and leaves word I.PRM+16 non-zero, a system crash will occur if RMS record locking was included in the Executive.

2.3 Connect to Interrupt Vector Directive

The new Connect to Interrupt Vector (CINT$) directive is described in detail in the RSX-11M Executive Reference Manual.

For examples of a program using the directive, see the support routines for the K-series Laboratory modules. You should concentrate your attention on the following modules:

KAST.MAC

KDISWP.MAC

KDRDRV.MAC

KGHIST.MAC

These source files are located in UFD [45,10] on the appropriate disk in your distribution kit.

2.4 Clarification in Use of Executive Swapping Option

Chapter 4 of the RSX-11M System Generation Manual explains the swapping option. Users must also consider the following additional factor:

> A task that is checkpointed by a lower priority task after its priority has been lowered by time will be placed into the partition wait queue according to its installed priority; it, therefore, is immediately eligible to to reclaim the memory it just vacated.

Therefore, to obtain maximum benefit and reduce unnecessary disk I/O due to ineffective swapping, use the following guidelines:

- Group all tasks that should swap with each other at the same priority.
- Consider the swapping priority range only within the context of individual swapping groups; use the parameter only as stated in the discussion of hint #1 (section 4.9.5) in the manual.
- Do not overlap the effective priorities of the individual swapping groups.

2.5 Restriction in the Use of Resident Libraries

The error code IE.SPC indicates that a directive's DPB or required buffer was outside of the user task's address space when a system directive was issued. There is an additional instance when IE.SPC can result from issuing a system directive. If the DPB or any buffer is located in a read-only resident library, the address check algorithm will fail and IE.SPC will result.

The problem can be circumvented by linking tasks to the resident library with read-write access, rather than read-only access.

3.0 DEVICE DRIVERS

3.1 LPA11-K Driver (LADRV)

3.1.1 **Restriction With PDP-11/70 Systems** - Due to the extreme flexibility of the LPA11-K microprocessor controller, there are potentially more data buffers in use than there are PDP 11/70 mapping registers (UMRs) to handle them. Therefore, the LPA11-K cannot be used on a PDP-11/70 system in 22-bit addressing mode.

3.1.2 **Caution Required When Unloading Driver** - To attempt to take advantage of the LPA11-K's power and flexibility, the driver was designed never to look busy to the RSX-11M Executive. The major problem with this technique is that a privileged user can Unload the driver while the LPA11-K is servicing one or more users. Therefore, you must exercise caution if you desire to Unload the driver.

3.1.3 **Microcode Loader** - The LPA11-K microcode loader is a privileged task that loads the microcode into the LPA's RAM, initializes the unit, and establishes the real-time clock rate. In addition, there is one task per LPA11-K unit that requests the loader, specifying unit characteristics. The loader taskname is LAINIT; the initiating tasknames are LAINn, where n represents the LPA11-K unit number.

When a powerfail recovery occurs, the LPA11-K driver terminates all outstanding activity and requests execution of the initiating task (LAINn) for each unit. This provides automatic powerfail recovery for the LPA11-K microprocessors provided the LAINn and LAINIT tasks are installed. Note that a simulated powerfail occurs when the system is bootstrapped and when a driver is loaded.

If the request for the initiating task or loader fails, the LPA11-K unit will not be initialized. Any attempt to use the LPA11-K will fail with the error code for device not ready (IE.DNR). To initialize the device manually, the tasks LAINn and LAINIT must be installed. Then enter the command RUN LAINn.

A command file is invoked automatically during SYSGEN2 if LPA11-K support was selected during SYSGEN part 1. Questions are asked to determine the target operating system environment and LPA11-K characteristics. There are additional options available for which questions are not offered; these may be selected by editing the task-build command files at the appropriate time.

3.2 Magnetic Tape Drivers (MTDRV/MMDRV)

To ensure that output tapes are not overwritten accidentally and that improper data is not read unknowingly from an input tape, a new powerfail procedure has been implemented in the magnetic tape drivers. Following a powerfail (or other loss of vacuum), the drivers will reject all user requests until it receives one to rewind to BOT (IO.RWD, IO.RWU, or IO.SMO). This ensures that a read or write does not occur when the tape is not at its proper position. Note that system boot and driver load are a simulated powerfail.

3.3 DMC-11 Driver (XMDRV)

In Version 3.0 of RSX-11M, Unibus Mapping Registers were assigned to devices at sysgen time; each DMC-11 was assigned five (5) UMRs. The DMC-11 driver used one UMR for the base table, two UMRs for active transmits, and two UMRs for active receives. Version 3.1 of RSX-11M, however, assigns UMR's dyamically.

Therefore, the DMC-11 driver will assign its UMRs when the initialize QIO (IO.INL) is processed, and will de-assign its UMRs whenever the device goes off line. An off-line condition may result from a terminate QIO (IO.TRM), a powerfail, or a fatal microprocessor error. Note that failure to assign UMRs will cause the initialize QIO to fail, in which case the IOSB code will be IE.RSU.

For some systems, five UMRs may be too many to dedicate to the DMC-11, especially if there is more than one DMC-11. The DMC-11 driver may be assembled to use only three (3) UMRs instead of five if the symbol XM$UMR is defined and if its value is 3. The special symbol may be defined in RSXMC.MAC or it may be edited into the driver source code (XMDRV.MAC).

3.4 Laboratory K-Series Support Routines

The K-Series support subroutines also support the LPS-11 and the LSI-11 version of the K-Series peripherals. The LPS-11 support does not include the relays or the DMA option.

When performing A/D input sampling with multiple channels or auto-gain ranging, the A/D conversion start is disabled when the interrupt routine is entered. This has the effect of missing A/D conversion starts if the clock overflow or external trigger rate is too fast. If this is undesirable, you may edit the source files [45,10]KADIN1.MAC, KADIN3.MAC, KADIN4.MAC, and KADIN5.MAC and delete the line labeled ".ADnDC::", where n is 1, 3, 4, or 5, corresponding to the module name. You may also use a GBLPAT at task build time and patch the instruction to a no-op (i.e. GBLPAT=segment:.ADnDC:240).

3.5 Additional Instances of the IE.PRI Error Code

Chapters 3 and 4 of the RSX-11M I/O Drivers Reference Manual discuss the IE.PRI error code for disk and DECtape drivers. Two reasons are given for the return of this error: a non-privileged task attempts to perform logical I/O (such as IO.WLB) to a mounted volume; or, a user attempts to attach (IO.ATT) a mounted volume. Here is an additional reason: an attempt to perform virtual block I/O (such as IO.RVB) to an unmounted volume.

Also, under Chapter 5, it is possible to get the IE.PRI error status from the magnetic tape drivers if the system has support for ANSI magnetic tape. In this case, the criteria for the error are the same as used by the disk drivers.

4.0 MACRO

Following are several updates to the IAS/RSX-11 MACRO-11 Reference Manual.

In the discussion of keyword arguments in a macro call (Section 7.3.6) the manual omits the fact that a macro call statement may be constructed such that an argument has been specified both as a positional argument and as a keyword argument. When this occurs, the positional specification takes precedence.

Two symbols were omitted from the Permanent Symbol Table in Appendix C. The symbols STA0 and STB0 are diagnostic floating-point op codes having the values 170005 and 170006, respectively.

In Appendix D, the description of the L assembly error states that it can be obtained only during a macro expansion. This is no longer true. An input line greater than 132 characters will be flagged with the L error; this condiiton no longer results in a fatal I/O error.

In Appendix D, the description of the E assembly error states that it is obtained by omitting a valid .END statement. It is now also possible to receive the E error during expression analysis if the assembler's stack overflows. In addition to flagging the statement with the E error, a question mark (?) will be inserted into the line at the point where the overflow occurred. To avoid this, simplify the expression or build Macro with a larger stack.

The size of the assembler's command line buffer has been increased from 80 to 132 characters. In addition, command input may continue to additional lines provided each line terminates with a hyphen. It should be noted in Appendix G that the subset (8K) assembler will support neither of these features.

5.0 MCR

5.1 Problems Booting an RK06 Based System

It is possible for BOOT to loop indefinitely when booting a system from an RK06. This is the result of encountering an ECC-correctable error when reading the system image. The bootstrap driver in BOOT is not large enough to allow for the inclusion of ECC logic. Therefore, it simply retries the read, which will never succeed.

To recover from this error, use PIP to copy the system image to another area on the disk and try booting the copy.

5.2 Problems Booting a Mapped System From an Unmapped System

It is possible for BOOT to loop indefinitely when booting a mapped system from an unmapped system on a processor with MOS memory. This is the result of random parity bit settings in the parity registers for memory above 28K. The disk controller therefore fails to successfully transfer the system image to that portion of memory and the bootstrap drivers will simply retry the read, which will never succeed.

This problem may be avoided either by running a standalone memory diagnostic before bringing the unmapped system into memory or by using a mapped system to boot the target mapped system.

5.3 Potential Problem With a Saved System

When booting a saved system on a configuration with less memory than the original one, SAV will adjust the length of the last partition (system-controlled only) to reflect the amount of memory actually in the configuration. A problem arises if you have a task installed in a partition that is larger than the new partition size after adjustment by SAV. If activated, the task will not be loaded into memory because the partition is now too small. Furthermore, the task can not be aborted because that also requires the task to be loaded into memory.

To avoid this problem you must manually remove any task that has become too large for its system-controlled partition.

5.4 SAV Relocates Vectors of a Saved System

In a saved system image locations 0-1000 (octal) are relocated by SAV. Because there is no OPEN command in VMR, these locations can only be examined or modified by using ZAP in absolute mode.

The relative address within the file of these locations is stored by SAV in the two words at $SYSIZ+2 and $SYSIZ+4. The first word contains the relative virtual block number and the second contains the byte offset into the block of relocated location 0.

6.0 UTILITIES

6.1 Undocumented DSC Error Message

When DSC attempts to initialize either an input or output magnetic tape, the driver may return an Illegal Function error code (IE.IFC). If so, DSC will display the following message:

```
DSC--*FATAL*40 I/O ERROR F ON MMx:
     ILLEGAL FUNCTION
```

This will occur after a system powerfail, initial boot, driver load, or loss of vacuum.

To recover from this error, you must force DSC to issue a Rewind command to the tape drive indicated in the error message. This can be accomplished by specifying /RW in the DSC command line.

6.2 DEC EDITOR (EDT)

The use of the new EDT text editor is described in the DEC EDITOR Reference Manual. Information specific for RSX-11M V3.1 is provided here. In general, these notes describe EDT invocation and file specification formats. Additional information concerning line sequencing, sequenced listings, and EDT defaults is also included.

6.2.1 EDT Invocation from MCR

In the RSX-11M system, EDT is invoked from the Monitor Console Routine (MCR). The input and output file specifications in the examples in this section follow the RSX-11M conventions (filename.type;ver). All EDT file specifications follow the same conventions as the file specifications in the RSX-11M system.

Example 1:

```
>RUN $EDT <CR>
EDT> [output file specification=] [input file specification] <CR>
* (EDT command level)
```

Example 2:

```
>EDT <CR>
EDT> [output file specification=] [input file specification] <CR>
* (EDT command level)
```

Example 3: (Invocation Command Line variations)

```
>EDT [output file specification=] [input file specification] <CR>
* (EDT command level)

>EDT [input file specification] <CR>
* (EDT command level)
(outfile defaults to infile;n + 1)

>EDT [output file specification=] <CR>
* (EDT command level)
(provide input at terminal)

>EDT <CR>
EDT> <CR>
* (EDT command level)
(provide input at terminal)
```

In this case, EDT will permit entry and editing of text, but will not accept an EXit command. To exit, you must use the EXit/rename option or the WRite command, followed by the QUIT command.

6.2.2 Line Sequencing

The /SEQuence option is not supported in the first release of EDT for RSX-11M V3.1. Line numbers are therefore not part of a file and are present only in the text buffer.

The PRint command produces a listing file with line numbers. Do not attempt to modify this file. Line numbers are part of the text in this file, and any editing of this file will result in an additional set of line numbers at the left margin of a subsequent printout.

6.2.3 EDT Command Defaults

A brief list of EDT commands, with defaults, is included here to indicate the defaults for EDT command lines in which no options (or file specifications, where applicable) have been entered by the user.

Command	Default
Change	current line
COpy	error (must have range)
Delete	current line
EXit	exit with output file or backup file saved
Find	current line
INClude	error (must have filename)
Insert	current line

(continued on next page)

Command	Default
Move	error (must have range)
PRint	error (must have filename)
QUIT	no default -- terminates with all text buffer contents lost
Replace	current line
RESequence	resequence all lines in text buffer
REStore	error (must have filename) Note: file must have been previously SAved
SAve	error (must have filename)
Substitute	error (must have object-string and replacement string)
WRite	error (must have filename)
Type	display current line

6.3 Peripheral Interchange Program (PIP)

6.3.1 PIP Partition Limitation

PIP has logic to use large buffers and perform multiple block virtual I/O when copying disk-to-disk. If PIP is installed or run in a partition 16K or larger, PIP will appear to lose large parts of files copied.

6.3.2 PIP List Switch (/LI)

There is an inconsistency in defaults under the PIP List switch (/LI). Here is an example. If an /LI command is given with all filename fields (filename, type, and version) completely blank, as follows:

```
PIP TEST.FTN/LI,[301,301]
```

you would expect PIP to list TEST.FTN in the current UIC and in UIC [301,301]. For the second UIC, however, PIP defaults to *.*;* and lists TEST.FTN in the current UIC and all filenames in UIC [301,301].

6.4 Line Text Editor (EDI)

Entering either of the following EDI commands, M* or Mn*, will cause EDI to terminate with a MEMORY PROTECT VIOLATION. The fix for this involves more work than has been available recently for EDI maintenance. This problem will be addressed at a later time.

7.0 DIAGNOSTICS

The TE10 and TE16 magnetic tape drives are replacements for the TU10 and TU16, respectively. For the most part, these units are program compatible at the user level. However, the system user-mode diagnostic tasks (TU10.TSK and TU16.TSK) cannot be used on the new hardware. Updated online diagnostics have not been developed for the new hardware.

8.0 MISCELLANEOUS

8.1 VT55 Support

Software support for the VT55 Graphics Display Terminal is included in this release of RSX-11M. The software consists of the following files:

[1,1]PLOT55.OBJ (in SYSLIB.OLB)

[200,200]TEST55.FTN

PLOT55 is a single FORTRAN-callable routine that provides access to the graphics features of the VT55 terminal.

PLOT55 is already incorporated into [1,1]SYSLIB.OLB on the kits, and thus will be automatically linked into your program at task-build time.

To test the PLOT55 package and the VT55 itself, run the test program as follows:

1. Login to [200,200] from a VT55 (Set baud rate as high as possible.)

2. >FOR TEST55,TEST55=TEST55 or

   >F4P TEST55,TEST55=TEST55

3. >TKB TEST55=TEST55

4. >RUN TEST55

The TEST55 program will test the terminal and software and demonstrate the capabilities of the VT55.

8.2 Pseudo Device LB:

A new psuedo-device (LB:) has been added to the system. This device is redirected by SAV to the system boot device. It represents the true system device in that the macro assembler and task builder search device LB: for system libraries. In addition, the MCR commands BOO, LOA, UNL, INS $, RUN $, HELLO, and HELP use device LB:. The VMR command INS also uses device LB:. The utilities SYE, PIP, FLX, F4P, EDT, DMP, and VFY use device LB: for error message text files. Finally, ACNT uses LB: to access the system account file.

This new feature introduces a new level of flexibility to RSX-11M multi-user systems. Now individual users can assign device SY: to a private disk pack without having to replicate all of the system files on that disk.

Note, however, that SAV automatically will remove any task that is not installed from device LB: when the system is bootstrapped. Therefore, tasks that must reside in the system permanently must be installed from device LB:. Users performing target Sysgens must be aware of this and carefully assign LB: to the target device.

8.3 Using RMS with the Dual RK05 Distribution Kit

If you intend to use RMS-11 with the dual RK05 distribution kit, you must copy the appropriate RMS libraries ([1,1]RMSMAC.MLB and [1,1]RMSLIB.OLB) to your SYSTEM disk after your SYSGEN is completed. SYSGEN3.CMD will not do this for you.

8.4 Sample Procedure

The system-building procedure (SYSGEN) makes extensive use of the operating system and its utilities and constitutes the sample procedure as mentioned in DIGITAL's Terms and Conditions of Sale.

Please cut along this line.

READER'S COMMENTS

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

Did you find errors in this manual? If so, specify by page.

_____

_____

_____

_____

_____

_____

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

_____

_____

_____

_____

_____

_____

Is there sufficient documentation on associated system programs required for use of the software described in this manual? If not, what material is missing and where should it be placed?

_____

_____

_____

_____

_____

_____

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

- ☐ Assembly language programmer
- ☐ Higher-level language programmer
- ☐ Occasional programmer (experienced)
- ☐ User with little programming experience
- ☐ Student programmer
- ☐ Non-programmer interested in computer concepts and capabilities

Name _____ Date _____

Organization _____

Street _____

City _____ State _____ Zip Code or Country _____

---------- Fold Here ----------

---------- Do Not Tear - Fold Here and Staple ----------

FIRST CLASS
PERMIT NO. 33
MAYNARD, MASS.

BUSINESS REPLY MAIL
NO POSTAGE STAMP NECESSARY IF MAILED IN THE UNITED STATES

Postage will be paid by:

digital

Software Documentation
146 Main Street ML5-5/E39
Maynard, Massachusetts 01754

digital equipment corporation

Printed in U.S.A.