March 1979

This manual summarizes the RT-11 V03B operating system
features that differentiate it from previous versions of RT-11.
Read this document carefully.

RT-11
System Release Notes

Order No. AA-5286B-TC

digital equipment corporation · maynard, massachusetts

The postage-prepaid READER'S COMMENTS form on the last page of this
document requests the user's critical evaluation to assist us in pre-
paring 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
VAX	VMS	SBI
DECnet	IAS	PDT
DATATRIEVE	TRAX	

UPDATE NOTICE No. 2

RT-11

System Release Notes

AD-5286B-T2

March 1979

Insert this Update Notice page in the manual as a means of maintaining
an up-to-date record of changes to the manual.

NEW AND CHANGED INFORMATION

This update includes information about the DECtape
II device and the PDT-11 processor.

Copyright © 1979 by Digital Equipment Corporation

INSTRUCTIONS

The following pages are to be placed in the RT-11
System Release Notes as replacements for, or addi-
tions to, current pages. The changes made on re-
placement pages are indicated in the outside mar-
gin by change bars (■) for additions and bullets
(●) for deletions.

Old Page	New Page
Title/Copyright	Title/Copyright
v/blank	v/vi
3-9/3-10	3-9/3-10
	3-10.1/blank
4-1/4-2	4-1/4-2
	4-2.1/blank
4-3/4-4	4-3/4-4
4-11/4-12	4-11/4-12
4-12.1/blank	4-12.1/blank
Index-1/Index-6	Index-1/Index-6

CONTENTS

CONTENTS (Cont.)

CONTENTS (Cont.)

CONTENTS (Cont.)

TABLES

CHAPTER 1

INTRODUCTION

This document defines the RT-11 Version 3 (V03) release and its subsequent maintenance release, V03B. RT-11 V03 is a major update from previous releases of RT-11. New capabilities, problem corrections, and a comparison between this and the previous release (V02C) are presented for you in the following sections. The system manager and other responsible system programmers should be thoroughly familiar with the contents of this manual before attempting to generate the system as described in the <u>RT-11 System Generation Manual</u>. The <u>RT-11 Documentation Directory</u> contains a complete listing of the RT-11 manuals.

1.1 SCOPE OF DOCUMENT

This manual is divided into four chapters:

CHAPTER 1 -- INTRODUCTION -- introduces the subject matter and presents the structure and scope of the manual.

CHAPTER 2 -- RT-11 V03 SYSTEM OVERVIEW -- presents a functional overview of the RT-11 system and summarizes the principal characteristics of the new functionality.

CHAPTER 3 -- CHANGES AND RESTRICTIONS -- presents a comparison between the RT-11 V03 release and the previous (V02C) release. It also lists all restrictions and problem corrections for all affected system programs.

CHAPTER 4 -- RT-11 V03B -- describes the new features of RT-11 V03B. It compares RT-11 V03 with RT-11 V03B, and lists all restrictions, problem corrections, and patches to V03B.

1.2 DOCUMENTATION CONVENTIONS

This section describes the symbolic conventions used throughout this manual. Familiarize yourself with these conventions before you continue reading this manual.

1. In examples throughout this manual, information that you must type is underlined, where necessary, to differentiate it from computer output.

2. Unless the manual indicates otherwise, terminate all commands or command strings with a carriage return. Where necessary, this manual uses the symbol (RET) to represent a carriage return, (LF) to represent a line feed, (SP) for a space, (ESC) for an ESCAPE or ALTMODE, and (TAB) to represent a tab.

CHAPTER 2

RT-11 V03 SYSTEM OVERVIEW

The primary goals of RT-11 V03 are to support additional hardware capability, to support new software technology while correcting any existing software problems, and to make the system easier to use for all levels of users. It is still top priority to ensure that RT-11 remains efficient and simple to use and to implement those things that benefit the largest number of users. The following sections provide an overview of the features provided in RT-11 V03 in an attempt to meet these goals.

2.1 EXPANDED SET OF MONITOR COMMANDS

RT-11 has an expanded set of monitor commands based on the DIGITAL command language standard. The new command language allows you to initiate and control system operations through simple, concise English-like commands (for example, COPY, DELETE, EXECUTE, COMPILE). You can type commands in a long command format that prompts you to supply specific information required for the command, such as file names and device names. This interactive prompting format is helpful until you become familiar with the new commands. You will probably prefer to use the short command format where you can enter all required information on a single command line; in this format, you are prompted only if you neglect to supply necessary information. There is even a new HELP command that provides you with on-line reference to the syntax and semantics of the new commands.

The KMON and CSI interface still exists, but you will find the new command language easier and faster to use: you can assemble, link, and run a MACRO program with a single two-word command line!

Included in the new command language is support for indirect command files. You can store a number of keyboard commands in a file and can invoke them with a single command. You can mix old and new format commands in the file and you can set up the file to run unattended or to pause at certain points for operator action. There is even an indirect file that can be invoked at system startup so that system parameters are reset automatically after each system bootstrap.

The new command language has several features that prevent it from becoming too verbose; these features include a minimum unique abbreviation scheme and a character string factoring facility for complicated command lines.

For detailed information on the new command language, see the Introduction to RT-11 and the RT-11 System User's Guide, Chapter 4.

2.2 NEW DOCUMENTATION SET

RT-11 has a new set of user documents, including a simple Introduction to RT-11 manual that will get you up and running on RT-11 in a minimum amount of time. The new documentation falls into three categories: introductory or once-only manuals, computer manuals, and desk manuals.

Once-only manuals are intended to be used once and then stored away. You will have little use for them once your system is in operation and you are familiar with its use.

Computer manuals are intended to be used at the computer. They tell you how to use the computer system, and they describe command usage and syntax, system messages, and general system operation. You will use them for reference when you are using the system.

Desk manuals are intended to be used at your desk. You will use them for reference as you write your own application programs. They contain specific programming information for the RT-11 V03 operating system and its associated programming languages.

See the RT-11 Documentation Directory for more information about specific manuals and for a reading guide to the documentation set.

2.3 NEW SYSTEM PROGRAMS

RT-11 includes several new system programs. The functionality incorporated in RT-11 V02C PIP has been expanded and spread across three system programs: PIP, DUP, and DIR.

2.3.1 PIP (Peripheral Interchange Program)

PIP primarily does file copy, rename, and delete operations. It has been enhanced to support more powerful wildcard and wild character operations and to optionally log on the console all files included in wildcard operations. Complementing the extensive wildcard support is an exclude option that causes PIP to operate on all files except the ones that you specify. PIP has also been enhanced for improved magtape operations and to preserve the date on files across a copy operation. The error messages that PIP prints have also been improved to give you more specific information (file name, etc.) when I/O errors occur.

Several options have been added that allow much more flexible copy operations. PIP now supports a no-replace option that helps prevent accidental deletion of files. If you use this option, PIP will not copy a file if a file with the same name as the output file already exists on the output device. For backing up a file to a device that contains an old copy of the file, there is a predelete option. If you use the option, PIP deletes the old copy of the file prior to transferring the new copy to the device. In case of errors during a copy operation, there is a single block read/write option. If you use this option, PIP does single block reads and writes when copying; on some devices, this increases the chance of an error-free transfer. Finally, there is an option to specify the number of copies of a file to make when transferring files to the line printer, terminal, or paper tape punch.

For detailed information on PIP, see Section 3.5 of this manual and Chapter 7 of the RT-11 System User's Guide.

2.3.2 DUP (Device Utility Program)

DUP is a new system program that includes both new functionality and functionality that formerly was included in PIP. DUP performs volume initialization, bad block scanning, volume compression, file extension, file creation (for recovering deleted files), image copy of devices, and system booting. On volume initialization, DUP supports user-specified volume and owner identifiers that are written on the volume, and it supports automatic covering of bad blocks with .BAD files (on RK06, RL01, and RK07, bad block replacement is supported).

DUP allows booting of an alternate monitor without permanently modifying the hardware bootstrap blocks of the specified volume.

DUP's image copy mode provides fast device backup to a similar device or to magtape. For detailed information on DUP, see Section 3.7 of this manual and Chapter 8 of the RT-11 System User's Guide.

2.3.3 DIR (Directory)

DIR is a new system program that performs directory listing operations formerly done by PIP. DIR includes powerful wildcard, wild character, and sorting capabilities that greatly facilitate directory listing operations. These facilities allow you to list directories sorted either by size, file creation date, position on device, file name, or file type. There is a reverse option so that you can, for example, choose to list a directory sorted in reverse alphabetical order of file name. You can also list all files created before, on, or after a specific date. You can list all files on a device that do not match a specific wildcard construction, such as all files with a file type other than .SYS. You can list directories in a specified number of columns across the terminal or line printer (two columns is the default). You can even list files that have been deleted from a device but whose file information is still intact in the directory, in order to recover accidentally deleted files.

See Section 3.6 of this manual and Chapter 9 of the RT-11 System User's Guide for more detailed information about DIR.

2.3.4 ERRUTL, PSE, SYE

ERRUTL, PSE, and SYE are new system programs that, together with the error log (EL) handler, make up the new, optional, error logging support for RT-11. If you generate an RT-11 system that includes error logging support, you will use these system programs to update your system error file and generate a system error report.

When you use the error log handler (EL.SYS), information on successful and unsuccessful system I/O transfers is collected in a buffer in the EL handler in memory. You use the error utility program (ERRUTL) to transfer this memory buffer to a file. PSE can then use the information in this file to update a cumulative error data file, and SYE can use it to generate a formatted system error report and summary.

See the RT-11 Advanced Programmer's Guide, Chapter 1, for more detailed information on error logging.

2.3.5 MDUP

MDUP is a new system program that is the logical replacement for the RT-11 V02C MBUILD program. You use MDUP when building an RT-11 system from magtape distribution. When you boot the distribution magtape, you select the version of MDUP (for either TM11 or TJU16 magtape) to be run.

MDUP is an abbreviated version of DUP that you can use to initialize your system device (with optional bad block scan), to transfer a minimal system from magtape to your system device, and to boot your system device. The minimal system has enough capability for you to then transfer all desired files from magtape to your system device and begin using the full RT-11 system. MDUP, together with DUP, also provides the capability of building bootable magtapes like the distribution magtape.

For specific uses of MDUP, see the RT-11 System Generation Manual.

2.3.6 PAT (Object Module Patch Utility)

PAT is a new system program that replaces PATCHO for RT-11. As a result of new functionality incorporated in MACRO (.PSECTs and global arithmetic), you cannot easily use PATCHO on object modules produced by RT-11 V03. PAT performs minor modifications to files that are in object format, like the output files produced by the MACRO-11 assembler.

The input for PAT is an object file containing corrections or additions for another object file; PAT applies these corrections and additions to the original file and produces a single updated object file. This process allows you to add relatively large patches to object files without performing complicated octal calculations. You prepare your correction input in MACRO-11 source form and assemble it with the MACRO-11 assembler. Thus, you have all the flexibility that MACRO-11 provides so that you can overlay lines or even add subroutines to the original object file.

See Section 3.12 of this manual and the RT-11 System User's Guide, Chapter 18, for more specific information on PAT.

2.3.7 TECO

TECO is a new system program that is being supplied with RT-11 but is not supported by DIGITAL. TECO is a powerful text editing program that can be used to edit any form of ASCII text such as source programs, listings, manuscripts, or correspondence.

TECO is very versatile, and, as such, complex. If you are a novice user, you must familiarize yourself carefully with the editing commands before using TECO, especially since TECO commands are often similar to EDIT commands in format but are rarely similar in result. Like EDIT, TECO supports VT11 and VT52 display terminals for displaying its text buffer if your system has the appropriate hardware.

TECO has powerful commands for searching for and modifying text and for iterating a complex set of editing commands. TECO also supports commands for conditional execution of commands and has a very flexible system of commands for manipulating text blocks. In addition, TECO

provides mechanisms for you to define editing macros (collections of commonly used editing commands) that you can store away and invoke with a single simple command.

For more detailed information on TECO, see the RT-11 TECO Release Notes and the PDP-11 TECO User's Guide.

2.4 IMPROVED SYSTEM PROGRAMS

2.4.1 Linker

The RT-11 linker is significantly enhanced both in functionality and performance. The entire linking process is speeded up and the .SAV file output process utilizes a disk caching scheme for improved performance. The link map has extra information and can optionally be alphabetized for easier reference. Linker error messages are improved and are more explicit.

The linker now supports complex (global) arithmetic for background jobs; this feature should greatly ease the writing of modular programs.

The linker, together with the new MACRO-11 assembler, also supports .PSECTs. This support provides the MACRO programmer much greater flexibility in designing an application program. Further, it provides significant benefits for overlayed programs since it allows globals defined in an instruction .PSECT to be resolved through the overlay handler, while those in a data .PSECT are resolved directly. Another benefit for overlayed programs is that space is allocated in the overlay handler only for globals in an overlay that are actually referenced outside that overlay.

The MACRO programmer can choose between concatenated and overlayed .PSECTs. In a concatenated .PSECT, the size of the .PSECT is the total size of all storage allocations. In an overlayed .PSECT, the size of the .PSECT is the size of the largest single storage allocation.

The linker has new command options for rounding and extending .PSECTs and for aligning .PSECTs on a specified address boundary. The linker also supports generation of a symbol definition file that can be used for resolving references between modules that are linked separately. See Section 3.9 of this manual and Chapter 11 of the RT-11 System User's Guide for a more detailed description of the linker.

2.4.2 Librarian

The RT-11 librarian is improved and includes the capability of producing user macro libraries for the MACRO-11 assembler. The librarian now has the capability to extract an object module from an object library file. This capability is useful since the linker is not able to link an object module from a library into an overlay region. If you need this capability, you can extract the desired module from the library and explicitly include it in the link operation in the desired overlay region.

Librarian error messages are improved, and the librarian explicitly prompts for globals and modules to be deleted or extracted from a library. See Section 3.10 of this manual and Chapter 12 in the RT-11 System User's Guide for more detailed information on the librarian.

2.4.3 BATCH

A large number of BATCH problems are corrected in RT-11 V03. There is now a provision for users with small system devices (such as diskette) to store some of the system programs on a volume other than the system volume if space on the system volume is critical. If you are a BATCH user, you must recompile your V02C BATCH jobs before using them under RT-11 V03 and you should carefully examine any RT-11 mode jobs in light of the modifications to the command language and the system programs.

See Section 3.4 of this manual and Appendix A in the RT-11 System User's Guide for more detailed information about BATCH.

2.4.4 Editor

The RT-11 text editor is improved, primarily in the area of error detection, reporting, and recovery. The integrity of long editing sessions is significantly enhanced because you now have the tools necessary to recover from previously fatal error conditions. If you accidentally type CTRL/C during an insert, there is no longer an automatic exit from the editor. If you exit via a CTRL/C at the command level, you can reenter the editor with no loss of text from the text buffer. The editor now allows file manipulation operations when an output file fills up, enabling you to open a new output file and continue editing. Editor error messages are more meaningful and more numerous.

For more detailed information about the editor, see Section 3.14 of this manual and Chapter 5 of the RT-11 System User's Guide.

2.4.5 MACRO-11

RT-11 now supports essentially the same MACRO-11 assembler as does RSX-11M, RSX-11D, and IAS. The new MACRO-11 includes a virtual (on disk) user symbol table so that mass storage rather than memory is now the limiting factor on the size of programs that you can assemble. The permanent symbol table for the new MACRO also includes more of the standard PDP-11 instruction mnemonics than were supported by RT-11 V02C MACRO. MACRO now supports complex (global) arithmetic, so you can more easily divide a large program into modules for separate assembly.

A major enhancement from RT-11 V02C is .PSECT support. .PSECTs give you absolute control over the memory allocation of a program at link time because the .PSECT attributes are all passed to the linker. These attributes allow you much more flexibility in designing your application program since the various storage allocations for a given .PSECT can be either concatenated or overlayed. .PSECTs also aid you in writing more modular programs and in partitioning memory more effectively. Overlayed programs benefit from the distinction of data and instruction .PSECTs since globals in a data .PSECT in an overlay segment do not cause extra storage to be allocated in the linker's

overlay handler. Previously, there was no way to prevent storage from being allocated for globals that were in overlays but were not entry points for those overlays.

The MACRO-11 assembler is also enhanced to support user macro libraries and keyword macros. The concept of a user macro library allows you to collect common macros into an installation library without modifying the distributed RT-11 system macro library (SYSMAC). As a result of parallel enhancements to MACRO and the librarian, macro libraries are true library files with a directory at the beginning of the file.

Support of keyword macros should make the creation and use of large multi-argument macros much easier than was possible with the V02C assembler where macros were strictly positional. You need not change your existing user macro definitions and invocations because of this keyword macro support.

For more detailed information about MACRO, see Section 3.21 of this manual, Chapter 10 of the RT-11 System User's Guide, and the PDP-11 MACRO-11 Language Reference Manual.

2.4.6 FILEX

RT-11 FILEX is enhanced to include support for file transfers between RT-11 format media and universal interchange (proposed ANSI format) diskettes. FILEX's error messages are also improved.

For more information about FILEX, see Section 3.13 of this manual and the RT-11 System User's Guide, Chapter 14.

2.4.7 DUMP

DUMP is improved, particularly in the area of magtape support. DUMP can now successfully dump magtapes, even in non-file-structured mode. DUMP identifies tape marks that it encounters when dumping magtape. There is also a new option that, in non-file-structured mode, dumps until it encounters the first tape mark. DUMP's error checking is improved, and DUMP output now has a title consisting of the command string you enter.

See the RT-11 System User's Guide, Chapter 13, for more information about DUMP.

2.4.8 PATCH

PATCH is enhanced in several ways. PATCH now has the capability to print the contents of examined locations interpreted either as ASCII characters or as packed Radix-50 characters. Likewise, PATCH can reset the contents of a location to the ASCII or Radix-50 value of characters that you type.

PATCH has a new checksum capability that allows DIGITAL to publish patches that include checksums and to have these checksums verified when you install the patch. This should eliminate many problems where patches were incorrectly installed or where a previously published required patch was never installed.

Another new facility is a command for modifying a value in the overlay handler or its tables in an overlaid program. This command permits PATCH to update its internal copy of the overlay handler and tables so that subsequent patches to overlay segments are properly interpreted.

See the RT-11 System User's Guide, Chapter 17, for more information about PATCH.

2.5 NEW DEVICE SUPPORT

Device support is enhanced in several ways. There is now a system generation option for error logging; including this option allows transfers to be error logged for DECtape and most disks. An error log (EL) handler is supplied along with system utilities (ERRUTL, PSE, SYE); together, the handlers and utilities provide an error logging subsystem that can collect statistics on successful and unsuccessful I/O transfers, update a cumulative error log data base file, and generate a system error log report.

See Section 4.1.1 for the new V03B devices.

2.5.1 RK06

RT-11 now fully supports the RK06 disk. The device utility program (DUP) can create a bad block replacement table when it initializes an RK06. The RK06 (DM) handler, upon encountering a bad block during an I/O transfer, checks this replacement table and uses the alternate block specified in the table to satisfy the I/O request.

2.5.2 Diskette

The RX01 diskette monitor (DXMNxx.SYS) can now optionally be SYSGENed to support four diskette drives (two controllers).

2.5.3 Magtape

Magtape support is completely redone in RT-11 V03. The primary goals of the device support are better comformance to ANSI standards, better error recovery and reporting, and minimization of handler size for users who do need full file-structure support.

The magtape handlers (MT and MM) consist of two parts: a hardware handler and a file-structure support module. The hardware handler can be used separately to perform read/write, tape positioning, error recovery, and other basic functions that the tape drive supports. You access the hardware handler via a non-file-structured .LOOKUP and with special function (.SPFUN) requests. For users who are short on memory and who do not require full magtape file-structure support, the hardware handler provides basic support. The hardware handler is combined with the file-structure support module to produce the standard file-structured magtape handler. This file-structured handler is larger than the hardware handler and includes extra functionality. The file-structured magtape handler supports file sequence numbers when reading and writing tape; you can specify sequence numbers to define tape position for .LOOKUP and .ENTER.

The error recovery algorithm for read/write errors is modified to increase the possibility of recovering from tape errors. The hardware handler has a much improved error reporting facility that includes qualifying information about the specific error encountered when hard errors are reported back to you.

The magtape handler has a special mechanism for performing directory operations without tying up the USR during long tape searches. Special function requests are provided that are the equivalent of a file-structured .LOOKUP and .ENTER but that are issued after a magtape is opened via a non-file-structured .LOOKUP. This mechanism is available in the hardware handler and is provided mainly for multi-user systems to avoid the possibly long delay (and USR lock) involved in standard magtape .LOOKUP and .ENTER. There are also special function equivalents of .READ and .WRITE for users of the hardware handler. There is a provision for reading and writing blocks of sizes other than 512 bytes, even when the standard file-structured magtape support is used.

RT-11 contains several new SET commands that allow setting of magtape characteristics such as parity, density, and number of tracks. Previously, it was necessary for you to patch the handler to change these characteristics.

RT-11 V02C format magtapes can be read but not written by the RT-11 V03 magtape support.

For more implications of this revised magtape support, refer to Section 3.18.1 of this manual and to the RT-11 Advanced Programmer's Guide, Chapter 1.

2.5.4 Papertape

The papertape reader and punch handlers (PR and PP) are combined into a single handler (PC). This means that papertape support takes up one less device slot in the monitor tables, leaving room for an extra device.

2.5.5 Card Reader

The card reader handler (CR) is modified to use the new .FORK process (see Section 3.3.1). The card reader buffers an entire card and then transfers the card image to the user buffer. In RT-11 V02C, this was all done at priority level 6 and caused some device latency problems. The new .FORK process allows handlers to perform lengthy, non-time-critical tasks like the full card transfer at a lower priority, thereby avoiding latency problems.

2.5.6 Null Handler

The null handler (NL) is a new pseudo-device. This handler is useful when you are debugging and do not need real output. Directing output to the null handler causes the output data to be discarded. On input, the null handler returns an immediate end-of-file.

2.6 SYSGEN (SYSTEM GENERATION)

As in previous versions, RT-11 is distributed in ready-to-use form; the vast majority of users can bootstrap the system and run without a lengthy installation procedure. For those users who do have special hardware or application requirements, however, RT-11 offers the flexibility of an optional system generation procedure that allows the installation to build a system customized for its needs.

This generation procedure (SYSGEN) includes an interactive, explanatory English dialogue, followed by automatic creation of the new system. You can customize the system along three dimensions:

- which services to include in the system
- which peripherals to support in the resulting system
- peculiarities of the hardware configuration

The primary software services that are not supplied in the distributed monitors and thus require special system generation are:

- error logging
- escape sequence support (for terminals)
- multi-terminal support (FB and XM monitors only)
- timer support (SJ monitor only)
- extended memory support

See the RT-11 System Generation Manual for further information.

2.7 NEW MONITOR FEATURES

Monitor functionality is greatly enhanced for RT-11 V03. This section describes the new monitor features available in RT-11.

2.7.1 Extended Memory Support

There is a new extended memory (XM) monitor available as a SYSGEN option that is essentially a superset of the FB monitor. The new monitor supports full 18-bit addressing for I/O and a new set of programmed requests that allow a user program to allocate extended memory and manage the KT11 memory mapping registers. This monitor support can extend the effective program logical address space beyond the 32K word virtual address limit imposed by the PDP-11 16-bit word.

In the RT-11 XM environment, a user program can allocate a large region of contiguous extended memory (above 28K) and, by means of programmed requests, the user program can selectively map portions of its virtual address into contiguous segments of the extended memory region. This support is available to both foreground and background jobs in the XM environment. You can issue programmed requests from extended memory although there are restrictions on requests that define storage areas used by the monitor (for example, .CDFN, .QSET).

I/O is supported within a given job's virtual address space. I/O buffers can even be unmapped once a transfer has been initiated and that portion of the program's address space can be mapped elsewhere until the data from the transfer is required.

Standard RT-11 programs can run unchanged as privileged jobs under the XM monitor but user device handlers must be modified to support 18-bit I/O for use in the extended memory system.

The XM monitor is not distributed on the RT-11 V03B kit. You must use SYSGEN to create the XM monitor file and the XM device handlers.

See the RT-11 Advanced Programmer's Guide, Chapter 3, for a detailed explanation of the extended memory support.

2.7.2 Multi-Terminal Support

The RT-11 FB and XM monitors offer optional support of more than one terminal.   This support is a SYSGEN option;   when you choose multi-terminal support at SYSGEN time, it is integrated into the FB and XM monitors, replacing the single-terminal support modules. Terminal characteristics of the multi-terminal systems appear identical with the single-terminal systems. Some features of this new support include:

1.   Support of wide range of hardware, including:

         KL11
         DL11-A,-B,-C,-D,-E
         DL11-W
         DLV11-E,-F
         DZ11-A,-B
         LPV11

     See Section 4.1.1 for the new devices supported by RT-11 V03B.

2.   Support of console interface and 7 additional terminals using DL11 interfaces or 16 additional terminals using DZ11 multiplexors.

3.   The background console can be moved to any local DL11 interface using a SET command.

4.   A foreground job can share the background console as in V02C, or it can be assigned its own independent terminal at run-time.

5.   Access to the terminals is provided by a new set of programmed requests.

6.   Support for remote lines includes modem and dial-in facilities, but not dial-out.

7.   Terminal characteristics can be modified or examined under program control.

8.   An optional feature detects dropped lines and missing interrupt enable bits, making the system less vulnerable to static discharges and line transients.

9.   Escape sequence support is an option.

10.   An option provides the program with event flags that signal the presence of input in the input ring buffer or an empty output ring buffer.   This reduces scanning overhead in applications using multiple terminals.

For further information on multi-terminal support, see Section 3.8 of this manual and Chapter 1 of the RT-11 Advanced Programmer's Guide.

2.7.3 Escape Sequence Support

RT-11 optionally supports terminal escape sequences. RT-11 can be user generated (through SYSGEN) to support either DEC or ANSI escape sequences. This support can be used for local terminal cursor control or for user-defined applications. When escape sequence support is included in RT-11, you can still choose not to use the support, in which case escape sequences are handled as in RT-11 V02C. If you do use the support, you can either have RT-11 echo escape sequences and not pass them to your program, or it cannot echo them and pass them to you.

When RT-11 passes escape sequences to you, the final character of an escape sequence is treated as a line terminator to allow you to respond immediately to the escape sequence. This means that you do not have to use special-mode terminal input to provide immediate response to an escape sequence. RT-11 flags characters that are part of an escape sequence when it passes them to a user program. Flag bits passed to you indicate whether a character is part of an escape sequence, is a control character embedded within an escape sequence, or is the final character of an escape sequence.

For further information on escape sequences, see Section 3.1.2.5 of this manual and the RT-11 Advanced Programmer's Guide, Chapter 1.

2.7.4 Fork Process

A new fork process is added to the RT-11 monitors. This process is intended for use by device handlers to allow them to perform lengthy non-time-critical tasks at lower priority than device interrupt level. This process is used for the system error logging support since it ensures serial access to the error log handler, and it is helpful in avoiding device latency problems that show up when handlers spend an inordinate amount of time at a high priority level.

For more information on the .FORK macro, see Section 3.3.1 of this manual and Chapter 1 of the RT-11 Advanced Programmer's Guide.

2.7.5 Single-Job Monitor Timer Support

The SJ monitor is enhanced to include optional timer support. If you select this option at system generation (SYSGEN) time, the SJ monitor is configured to include the mark time (.MRKT) and cancel mark time (.CMKT) programmed requests plus full support for the .FORK process. The timer support fills a significant gap in the SJ monitor functionality. (Timer support includes date rollover.) Previously, many users switched to the FB monitor just to get timer support even though they were not doing a foreground/background application. For more information on SJ timer support, see Section 3.2.1 of the RT-11 System Generation Manual.

2.7.6 Device Handler Format

The device handler format changed slightly due to the addition of a system generation process and many new handler options. Changes are implemented through a new set of system macros for device handlers that should make conversion easier. The new handler format makes for fast and easy reconfiguration of monitor device support. In earlier

versions of RT-11, adding or deleting devices required you to
permanently patch the monitor file in several places. Now, the new
handler format allows dynamic device configuration via the keyboard
monitor INSTALL and REMOVE commands. You can REMOVE an unneeded
device from the system configuration and INSTALL a new handler for use
during a brief debugging session with no permanent change to the
monitor. You can effectively make these commands permanent by
including them in the standard system startup indirect command file.

The system macros supplied for handlers set up the handler header, the
interrupt and abort entry points, I/O completion, and handler end
code. These macros, together with new system routines and programmed
requests, allow handlers to take advantage of enhanced monitor support
including: multi-vectored handlers, INSTALL and REMOVE for handlers,
optional error logging, optional device time-out, extended memory
support and .FORK processing. The multi-vector support, for example,
allows a single handler (PC.SYS) to support both the high-speed
papertape reader and punch (formerly PR.SYS and PP.SYS were needed).
Multi-vector support also makes it easier to support multi-controller
devices (such as two controllers handling four diskette drives).

New monitor subroutines, along with the general handler macros,
simplify modification of handlers to support full 18-bit addressing
for I/O in the extended memory environment; the monitor subroutines
perform all necessary memory mapping for both NPR (Non-processor
request) and programmed transfer devices. The fork process allows
handlers to execute lengthy but non-critical interrupt handling code
(for example, card image transfer to a user's buffer), as a
serialized, priority 0 system process. This support is similar to
that provided in other PDP-11 operating systems and can help avoid
device latency problems caused by handlers staying at a high priority
for long periods of time.

For more information on device handler format, see Section 3.3.1 of
this manual and Chapter 1 of the RT-11 Advanced Programmer's Guide.

2.7.7 **New Error Message Format**

RT-11 has a new standard error message format. System error messages
now include the name of the system component that generates the
message (PIP, DUP, etc.), the severity of the message, and an
English-language message, rather than a cryptic abbreviation. All the
monitors now include a message to report the occurrence of a power
fail, once power is restored.

The single-job monitor includes a stack overflow message, and it no
longer halts if a system device write-lock error occurs. Rather, the
system prints a message and returns control to KMON where execution
can continue. For more information on error reporting and recovery,
see the RT-11 System Message Manual.

2.7.8 **Special Device Interface**

RT-11 has improved monitor support for special devices. A mechanism
now exists for special devices to receive a file length specification
for .ENTER and to return a file length for .LOOKUP. Previously, this
information could not be communicated between user programs and
special device handlers. The error reporting facility for special
device handlers is also improved. For more information on the special
device interface, see Section 3.1.3 of this manual.

2.7.9 New Programmed Requests

In addition to the multi-terminal and extended memory support
programmed requests, several new programmed requests exist in RT-11.
An .SCCA request intercepts single or double CTRL/Cs typed at the
console. For example, RT-11 BASIC-11 now uses CTRL/C to stop a
runaway BASIC program without actually aborting BASIC. A new
.UNPROTECT requests exists in the RT-11 FB and XM monitors to
complement the .PROTECT request. All monitors include a new .GVAL
request that is used to pick up the contents of a monitor fixed offset
location and a .GTLIN request that is used to collect a line of input
from either the console terminal or from an indirect command file, if
one is active. The .DEVICE request supports an optional linked list
format that allows successive requests to be concatenated instead of
being overlaid. For more information on programmed requests, see
Section 3.1.2.1 of this manual and Chapter 2 of the RT-11 Advanced
Programmer's Guide.

2.7.10 Memory Parity

All RT-11 monitors have optional memory parity support. Monitors that
you generate to include both memory parity and error logging support
log all memory parity errors that the system detects. Support for
cache memory is available. The system records recoverable cache
errors, as well as nonrecoverable backing store errors, in the error
log. For more information on memory parity, see Section 3.1.2.3 of
this manual.

CHAPTER 3

CHANGES AND RESTRICTIONS

This chapter provides you with all of the changes, restrictions, and problem corrections for all of the RT-11 V03 system components.

3.1 MONITOR

This section provides you with all the changes, problem corrections, and restrictions for the RT-11 V03 monitors.

3.1.1 KMON and USR Changes

1. RT-11 includes an entirely new monitor command set. This new command language consists of English-language commands (COPY, DELETE, EXECUTE, etc.) and options (NOREPLACE, BINARY, QUERY, LIBRARY, etc.). In general, a single command is all that you need to execute a single function (copy a file, delete a file, assemble a file). The details of the new language are too numerous to explain here. You should carefully read the Introduction to RT-11 and Chapter 4 of the RT-11 System User's Guide to familiarize yourself with these new commands.

NOTE

The old keyboard monitor commands and the Command String Interpreter (CSI) interface still exist, but several significant differences exist between the CSI interface and the new command language. At CSI level, input files are on the right side of the command string and the output files are on the left. In the new COPY command, for example, input files are on the left side of the command string and output files are on the right, for a more logical left to right command operation. Therefore, you should use the prompting format of the new commands until you become accustomed to this change.

Further, the new command language does not support "sticky" device names. At CSI level, the construction:

    DT:A.MAC,B.MAC

is equivalent to

        DT:A.MAC,DT:B.MAC

The new command language does not propagate devices across
commas in this manner; so, for example:

        DELETE DT:A.MAC,*.SAV

is equivalent to

        DELETE DT:A.MAC,DK:*.SAV

not to

        DELETE DT:A.MAC,DT:*.SAV

To achieve sticky device names at the KMON level, you must
use factoring. Thus,

        DELETE DT:(A.MAC,*.SAV,B.OBJ)

is equivalent to

        DELETE DT:A.MAC,DT:*.SAV,DT:B.OBJ

The factoring facility is very useful for complicated
commands   since   it is basically a string replacement
facility. For example, a command like:

        COPY DT:FILE(1,2,3).MAC RK:*.*

is equivalent to

        COPY DT:FILE1.MAC,DT:FILE2.MAC,DT:FILE3.MAC RK:*.*

To ease your transition to the command language, use the
interactive prompting built into the command language and
the HELP command which   provides   on-line   access   to
descriptions of command syntax and semantics.

2.   The ASSIGN command is now two separate commands: ASSIGN and
      DEASSIGN.   The ASSIGN command requires a space between the
      physical device name and the logical name.

3.   The RT-11 V02C INIT command is renamed RESET.   The
      INITIALIZE command performs volume initialization.

4.   The LOAD, UNLOAD, Examine, and Deposit commands now perform
      stricter syntax checking; valid separators for these
      commands are the equal sign and comma.

5.   The FRUN, RUN, GET, and SAVE commands now automatically LOAD
      a device handler if the specified device is not the same
      type as the system device. (This is so that KMON no longer
      needs to reserve space for the largest possible device
      handler to be .FETCHed internally to handle these cases.)
      Thus, you no longer need to manually LOAD the handler for an
      overlaid program that is RUN or FRUN from a device other
      than the system device.

6.   The Command String Interpreter and the FRUN and GT ON
      commands no longer accept exclamation mark as a decimal
      radix indicator in options (for example, /S!19). The new

monitor commands use exclamation mark as a comment delimiter, and they use decimal point to indicate decimal radix (for example, /S:19.).

7.   The FRUN command changed slightly. In addition, the REL file format is changed to correct several deficiencies. See Section 3.9 of this manual for details.

8.   When you invoke the Command String Interpreter in special mode (.CSISPC), it now accepts % and * wildcards embedded in file specifications. Formerly, only * was legal, and it could only replace either the entire file name or the file type; it could not be embedded within either.

   If you embed either % or * in a file specification, the system translates it and passes it to the user program. The system translates % to Radix-50 code 34 octal (.) and * to Radix-50 code 35 (undefined). If you do an .ENTER with one of these Radix-50 codes still embedded in the file specification, the resulting file name appears to have "." embedded where you specified "%" and either "/" or "*" embedded where you specified "*", depending on whether PIP or DIR lists the file name. You should delete or rename such a file (generally possible only by doing a wildcard delete or rename and using query mode).

9.   A number of new keyboard monitor SET commands allow you to set:

   ● magtape parity, density, and number of tasks
   ● default system editor (EDIT or TECO)
   ● level of error that causes indirect command file termination (NONE, WARNING, ERROR, SEVERE ERROR)
   ● unit number of the console terminal in a multi-terminal system (CONSOL=n)
   ● echo or no echo of indirect command file lines on the console (TT QUIET, NOQUIET)
   ● explicit or implicit wildcards in keyboard monitor commands (WILDCARDS EXPLICIT, IMPLICIT)

   See Table 4-13 in the RT-11 System User's Guide for more information.

10. The .QSET programmed request resides in the USR buffer for all RT-11 monitors. In addition, in the SJ monitor only, the .CDFN, .SRESET, and .HRESET requests also reside in the USR buffer. As a result, whenever these requests are issued, a fresh copy of the USR must be read into memory, even if the USR is set to NOSWAP.

11. The V02C RSUME command is renamed RESUME.

12. The RUN command accepts an optional argument or an optional input and output specification list. Thus, in a single keyboard monitor command line, you can invoke a user application program and give it a command string to execute.

13. The V02C SET TTY command is changed to SET TT to be consistent with other device-oriented SET commands.

14. Old volumes from V02C give warning messages when you use the SQUEEZE command or DUP program to compress them. Use the INITIALIZE/VOLUME:ONLY command to change the volume identification of the disk and eliminate the warning message. Note that the squeeze operation executes properly in both cases.

15. If an error occurs while you are performing a SQUEEZE or COPY/DEVICE command, it indicates one of the following conditions:

     a. If it is a directory error, the output device is not reliable.

     b. If it is a read/write error, data copying reverts to one block at a time. Hence, if the disk is full, the command takes 10 or 20 times longer than the normal command execution time. Do not stop the command because the output device will not be reliable.

     c. If only one error message appears, the output device should be normal. It is always good practice to check out your output devices before proceeding with file manipulation operations.

16. The MACRO/ENABL:ABS command causes a default type of .OBJ instead of .LDA.

3.1.2 RMON Changes

The following sections detail the changes in programmed requests and resident monitor functionality.

3.1.2.1 **Programmed Requests** -- Table 3-1 summarizes the new and changed programmed requests. Details on multi-terminal, device handler, and extended memory requests are not included here. See the RT-11 Advanced Programmer's Guide for detailed information on programmed requests.

Table 3-1
New and Changed Programmed Requests

Request	Explanation
.CSIGEN and .CSISPC	Changed to optionally return a copy of the command line that you typed and to accept input from an indirect command file if one is active. V02C programs that use .CSIGEN or .CSISPC with console terminal input automatically work with indirect command files. You need make no programming changes unless you want to get a copy of the command line that you typed.
.DATE	Changed to use an EMT instruction. .DATE is compatible at the source program level with the V02C .DATE request.
.DEVICE	Extended to support a linked list structure. Existing programs still function using the single list format. New programs can utilize the linked list format by specifying an additional argument (FB, XM only).

(continued on next page)

Table 3-1 (Cont.)
New and Changed Programmed Requests

Request	Explanation
.EXIT	Modified to optionally pass commands or an indirect command file to the keyboard monitor at the program's exit.
.GTJB	Changed to return an additional argument in one of the formerly unused fields of the parameter block. Word 5 of the supplied parameter block is now filled with the address of the job's impure area (FB, XM only). No programming changes are necessary for this addition.
.GTLIN	New request that collects an input line from the console terminal or from an indirect command file, if one is active. An optional argument in the request is a prompt string to be printed on the console terminal prior to collecting the line. You can suppress the prompt string by using SET TT QUIET if input is from an indirect file.
.GVAL	New request that returns the contents of a monitor fixed offset to the program through the EMT interface without directly referencing the monitor region. Supported in all monitors; its use is recommended in all programs. Its use is essential in the XM monitor when a program is running in a protected (virtual memory) environment.
.QSET ·	Changed to require 10 words per queue element when running under the XM monitor.
.SCCA	New request that inhibits the action of single and double CTRL/C characters. Also sets an event flag supplied by the program when a double CTRL/C is struck.
.SYNCH	Extended to optionally produce position independent code; useful when the macro is invoked from a device handler.
.UNPROTECT	New request that complements the .PROTECT request (FB, XM only).

3.1.2.2 **Optional KW11-P Clock Support** -- The monitor can now optionally support the KW11-P programmable clock as the system clock. In addition, the bootstrap checks for the existence of the KW11-P and sets a bit in the CONFIG monitor fixed offset (300) if the hardware is present and is not being used by the monitor as the system clock. See the RT-11 System Generation Manual for instructions on including KW11-P support in RT-11.

3.1.2.3 **Memory Parity** -- RT-11 can now optionally support memory parity. If a memory parity error occurs, the system prints an error message. Additionally, the system logs the error if you select the error logging option at SYSGEN and the error log handler is in memory. The parity error routine distinguishes between recoverable cache memory errors and main memory errors. Only main memory errors abort a job, but the system logs both types of errors. See the RT-11 System Generation Manual for instructions on including memory parity support in RT-11.

3.1.2.4 **User Error Byte** -- A user error byte (byte 53) exists that your application program can set to report to the monitor errors you detect. The monitor uses this byte to determine if indirect command files should be aborted. System programs use this byte to report errors so that execution does not continue after serious assembly or link errors occur. Continuing execution can cause problems if subsequent commands in an indirect file depend on successful completion of all prior commands. Errors reported vary from warning to severe error, aNd new keyboard monitor SET ERROR command governs the level of error at which execution is aborted. MACRO assembly errors set the WARNING bit of the error byte. When this bit is set, an EXECUTE command or an indirect command file does not abort on assembly errors. You can use the SET ERROR WARNING command to force an abort when assembly errors occur. In an indirect file, use the SET ERROR command as frequently as necessary to vary the error abort severity level to suit your needs. For more information, see Section 2.2.4.3 of the RT-11 Advanced Programmer's Guide.

3.1.2.5 **Escape Sequence Support** -- Escape sequence support is a system generation option. You can choose either DIGITAL or ANSI standard escape sequence support, but not both.

Features

Escape sequence support is provided by the user or application program setting the following bits in the Job Status Word (JSW).

ESCSQ$ (bit 4)	This bit must be set to enable any support. If not set, the same support is provided as in V02C.
ESCFL$ (bit 5)	This bit is set to specify that escape sequences are to be echoed but not passed to the program. If this bit is not set, escape sequences are passed to the user program but are not echoed. Regardless of the state of the ESCFL$ bit, escape sequences are not echoed if the program is doing special mode terminal I/O (specified by setting JSW bit 12).

The final character of an escape sequence on input is treated as an end-of-line character. This permits a program to respond immediately to escape sequences without having to use special mode terminal I/O.

If ESCSQ$ is set in the JSW, escape sequence status bits are set in R0 upon return from a terminal input EMT (.TTYIN, .TTINR).

ES.INP (bit 15)	If this bit is set, it implies that an escape sequence is in progress.

ES.CC (bit 14)   If this bit is set, it implies that this character is a control character (0-37 and 177) and is not part of the escape sequence in progress (set only if bit 15 is set). Control characters embedded in an escape sequence are not considered part of the sequence but are processed in whatever capacity the control character serves.

ES.FIN (bit 13)   If this bit is set, it indicates that this character is the final character of an escape sequence (set only if ES.INP is set and ES.CC is clear).

The primary difference between DIGITAL and ANSI support is that the sequence "escape left-bracket ([)" is considered to be a complete escape sequence in DIGITAL mode, whereas it is the beginning of a sequence in ANSI mode.

Restrictions

The following are restrictions on escape sequence support:

1.   If a program is outputting an escape sequence, keyboard input cannot be echoed. Echoing such input would cause random characters to be inserted in the middle of the escape sequence being output. The keyboard input character is inserted into the input ring buffer if there is room in the buffer and discarded if there is not.

2.   If ESCFL$ is set, user program terminal output is inhibited while an escape sequence is being entered from the keyboard. Thus, if you type ⟨ESC⟩ while the program is sending output to the terminal, the output stops until you type a valid escape sequence terminator.

See the RT-11 System Generation Manual for instructions on including escape sequence support in RT-11.

3.1.3 Monitor Problems Corrected

The following monitor problems that existed in RT-11 V02C are corrected in the RT-11 V03 monitors.

1.   The monitor checks for attempts to queue more than 255 requests to a single channel. If you attempt this procedure, the monitor waits until the count of outstanding I/O requests on the channel drops below 255 before queueing another request on that channel.

2.   Several race conditions in the monitor are eliminated. One of these conditions caused an occasional failure to schedule a foreground job waiting for I/O immediately after I/O completed.

3.   The .EXIT request clears the .SERR flag, restoring hard error processing on exit.

4.   A problem is corrected in the .READ/.WRITE code that prevented doing I/O to files larger than 32,767 blocks.

5.   The TIME command is modified to prevent an undesired date rollover when you enter the time.

6.   Several SJ console terminal problems are corrected. It is possible to print more than 32,767 characters without a line feed.

7.   Several VT11 scroller problems are corrected.

8.   The message channel I/O count is properly initialized after a job exits or aborts in an FB environment.

9.   The special device interface is improved by correction of several problems and extension of functionality.

   a.   .RENAME is passed to special device handlers. If the handler cannot process a .RENAME, it must return an error.

   b.   The size of the file requested on an .ENTER request is passed to all special device handlers. The handler must return the size of the file actually created to the monitor, which returns it to you in register R0. The file size data is transmitted to and from the special device handler in Word 6 of the queue element. This corrects the problem of R0 containing erroneous information on return from a special .LOOKUP or .ENTER.

   c.   The SPUSR monitor fixed offset (272) is redefined to permit special device handlers to return meaningful error information. Previously, SPUSR was set non-zero to indicate that an error occurred, but the specific error condition was not available to the monitor for return to the calling program. Now the special device handler must return a specific error code in SPUSR.

   The following error codes are reported on a .LOOKUP:

         0   channel already open (existing)
         1   file not found (existing)
         2   device in use (new)

   Error codes for .ENTER are:

         0   channel in use (existing)
         1   no hole of size M, or device full (existing)
         2   device in use (new)
         3   file exists, cannot be deleted (new)

   Error codes for .RENAME:

         0   channel already open (existing)
         1   file not found (existing)
         2   illegal request for this device (new)

   d.   The .READ/.WRITE interface now additionally checks for end-of-file (EOF) on return from the device handler. Previously, end-of-file was checked prior to calling the handler but not on return. This required an extra .READ beyond end-of-file on sequential devices such as magtape and cassette. The additional check now detects EOF as soon as the EOF record is read and improves the appearance of device independence within the system. However, the change can affect other sequential devices

such as papertape. All affected handlers for sequential devices are changed to report EOF on the next read request rather than on the request during which the EOF (actually end-of-medium) condition occurred.

10. A problem is corrected that caused parity errors on incoming characters to be printed as control characters when using the DL11-E.

11. A problem is corrected that caused incorrect results when SAVEing a program that overlayed KMON.

12. A problem is corrected that caused incorrect results when examining locations in a program that overlayed KMON.

13. A problem is corrected that caused subsequent references to an overlay to fail when a .SRESET was issued in a foreground program.

14. A problem is corrected that caused the system to hang when you entered more than 81 characters per line to the keyboard monitor.

15. The UNLOAD command failed to check for a null (non-interrupt device) vector in the handler being unloaded. This resulted in locations 0 and 2 being erroneously cleared. This problem is corrected.

16. A problem is corrected that caused console terminal output to be garbled if a program used both .PRINT and .WRITE requests for terminal output.

17. A problem is corrected that prevented characters in the terminal output buffer of a foreground job from being printed if a background job was sending output to the console using the FORTRAN OTS routines or if there was no background job running.

18. A problem is corrected that caused midnight rollover for the FB monitor to malfunction when the line frequency was 50Hz.

19. A problem is corrected that existed with the system console handler when GT ON was active. When you transmitted more than 16K characters, the handler transmitted 16K fill characters.

20. In the SJ monitor, an error message is now generated if you attempt to run a program requiring a device write on a write-locked system device. SJ no longer halts in this condition.

21. The .SPFUN request in the SJ monitor no longer tests the wrong bit in the $STAT table.

22. The RT-11 boot message was moved to avoid its being destroyed when you install patches at the beginning of the bootstrap.

23. The UNLOAD command correctly unprotects vectors that were protected by a LOAD command.

24. An error message is generated in the SJ monitor if stack overflow occurs.

25. A problem in the SJ monitor's USR swapping code is corrected.

26. The SJ monitor correctly handles FIS exception errors.

3.1.4 Monitor Restrictions

The following restrictions are applicable to the RT-11 monitors:

1. Extended Memory Support

    a. In the XM monitor, the area supplied for extra queue elements or channels (.QSET and .CDFN) must lie in the lower 28K words of memory. In addition, it must not be in the area mapped by kernel PAR1, specifically the area from 20000 to 37776 (octal). If you supply an illegal area, the system generates an error message.

    b. Interrupt service routines must lie in the lower 28K words of memory. They must not be located in the area mapped by kernel PAR1 -- the area from 20000 to 37776 (octal).

    c. In the XM monitor, you can load virtual background jobs only with the R command. Consequently, they must be resident on the system device, SY:. You can load virtual foreground jobs from any device by using the FRUN command.

2. Graphics Support

    a. Minimal support was added to the scroller for VS60 hardware. No conversion was done to VTHDLR to support the VS60. VTHDLR only supports VT11 hardware.

    b. If an indirect command file issues a GT ON command, part of the command may echo on the terminal and the rest echoes on the graphics screen.

3. Message Handling

    Message handling in the FB monitor does not check for a word count of 0 before queueing a send or receive data request. Since RT-11 distinguishes a send from a receive by complementing the word count, a .SDATW of 0 words is treated as a .RCVDW of 0 words. Avoid a word count of 0 at all times when using .SDATW and .RCVDW.

4. Keyboard Commands

    In the LIBRARY command, the /MACRO and /LIST options can only be on the command side of the command string and not on the file specification. For example, the command:

        LIBRARY/MACRO filnam

    is valid, but the following command is not.
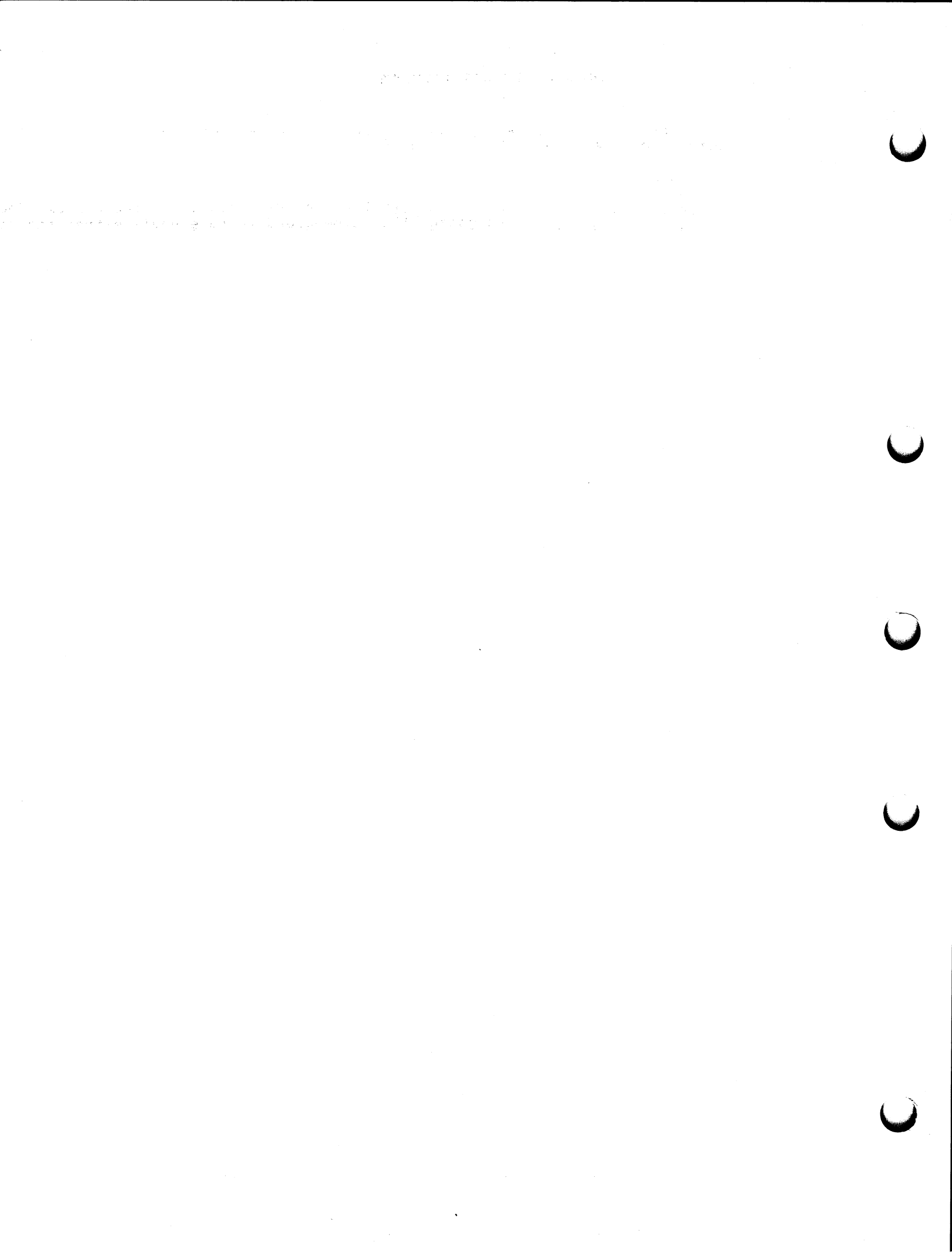
        LIBRARY filnam/MACRO

March 1979

In other commands, these options can appear on either the command or the file specification.

5. Type-Ahead

After you bootstrap any RT-11 monitor, the system does not recognize the type-ahead feature until either the keyboard

March 1979

prompting character (.) prints or the startup indirect command file begins executing. If you type ahead prior to this, the system either ignores or truncates your type-ahead. You can type ahead while the startup command file is executing.

6. CTRL/O in SJ Monitor

   When you are using the CTRL/O feature to suppress terminal output in the SJ monitor, the system can print an extraneous character after the monitor echoes the CTRL/O and a carriage return/line feed.

7. GT ON

   If you type the GT ON command followed by CTRL/E, the initial line on the terminal overprints when you type GT OFF.

8. CTRL/Z

   When you type double CTRL/Zs as a type-ahead to the console terminal, the system crashes.

9. LOAD

   When you type LOAD LP=FB, an error message appears, but the system loads LP and assigns it to the foreground job.

3.2 BOOTSTRAP

This section provides you with all changes, problem corrections, and restrictions for the RT-11 V03 bootstraps.

3.2.1 Bootstrap Changes

The RT-11 V03 bootstrap includes the following changes:

1. The bootstrap is now five blocks in size, and it occupies absolute blocks 0, 2, 3, 4, and 5 of the system device.

2. The XM bootstrap zeroes memory between 8K words and 124K words for operation on systems with MOS memory.

3. The bootstrap now searches only for device handlers associated with device names in the permanent name table ($PNAME). If it finds the device name in $PNAME, it checks the device SYSGEN options against the monitor SYSGEN options. If the options match, the bootstrap updates the $STAT and $DVREC tables in the resident monitor in memory. If the handler is not found, or if its SYSGEN options do not match the monitor options, the bootstrap zeroes the permanent name table entry for that device so that it cannot mistakenly be used by the system. The exceptions to this rule are the TT and BA handlers, whose names must be present at all times in the $PNAME table.

4. The bootstrap program checks for the following new hardware options:

> KW11-P
> EIS
> PDP-11/60 or PDP-11/70 processor
> parity memory
> cache memory
> light register
> switch register
> VS60 display
> KT11 memory management

5. If you do a software bootstrap through the BOOT command or the DUP bootstrap option (/O), the bootstrap program sets up date and time. If you use a hardware bootstrap, the bootstrap does not set up date and time.

6. The bootstrap executes an indirect file (STARTS.COM, STARTF.COM, or STARTX.COM) on exit to the monitor, according to the monitor type. This is a standard feature, but you can delete it with SYSGEN. You can also modify these files for your system configuration (for example, add a SET TT SCOPE command to your startup file if you have a video terminal).

7. The bootstrap initializes and enables all terminals when booting a system with multi-terminal support. It also does a time-out on each controller that you specify at SYSGEN and deletes from the system any terminal that it does not find.

8. The monitor file name is changed. The bootstrap no longer looks for a MONITR.SYS file name. Now it looks for the particular monitor file specified in the COPY/BOOT command. This name must follow the standard RT-11 naming conventions for monitors. That is, it must be of the form ddMNmm.SYS, where dd is the device name (for example, RK) and mm is the monitor name (for example, SJ). For example, the file name for the SJ monitor using the RK05 as a system device is RKMNSJ.SYS.

3.2.2 Bootstrap Problems Corrected

A problem is corrected that occasionally caused a crash rather than a bootstrap error message when the system being used had insufficient memory.

3.3 DEVICE HANDLERS

The information in this section is not device specific and pertains to all RT-11 supported devices. In addition, this section provides you with the changes made in RT-11 device handler interface architecture.

3.3.1 Device Handler Changes

There are several significant extensions to device handler support. All user-written device handlers require some changes to convert to the new format. To facilitate this conversion, the system macro library (SYSMAC) provides a set of device handler macros.

1. RT-11 handlers can support devices with multiple vectors. The addition of this support is transparent to existing devices and does not require conversion. Some RT-11 handlers are modified to use this new function. The DX and DY handlers optionally support a second controller. A single PC handler replaces the PR and PP handlers, thus saving a slot in the device tables.

2. The $DVSIZ and $HSIZ tables in the USR are eliminated. The device block size and handler word size must now be supplied in block 0 of the handler file. Word 52 must contain the handler size in bytes and word 54 must contain the number of 256-word blocks contained in the device. In addition, word 56 must contain the device status word from which the bootstrap updates the $STAT table on every boot. Word 60 contains the system generation options bit mask. For handlers not using any of the new features, this word should be 0.

    The new format requires an extra read operation when doing a .FETCH or .DSTATUS to read the necessary information from the handler file; however, it provides many advantages. The handler file now contains all relevant device information that is automatically updated whenever the handler is assembled. This permits the addition of the INSTALL and REMOVE commands, which greatly simplify the installation of a new device handler. It also eliminates the need for user calculations when the size of a handler changes.

    See the RT-11 Advanced Programmer's Guide for complete conversion instructions for V02C handlers.

3. A new process, called a fork process, is available to device handlers through the .FORK system macro. The fork process allows a handler with a lengthy but non-time-critical task to perform that task at a lower priority. Fork requests are placed in a serial, first in-first out (FIFO) queue. The fork queue manager is entered after the last interrupt has exited but before going to the scheduler. Control returns to the handler at priority level 0, in system state.

    The fork process is useful in device handlers such as the card reader (CR) handler. The CR handler internally buffers a card image, then converts and transmits the 80-character image to the user buffer when the entire card is read. The conversion and transmission are done at fork level to avoid device latency problems since the CR handler normally runs at priority level 6.

    Another use of .FORK is in the addition of error logging to device handlers. The error logger, by its nature, is not reentrant. Since fork processes are serialized, you can avoid reentrancy problems by requiring all device handlers doing error logging to call the error logger at fork level.

NOTE

> The full fork process is standard in the
> FB and XM monitors but is optional in
> the SJ monitor. The distributed SJ
> monitor supports the fork system macro,
> but a handler's fork task is done when
> the .FORK is issued, not after the last
> interrupt has exited. You can include
> full SJ fork processing by selecting
> timer support at SYSGEN time.

4.   The XM monitor contains a number of monitor service routines
     to allow device handlers to support full 18-bit address I/O.
     There are routines for both NPR (Non-processor request) and
     programmed transfer devices. The use of these common monitor
     routines simplifies address conversion and reduces overall
     system size in the extended memory environment.

5.   RT-11 optionally permits device handlers to do time-outs on
     I/O transfers. SYSGEN generates this support by defining the
     symbol TIM$IT=1. Access the support through the .TIMIO and
     .CTIMIO macros. RT-11 device handlers do not use this
     support, but the multi-terminal service time-out option does.
     Therefore, the symbol TIM$IT is defined when you generate a
     system and when you select remote DZ11 support.

6.   Device handlers used with the XM monitor must handle 18-bit
     memory addresses; consequently they are different from the
     set of handlers used with SJ and FB. To allow the two sets
     of handlers to coexist on the same disk, the XM handlers have
     three-letter file names, the usual two-letter names followed
     by an X (for example, RK.SYS becomes RKX.SYS in the XM
     environment. The bootstrap (and the INSTALL and SET
     commands) looks for the appropriate set of handlers,
     depending upon monitor type. Note that device handlers for
     use with the XM monitor are not part of the RT-11
     distribution kit. You must perform a SYSGEN to create these
     files.

7.   The addition of a system generation procedure and the
     existence of several options that affect device handlers
     required the addition of several safeguards to prevent
     mismatched monitors and handlers.

     It is essential that the monitors and handlers be assembled
     with the same option file. To prevent possible errors, the
     bootstrap and the INSTALL command compare the handler options
     (contained in word 60 of block 0) with the monitor options
     (low byte of SYSGEN word at monitor fixed offset 372). The
     bootstrap deletes a mismatched handler from the system
     permanent name table ($PNAME). INSTALL rejects attempts to
     reinstall the handler. In addition, the .FETCH request
     checks the option words.

See the RT-11 Advanced Programmer's Guide, Chapter 1, for complete
details on the device handler format changes and on how to make the
conversion to V03 format.

3.3.2 Device Structure Changes

The RT-11 device structure is modified slightly by defining new uses for previously reserved areas. The changes are fully compatible with V02 format, and devices and directories are interchangeable in both directions. However, some system programs (such as DUP) print a warning message when you perform certain functions on a volume that does not have the RT-11 V03 device structure modifications. Use the INITIALIZE/VOLUMEID:ONLY command to write a V03 format volume identification on the device without initializing (zeroing) the device; this eliminates warning messages. Functions operate properly, even if a warning is issued.

The device structure changes are as follows:

1. The bootstrap is extended to occupy five blocks of the device, specifically block 0 and blocks 2 to 5. Previously, the bootstrap occupied blocks 0 and 2, which severely constrained the functionality of the bootstrap.

2. Block 1 of the device, formerly reserved as the home block, now supports the DIGITAL volume identification standard. This standard defines the contents of certain words in block 1 of a system volume. Support of this standard permits identification of the system type and file format of a volume across DIGITAL operating systems. The volume identification block is defined as follows:

Word (octal) Offset	Contents
0-353	Reserved for specific use by the operating system. See below for definition of this area.
354-361	Volume identification. This is a 12-character (decimal) volume identification, stored in packed ASCII bytes. This is the physical volume identification (label); it can be assigned by the owner when the volume is initialized.
362-367	Owner name. This is a 12-character (decimal) packed ASCII field containing the name of the volume owner. This field can be spaces or the word "SYSTEM" if the owner is unknown or the volume is a system volume.
370-375	Format type. This is a 12-character (decimal) packed ASCII field containing the file format on the volume. For all RT-11 systems, this field must contain the characters "DECRT11A".
376	Unused.
377	Checksum, a 16-bit integer.

In the preceding fields, characters can be in the range of 40 (octal) to 176 (octal) in seven-bit ASCII. Unused positions in the field are padded with spaces and all names are left-justified.

All other words are reserved for system use only. Some words currently used are:

Word	Contents
0-100	Bad block replacement directory for RK06 only.
352	Directory starting block. This word contains the absolute block number where the RT-11 directory begins. It is a 16-bit integer value. (RT-11 directories start at block 6.)
353	Contains the Radix-50 value of the letters "V3A" (for V03 systems), or "V3B" (for V03B systems), providing a single one-word check that the volume identification block was properly initialized by the utility program DUP.

3.4 BATCH

3.4.1 BATCH Changes

Several changes between RT-11 V02C and RT-11 V03 have a significant impact on BATCH users. As a result of the following changes, BATCH users must recompile their BATCH jobs to produce new control (.CTL) files. If you use RT-11 mode of BATCH, examine your BATCH files carefully, keeping in mind the new keyboard monitor commands and the changes listed below with regard to PIP functionality, FORLIB, and SYSLIB.

1. The formats of the new RT-11 keyboard monitor commands and the BATCH commands are very similar, but the two command languages are definitely not interchangeable. The BATCH commands are not compatible with the new keyboard commands, so BATCH users must be careful when switching between the two, as in the RT-11 mode of BATCH.

2. The functionality incorporated in RT-11 V02C PIP is now contained in PIP, DUP, and DIR. BATCH now uses DIR to print directory listings for $DIRECTORY commands; therefore, the format of these listings is changed.

3. You must build your own FORTRAN library. To do so, you can combine SYSF4.OBJ (the file for RT-11 FORTRAN callable routines) with other FORTRAN object modules to create a file called SYSLIB.OBJ. SYSLIB.OBJ is the default library searched by the linker. BATCH uses SYSLIB.OBJ as the default library for linking FORTRAN programs. BATCH assumes that SYSLIB resides on SY:.

4. BASIC-11/RT-11 is changed significantly. If you run BASIC programs under BATCH, you must read the <u>BASIC-11/RT-11 Installation Guide</u> for a list of differences between V01B and V02 of BASIC-11.

5. RT-11 FORTRAN is also changed. If you run FORTRAN programs under BATCH, you must read Appendix D of the <u>RT-11/RSTS/E FORTRAN IV User's Guide</u> for a list of differences between V01C and V02 of FORTRAN IV.

6. The keyboard monitor ASSIGN command is now two separate commands: ASSIGN and DEASSIGN. BATCH uses these commands to assign and deassign logical device names in the $MOUNT and $DISMOUNT commands.

    The keyboard monitor ASSIGN command requires a space between the physical and logical device names. If you have RT-11 mode ASSIGN commands in your BATCH files, you must change them to the new format ASSIGN and DEASSIGN commands.

7. The keyboard monitor Examine (E) and Deposit (D) commands now perform stricter syntax checking. Check your BATCH files for any RT-11 mode E or D commands.

8. The RT-11 V02C INIT keyboard command is renamed RESET. The INITIALIZE command initializes (zeroes) an RT-11 volume.

9. The Command String Interpreter (CSI) and the FRUN and GT ON keyboard commands no longer accept exclamation marks as decimal radix indicators in options (for example, /S!19). Exclamation marks are now used as comment delimiters for indirect command files. Decimal points indicate decimal radix (for example, /S:19.). If you use the RT-11 mode of BATCH, check your BATCH streams and change to the new syntax.

10. The BATCH commands $COPY, $DELETE, $DIRECTORY, and $PRINT support wildcards of the type *.MAC, TEST.*, and *.*. These commands do not support the extended wildcard and wild character format of the new keyboard commands (for example, A%B.T*). Note, however, that you can freely use the new keyboard monitor commands in the RT-11 mode of BATCH with no restrictions on wildcard usage. For instance, the following is a valid RT-11 mode BATCH job containing an indirect command file.

    ```
    $JOB/RT11
            TTYIO
    .COPY A%B.MAC *.NEW
    .DIRECTORY/COLUMNS:4 A%B.T*
    .@TESFIL
    .PRINT TESFIL.COM
    $EOJ
    ```

    The indirect command file (TESFIL.COM) has the same format whether it is invoked interactively or through BATCH; no "." or "*" characters are necessary in the indirect file, even though it is used under BATCH.

    See the <u>RT-11 System User's Guide</u>, Chapter 4, for detailed information about indirect command files.

11. When using the new keyboard commands in RT-11 mode of BATCH, be careful to anticipate the additional querying incorporated in commands like DELETE *.TMP. The easiest way to avoid this problem is to use the /NOQUERY option where needed (for example, DELETE/NOQUERY *.TMP); failure to do this results in having to follow the command line with a line like *Y (RET) for every file you are deleting. (Note that the BATCH $DELETE command still works as in V02C.) You have the choice of including responses in the BATCH stream, as follows:

```
$JOB/RT11
.LINK/TRANSFER ODT
*O.ODT                    (response to the linker's
                          Transfer address? prompt)
$EOJ
```

12. PIP no longer supports operations of the following form:

    *A.MAC,B.MAC=C.MAC,D.MAC (RET)

    to copy C.MAC to A.MAC and D.MAC to B.MAC. The corresponding BATCH operation in RT-11 V02C was:

    $COPY A.MAC+B.MAC/OUTPUT C.MAC+D.MAC/INPUT

    This command is not legal in RT-11 V03 BATCH; its use causes a BATCH-F-ILLEGAL '+' error message.

3.4.2 BATCH Problems Corrected

The following problems are corrected in RT-11 V03 BATCH.

1. The BATCH compiler no longer malfunctions when an output device is full.

2. The $MACRO/RUN, $FORTRAN/RUN and $LINK/RUN commands can now generally contain in-line data. Previously, if these commands caused a temporary .SAV file to be created, such in-line data could not be accessed. These commands now also operate correctly even if DK: is not the same device as SY:.

3. BATCH correctly detects end-of-file when input is from the card reader.

4. BATCH no longer malfunctions if it detects end-of-file before it encounters a $JOB card.

5. $DATA is usable even if $JOB/TIME is specified.

6. BATCH correctly logs out lines in the .CTL file containing \ (backslash) when it detects an error.

7. BATCH now accepts an 80-character input line.

8. The BA handler correctly accepts multi-character \ (backslash) operator directives from the console terminal. Previously, only two-character directives were handled properly.

9.   BATCH generates an error message instead of a monitor   trap
      if not   enough memory exists   to .FETCH a required device
      handler.

10.   The BA handler no longer malfunctions when a   label   in   the
      .CTL file spans a physical block boundary.

11.   BATCH automatically unloads the BA handler in response   to
      the /U option. You no longer need to manually UNLOAD BA.

12.   Previously, BATCH error messages were not always printed   on
      the console terminal and were sometimes lost. This error is
      corrected where applicable.

3.4.3 BATCH Restrictions

1. $EOJ must be the last line in a BATCH file that   is $CALLed
      from another BATCH file. If any lines, or any characters
      other than carriage return and line feed, follow the $EOJ
      command, a malfunction occurs in BATCH. If extra blank
      lines or data lines appear after the $EOJ line, the END
      BATCH message appears prematurely on return from the $CALLed
      file.

2.   If a $MESSAGE appears in a BATCH file immediately following
      a   program   that   performs its own .EXIT, BATCH can send the
      message to the LOG device rather than to the console
      terminal.   (RT-11   PATCH, for example, does an .EXIT
      programmed request in response to the E command.) To avoid
      this problem, place an RT-11 mode .DATE or .TIME command
      between the program exit and the $MESSAGE command.

3.   BATCH no longer deletes the temporary .SAV files it produces
      for $MACRO, $FORTRAN, and $LINK commands that contain no
      explicit name for the executable file.   The deletion of
      these .SAV files was preventing the use of $DATA with
      commands that generated a temporary .SAV file. If space on
      your system device is a problem, you can modify BATCH jobs
      to explicitly delete the temporary .SAV files after
      execution of the $MACRO, $FORTRAN, and $LINK commands. To
      delete any temporary .SAV files that are created, include
      the following commands in your BATCH job.

```
$RT11 (RET)
.DELETE/NOQUERY 000000.SAV (RET)
$EOJ (RET)
```


3.5 PIP

PIP remains a standard part of the operating system.   Many options
included in other utility programs are deleted from PIP and many of
the V02C PIP options have new meanings. If you used previous versions
of RT-11 PIP, you should familiarize yourself with these new options
by reading the following material and Chapter 7 of the RT-11 System
User's Guide before using PIP.

3.5.1 PIP Changes

1. The DIR program performs the directory listing functions previously done by the /E, /F, /L, and /W options in PIP.

2. The DUP program performs the volume initialization, compression, bootstrap, and bad block scanning functions previously done by the /S, /T, /Z:n, /U, /K, /N:n, and /O options in PIP.

3. PIP magtape support is improved so that only one pass is made on a magtape when you use the /M:-1 option or use a wildcard. PIP V02C made two passes.

4. The PIP /W option logs on the terminal the names of files that PIP copies, renames, and deletes.

5. PIP no longer concatenates files by default. That is, you need not (and indeed cannot) use the /X option to copy files separately. You must use the /U option to explicitly concatenate files unless you use the keyboard monitor COPY command and separate the input files with a plus sign. For example,

    .COPY A.MAC+B.MAC+C.MAC D.MAC (RET)

    or

    .R PIP (RET)
    *D.MAC=A.MAC,B.MAC,C.MAC/U (RET)

    concatenates files A.MAC, B.MAC, and C.MAC to produce file D.MAC.

6. PIP preserves the current file creation date across copy and rename operations unless you specify the /T option. If the input file has no date, PIP provides the current date.

7. PIP allows only one output file specification rather than the three that PIP V02C accepted.

8. PIP no longer causes any movement of files when you specify the same file for input and output. For example:

    .R PIP (RET)
    DT:.*=DT:*.* (RET)
    RK1:.*=RK1:A.MAC (RET)
    *RK1:A.MAC=RK1:A.MAC (RET)

    causes no change in the physical position of the files on the device. This sequence of commands does, however, cause an input file with no date to be given the current system date.

9. PIP issues error messages as soon as it encounters an error, but PIP can have other processing to do before it returns to the command level (or to the keyboard monitor dot if you use a keyboard command like COPY). Therefore, to avoid unpredictable results, do not abort PIP until it returns to the command level.

10. PIP's wildcard capabilities are enhanced to include wild characters (%) and wildcards (*) embedded within a file specification.

11.   PIP has an exclude option that causes a command to operate
       on all files except those you specify (/P).

12.   PIP has a no-replace option that helps prevent accidental
       file deletion (/N).

13.   PIP has a predelete option that helps avoid device full
       errors when transferring a file to a device that contains an
       old version of the file (/O).

14.   PIP has a single block transfer option that can help avoid
       hard device errors (/S).

15.   PIP has a number-of-copies option that you can use for
       making multiple copies of a file on the terminal, line
       printer, or high-speed punch (/K:n).

3.5.2 PIP Problems Corrected

The following problems that existed in V02C PIP are corrected.

1.   When you use the /G option in a command string, PIP no
       longer goes into an infinite loop or incorrectly copies a
       file if an error occurs.

2.   A problem is corrected that caused PIP to incorrectly clear
       the directory residing in the USR buffer.

3.   A problem is corrected that caused PIP to create abnormally
       large files when it received input from the console terminal
       or the papertape reader.

4.   PIP retains the old date when transferring files.   If you
       include the /T option, PIP gives the copied files the
       current system date.

5.   The ?OUT ERR error message that appeared when you tried to
       write a file onto a device that was full was unclear. The
       following error messages now appear to clearly state the
       problem:

           ?PIP-F-Output error dev:filnam

       appears when a fatal hardware occurs, and

           ?PIP-F-Output file full dev:filnam

       appears when there is not enough space for the output file.

3.5.3 PIP Restrictions

If you must use PIP with the system device dismounted, use the
following command sequence:

   .SET USR NOSWAP (RET)
   .R PIP (RET)

   (now dismount system device)

   *command/I:DIS (RET)

The /I:DIS option tells PIP to ignore certain errors caused by the system device being dismounted.

3.6 DIRECTORY PROGRAM (DIR)

DIR, a new RT-11 program, performs some directory listing operations formerly performed by PIP and includes some new functions. DIR performs the directory listing functions formerly done by the PIP options /E, /F, /L, and /W. These options are not in PIP. See the RT-11 System User's Guide, Chapter 9, for a detailed description of DIR.

Some of the principal features of DIR are as follows:

1.   DIR supports both wildcard and wild character functions (for example, A%B.*).

2.   DIR can list directories in multiple columns, as requested (for example, /C:3).

3.   DIR can sort directory listings:

   - alphabetically by file name or file type (/S:nam or /S:typ)
   - chronologically by file creation date (/S:date)
   - by file size (/S:siz)

   You can reverse the sort order of any of the above (for example, /S:nam/R to list files in reverse alphabetical order).

4.   DIR can list starting block numbers in either octal (/O/B) or decimal (/B).

5.   DIR can list only the files with a specific creation date (/D:date).

6.   DIR can list only the files that appear after a specified file in the directory (/G).

7.   DIR can list only the files created after a specific date (/J:date) or before a specific date (/K:date).

8.   DIR can list only the unused areas on a specified device (/M).

9.   DIR can list all files other than the specified files (for example, *.SYS,*.SAV/P to list all files with a file type other than .SYS or .SAV).

10.   DIR can list the directory information for files that were deleted but whose directory information is not yet destroyed (/Q).

11.   DIR can list a directory summary (/N).

3.7 DEVICE UTILITY PROGRAM (DUP)

DUP, a new RT-11 program, incorporates some file-oriented functions that PIP formerly performed and includes some new functions as well.

3.7.1 DUP Features

DUP performs the volume initialization and compression, bad block scanning, bootstrap and bootstrap copy, and file extension functions formerly done by the PIP options /Z, /N, /S, /K, /O, /U, and /T. These options are no longer in PIP. See the RT-11 System User's Guide, Chapter 8, for a more detailed explanation of DUP.

Some of the principal features of DUP are:

1.   Bad Block Support

     DUP provides bad block support by:

     a.   Supporting bad block replacement on the RK06/07 and RL01 disks at volume initialization time (/Z/R).

     b.   Automatically scanning a device for bad blocks at volume initialization time and covering those blocks with .BAD files to prevent them from being accessed (/Z/B).

     c.   Scanning a volume and listing the name of the file that contains a bad block and the relative block number of the bad block in the file (/K/F).

     d.   Optionally trying to recover a bad block it detects when scanning a volume by writing and rereading the block (/K/H).

     e.   Supporting partial bad block scans of a volume with user-supplied start and end block numbers (/K:start:end).

2.   Recovery of Deleted Files

     DUP supports a file create option with a user-supplied starting block number and file size that, together with DIR's ability to list directory information for deleted files, provides a simple way to recover an accidentally deleted file (/C:start:size).

3.   Image Copy

     DUP supports an image copy for fast disk backup to another disk or to magtape (/I).

4.   Volume Compression (SQUEEZE)

     DUP supports volume compression by:

     a.   Checking for an initialized output volume in a two-device compress operation to prevent accidental destruction of a volume.

     b.   Compressing around .BAD files to prevent the bad blocks within such files from being reused.

     c.   Printing a warning message if the device to be compressed does not have a V03 format volume identification and owner name. (You can compress V02C volumes since there is no change in device structure other than in the previously unused block 1.)

d.   Reverting to single-block transfers if an I/O error occurs during a multi-block read or write.

e.   Not compressing the system device while a foreground job is loaded.

f.   Automatically rebooting the system after compressing the system device. You can suppress this rebootstrap if no system files move during the compression.

5.   Volume Identification

DUP supports a user-supplied volume identification and owner name written on block 1 of a volume (or in the VOL1 header on magtape). DUP supports listing of the volume identification and owner name of a specified volume. DIR can also list this information as part of a directory listing.

6.   Bootstrap Operations

DUP supports both a hardware bootstrap of a specified device (/O) and a software bootstrap of a specified monitor file (ddMNxx.SYS/O). The latter does not affect the bootstrap blocks of the device, and it makes switching between monitors easy and non-permanent. (The bootstrap blocks of a volume are 0 and 2 to 5; block 1 is the home block and contains the volume identification.)

DUP passes the current date and time to the monitor that it is bootstrapping. This means that you need not reenter them if you are just switching monitors or rebooting after a system device compress operation.

7.   Single Disk Support

DUP supports a mode of operation that allows you to perform utility functions on another disk in a single-disk system (/W). In this mode, DUP pauses before actually beginning an operation to allow you to switch volumes on a single-disk system.

3.7.2 DUP Restrictions

When you use DUP with the squeeze (/S) option, you must initialize the output device (if it is different from the input device) to ensure proper operation.

The volume containing DUP must be mounted when you use the /W option.

3.8 MULTI-TERMINAL SUPPORT

3.8.1 Multi-Terminal Features

The most important features of the RT-11 multi-terminal support are as follows:

1. One terminal is always designated as the background console terminal. The boot-time default is always the console interface, which is designated logical unit number (lun) 0. The KMON and background jobs use this console terminal.

2. The foreground job can share the background terminal, as in V02C. However, you can designate another lun as the foreground console at run time, using the FRUN /T:lun option.

3. The .TTYIN, .PRINT, .TTYOUT, .CSIGEN, .CSISPC and .GTLIN requests and all reads and writes to TT: use the console terminal that is attached to the job.

4. You access the additional terminals on the system with the following new programmed requests.

Request	Meaning
.MTATCH	Attach a terminal for exclusive use by this job.
.MTDTCH	Detach a terminal.
.MTGET	Get terminal characteristics.
.MTIN	Get one or more characters.
.MTOUT	Print one or more characters.
.MTPRNT	Print a line.
.MTRCTO	Reset CTRL/O flag.
.MTSET	Set terminal characteristics.

5. Remote terminal support is available with the multi-terminal handler. Modem control is available for both DL11-E and DZ11 interfaces. The DL11 control answers ring interrupts, permitting terminals to dial-in to the system. Dial-in is possible with the DZ11 interface, despite the lack of a ring interrupt in the DZ11, if the modem is operated in auto-answer mode. This is achieved with a polling routine that periodically checks the status of each line on the multiplexor.

6. RT-11 multi-terminal support offers a time-out option to improve system reliability. If you select the option, time-outs are done on transmitted characters. If a transmitted character does not generate an interrupt within the timing interval, a support routine is called to restart the line. In addition, a polling routine periodically checks the input interrupt enable bits of active lines. This reduces the effect of static discharges and improves system reliability.

7. Another option provides a program with a set of event flags for each terminal it has attached. These flags are asynchronously updated as the status of each line changes.

Flags are provided for the following events:

- input is available on this line
- the output buffer for this line is empty
- a double CTRL/C was entered on this line
- the line has hung up (remote line)

These event flags are useful for applications monitoring many lines. They reduce program overhead since lines need not be polled to determine if they need service.

3.8.2 Multi-Terminal Restrictions

The following restrictions apply to multi-terminal support.

1. Support of the DL11-W interface requires the presence of a REV E module, or an ECO (DEC-O-LOG M7856-S0002) must be applied to the M7856 module. This is true for any terminal.

2. Dial-up support of DZ11 interface requires a BELL 103A type modem with "common clear to send and carrier" jumpers installed. With this option installed, the modem operates in an auto-answer mode, which is necessary since the DZ11 lacks a ring interrupt.

3. The console terminal interface must be a DL11 type interface and the terminal must be local.

4. The number of DL11 interfaces supported, both local and remote, is limited to eight, including the console terminal interface.

5. The number of DZ11 controllers is limited to 2, for a total of 16 lines plus the console interface.

6. The VT11 scroller option is disabled when multi-terminal support is present. Graphics is still supported, and the display support in EDIT and TECO functions with multi-terminal support.

7. The maximum input single terminal data rate for all systems is 300 baud and an aggregate total of 4800 baud on multi-terminal systems. The output can be set at any baud rate.

3.9 LINKER

This section provides you with all the changes, problem corrections, and restrictions for the RT-11 linker (V05).

3.9.1 Linker Changes

The RT-11 linker is significantly changed. (Read Chapter 11 of the RT-11 System User's Guide for detailed information.) Many changes are internal and do not affect you except to improve performance. Some changes, however, have impact on linker users. These changes are as follows:

1. The third output file in the command string is now a symbol definition file (STB).

2. The linker automatically searches a default system library (SY:SYSLIB.OBJ) last to locate any remaining unresolved globals. DIGITAL recommends that you include in SYSLIB.OBJ any language libraries, such as the FORTRAN Object Time System (OTS). If you use the FORTRAN-callable system routines contained in the concatenated binary file SYSF4.OBJ, you should also include them in SYSLIB.OBJ. Although the /F option is still available to include FORLIB.OBJ, you should avoid using /F because it can hinder proper linker performance and proper execution of the monitor EXECUTE and LINK commands.

3. The linker load map has more information than before. It contains the .IDENT value, date and time the file was created, program-section attributes, and segment size in both bytes and decimal words. The load map always lists named program sections even if their size is zero.

4. The linker automatically fills all unused locations in the load module with zeroes rather than random data, as the V02C linker did. You can change the value to something other than zero by using the /Z:n option.

5. The entire error message processing procedure is changed and greatly expanded. For more information, refer to the RT-11 System User's Guide, Chapter 11, and to the RT-11 System Message Manual.

6. The major change that has a direct effect on the load module is the new support of program sections (.ASECTs, .PSECTs, and .CSECTs). The linker concatenates all blank (unnamed) sections as one local section for each segment. This feature can change memory allocations. The sections are allocated in the order in which they appear in the input files list, except for library files, which always come last.

7. The linker treats global symbols and program section names separately. It does not consider the names assigned to sections as global symbols and does not reference them as such. A global symbol and a program-section name can have exactly the same name, but the linker treats them separately.

8. .CSECTs are implemented (internally via the MACRO assembler and linker) by a .PSECT with attributes equivalent to those of a .CSECT. See Section 6.8.2 of the PDP-11 MACRO-11 Language Reference Manual.

9. In an overlaid program, a global (GBL) is known throughout the entire program. If two or more segments contribute to a global section, then the linker allocates that global section in the root segment. In contrast to global sections, local sections are only known (allocated) within a particular program segment.

10. In an overlaid program, global symbols defined in a data program-section are resolved directly rather than through the overlay handler.

11. The following changes affect foreground programs.

NOTE

In RT-11 V03, you must relink
all V02C foreground programs
(REL files).

a. The FRUN command no longer takes the /S:n option for
stack space. The argument of the linker /R[:n] option
(or of the /FOREGROUND[:n] option for the monitor LINK
command) determines the foreground job's stack size.
The default size is 128 (decimal) bytes of stack space.

b. The REL file format is slightly changed to correct some
problems.

c. The default base address for foreground programs is
1000. FRUN takes this into account when the program is
loaded. This permits .ASECT values below 1000 to be
stored.

d. The relative word offset as well as the code to be
relocated is now stored for words to be relocated that
are located in the root segment of the foreground job.
This allows a particular location to be filled more
than once (using location counter arithmetic or OVR
section attributes).

3.9.2 Linker Problems Corrected

The following problems are corrected in the RT-11 linker:

1. A problem is corrected that caused a subroutine to be offset
one byte if the subroutine contained a .BLKB storage
allocation directive and was linked into an overlay region.

2. A problem is corrected that caused load maps to occasionally
lose characters.

3. When constructing a REL file, the linker checks to determine
whether space is available in its output buffer before it
writes out the 177776 terminating flag. If there is no
space available, the linker generates a ?LINK-F-REL Write
beyond end of file error message.

4. The code involving the use of the /S option in the linker is
completely rewritten.

5. An error message ?LINK-F-SAV device full appears if you
attempt to output a file to a full device. Previously, the
message ?HARD I/O ERROR appeared.

6. The undefined globals message is always displayed on the
console terminal.

7. The linker correctly links a library routine referenced from
the last overlay specified in a link command.

8. The linker generates the following error messages if program
space exceeds address capability.

?LINK-F-Address space exceeded

appears if the high limit of all program sections exceeds 32K words when all sections are concatenated.

?LINK-F-Size overflow of section (section name)

appears when the program section named in the message increases in size by more than 32K words.

9. The memory available for the linker's internal tables is dependent on the amount of memory that your system has. Previously, memory was allocated independent of the system's memory size.

3.9.3 Linker Restrictions

The following restrictions apply to the RT-11 linker:

1. Library files can never contain a FORTRAN BLOCK DATA subprogram because there is no undefined global symbol to cause the linker to load it automatically. The file containing the BLOCK DATA subprogram must be specified as one of the input files.

2. Complex relocation (global arithmetic) is not supported for foreground programs. If you use complex relocation, the following error message appears:

?LINK-W-Complex relocation of <symbol>

where <symbol> represents the global symbol involved.

3. Modules in one library can call modules from another library. However, the libraries must appear in the command string in the order in which they are called.

4. The librarian does not automatically include library modules in overlay segments. You must do this manually by extracting the module from the library with the librarian and then including the module explicitly in the desired overlay segment of the link command.

3.10 LIBRARIAN (LIBR)

This section provides you with all the changes, problem corrections, and restrictions for the RT-11 librarian.

3.10.1 Librarian Changes

The following changes are made to the RT-11 librarian.

1. The librarian now creates macro library files for use with the MACRO-11 assembler.

2. The object library header is reduced in length by one word. The last word of the header was removed to allow the header size to be a multiple of four words so that the linker processing time of library files is reduced. Also, the time of creation is stored in two words after the date word.

3. The object library directory does not include module names or program section names. The linker does not use them as part of automatic resolutions of undefined globals. You can include them, if required, by using the /N or /P option (refer to the RT-11 System User's Guide for complete details).

3.10.2 Librarian Problems Corrected

The message ?OUT ERR was not clearly defined when you attempted to write a file onto a device that was full. Two messages now appear in its place:

    ?LIBR-F-Output write error

appears when a fatal hardware error occurs.

    ?LIBR-F-Output file full

appears when there was not enough space allocated for the output file.

3.10.3 Librarian Restrictions

The following restrictions apply to the RT-11 librarian:

1. Library files can never contain a FORTRAN BLOCK DATA subprogram because there is no undefined global symbol to cause the linker to load it automatically from the library file.

2. The librarian does not support a directory listing of a macro library file.

3.11 SYSF4 FORTRAN SYSTEM SUBROUTINES

This section provides you with all of the changes, problems corrected, and restrictions for RT-11 V03 SYSF4 (formerly referred to as SYSLIB in V02C) FORTRAN system subroutines.

3.11.1 SYSF4 Changes

The FORTRAN-callable system subroutines are distributed as a concatenated set of object modules called SYSF4.OBJ. This set of routines is normally included with the FORTRAN OTS routines in the default system library (SYSLIB.OBJ) used by the linker. If you do not use the SYSF4 routines, you need not include them as part of the default system library. If you do use the SYSF4 routines, you must build SYSLIB.OBJ as detailed in Section 2.3.6 of the RT-11 System Generation Manual.

SYSF4 contains the following new subroutines:

Routine	Explanation
GTLIN	Get a line of ASCII input from console terminal or indirect file.
IGETSP	Get address and size of free space from FORTRAN OTS free space area.
IPEEKB and IPOKEB	Peek and poke for byte addresses.
MTATCH MTDTCH MTGET MTIN MTOUT MTPRNT MTRCTO MTSET	Multi-terminal service routines.
SCCA	Provides CTRL/C intercept.
SETCMD	Stores a command string in the chain area to be executed when the program EXIT's.

The following routines are changed.

Routine	Explanation
GETSTR and PUTSTR	The error flag argument is required and returns I/O errors or end-of-file conditions.
ILUN	Corrected so that a call made to the ILUN routine does not create the appearance that one I/O channel is not available to the FORTRAN OTS.
$TIME1	Corrected for 50Hz clock; this routine is called by ISCHED.

3.11.2 SYSF4 Problems Corrected

The following problems are corrected in the SYSF4 routines:

1. The GETSTR and PUTSTR routines now allow the user program to handle errors. A cleared error flag on return from these routines indicates that no error occurred. The error flag is set upon receipt of an I/O error or end-of-file condition.

2. An error is corrected in the SYSF4 $TIME1 routine that always calculated elapsed time based on the 60Hz clock. The calculation is now made for both 50Hz and 60Hz clocks.

3.11.3 SYSF4 Restrictions

The SYSF4 subroutines do not support the XM monitor mapping programmed
requests. You can use the FORTRAN virtual arrays to access extended
memory.

3.12 OBJECT MODULE PATCH UTILITY (PAT)

The PAT.SAV program replaces the RT-11 V02C PATCHO program.   PAT
operates in a different manner from PATCHO but accomplishes the same
functions while offering greater flexibility.   For a more detailed
description of PAT, see Section 2.3.6 of this manual and the RT-11
System User's Guide, Chapter 18.

3.12.1 PAT Restrictions

You must assemble the input object file and input corrections of
object files submitted to PAT with the same release of MACRO. You
must assemble both input files with RT-11 V03 MACRO, or you must
assemble both input files with RT-11 V02C MACRO.

3.13 FILEX

FILEX is changed and improved. This section provides you with the
improvements, problems corrected, and restrictions in the new FILEX
program.

3.13.1 FILEX Improvements

FILEX supports the universal interchange diskette format.   This
support allows you to transfer files between block replaceable devices
used by RT-11 and interchange format (proposed ANSI format) diskettes.
For complete details about FILEX, see the RT-11 System User's Guide,
Chapter 14.

3.13.2 FILEX Problems Corrected

The following FILEX problems are corrected:

1. A problem is corrected that caused duplication of file
    specifications on a DOS tape if the output file of the FILEX
    command line was the same as an already existing file.

2. A problem is corrected that caused FILEX to incorrectly
    check the DOS directory and input file name for duplication.

3. FILEX now generates an error message if you attempt to pass
    an illegal command string (such as DT:filnam.*/S=*.typ).

3.13.3 FILEX Restrictions

The following restrictions apply to FILEX:

1.   FILEX requires a system with a memory size of 12K words or greater.

2.   FILEX does not support the use of embedded wildcards in any file name or file type.

3.   FILEX does not support magtape or cassette in any operation.

4.   If you attempt to use RT-11 devices for both input and output, FILEX generates an error message.

5.   FILEX ignores a number used with the /U option if it is specified on the input side of a command line or if it is used with the /A or /I option.

6.   FILEX can list device directories only on the console terminal.   FILEX can only list directories of block replaceable devices.

3.14 EDIT

This section provides you with all the changes and problem corrections for the RT-11 editor, EDIT.

3.14.1 EDIT Changes

1.   When a SET TT SCOPE command is in effect, EDIT echoes BACKSPACE,SPACE,BACKSPACE to erase the character on the screen rather than using the backslash convention appropriate for rubout on hard-copy terminals.

2.   To exit from the editor with CTRL/C, you must type (CTRL/C) (ESC) (ESC).

3.   You can reenter EDIT immediately after typing a CTRL/C without losing text from the text buffer. The macro and save buffers are reinitialized as before, but the text buffer remains intact.

4.   When an output file is full, you can close the output file with an EF command or open another output file with an EW command. If the output file is full and EDIT returns an ?EDIT-F-Output file full message, the pointer is moved in the text buffer to the point at which successful output stopped. You can complete the editing session without loss of text by deleting the text between the start of the buffer and the moved pointer, closing the full output file, opening a new output file, and resuming the editing operation.

Note that this error condition is the only case in which write operations cause the pointer to be moved in the buffer. In all successful writes, the pointer remains intact. An example of the recovery procedure follows:

```
.R EDIT (RET)
*EWFILE1.MAC (ESC) (ESC)
*ERINPUT.MAC (ESC) (ESC)
*EX (ESC) (ESC)                    ;copy file over
?EDIT-F-Output file full           ;file is full
*-100K (ESC) (ESC)                 ;delete lines successfully
                                   ;written
*EWFILE2.MAC (ESC) (ESC)           ;close FILE1 and open
                                   ;FILE2
*EX (ESC) (ESC)                    ;close out edit
```

Note that you can often avoid this error condition by using the square bracket [size] construction when opening a large file for editing with the EB or EW commands (use /ALLOCATE:n for EDIT/CREATE commands.

5.   EDIT error messages are significantly expanded to be more readable and to supply additional information. Among the changes are:

a.   ?EDIT-F-Superseding existing file

warns that you issued an EW command for a file name that already exists.

b.   If a device is write-locked, the editor processes the error by issuing the following message:

?EDIT-F-System I/O error

rather than a fatal monitor error, allowing you to correct the condition and retry.

c.   The editor distinguishes between full directory and full device conditions.

d.   The editor also distinguishes between no file open for input and no file open for output conditions.

e.   The editor reports hardware read and write errors differently.

6.   You can enter an ER command subsequent to an EB command to alter the input file without affecting the output file. An EW command following an EB command closes the output file and renames the original input file to .BAK without affecting the current status of the input file. This feature allows greater flexibility in merging and separating files.

3.14.2 EDIT Problems Corrected

The following problems are corrected in EDIT.

1.   A problem is corrected that permitted the editor to lose characters if the first character in the buffer was a line feed.

2.   A problem is corrected that caused the editor to lose the last character of any file that was exactly n blocks long.

3.   A problem is corrected that caused the editor to append extra characters to the output file when you issued ER commands subsequent to a write operation.

4.   A problem in the display conditionals is corrected that prevented the NO-DISPLAY version of the editor from assembling correctly.

5.   A problem is corrected that prevented EDIT from recognizing (RET) (LF) and (RET) (LF) (LF) sequences as containing an END-OF-LINE (RET) (LF) pair.

3.15   ON-LINE DEBUGGING TECHNIQUE (ODT)

The following changes and restrictions apply to ODT.   See the RT-11 System User's Guide, Chapter 16, for additional information.

1.   A problem is corrected that caused the processor priority level to be set at level 6 rather than at level 7.

2.   Ten (octal) words are added to ODT's stack. As a result of this change, the starting address is changed from 1222 to 1232. This change affects only the source disk.

NOTE

Examples in the RT-11 System User's Guide use the 1222 address because the manual was printed before the change took place. The correct starting address for ODT is 1232.

3.16   DUMP

DUMP now can dump file-structured magtapes. The listing output contains a header message consisting of your command string as well as tape marks. See the RT-11 System User's Guide, Chapter 13, for further information.

3.17   MAC8K

The following restrictions apply to MAC8K.

1.   MAC8K runs only under the SJ base-line monitor (xxMNSJ.BL).

2.   When you run MAC8K in 8K words of memory, there is insufficient space to do a listing to the terminal handler (TT:) or the line printer handler (LP:). If you issue a command such as the following, the system halts.

     .MACRO/LIST:TT:/NOOBJ SY:DEMOBG

To obtain a listing, send the listing to a volume on any unit of the system device and then copy this output to the terminal or line printer.

3.18 DEVICE HANDLERS

This section provides you with all the specific device handler changes and restrictions. These changes are in addition to the changes made to the device handler format (Section 2.7.6), which apply to all RT-11 devices.

3.18.1 Magtape Handler

The following changes and restrictions apply to the magtape handler. For further information on magtape operation, see the RT-11 Advanced Programmer's Guide, Chapter 1.

3.18.1.1 Magtape Handler Changes -- The following changes are made to RT-11 magtape handlers.

1. The system increments sequence numbers for each file on magtape. Sequence numbers were not used in RT-11 V02C.

2. You can perform absolute positioning of the magtape drive by specifying exact sequence numbers. For example, if you desire to read file 4, you include the number 4 as the sequence number argument of the read request. The handler positions the tape in the gap preceding file 4 and then performs   the   read   request.   In   RT-11   V02C,   this specification (the number 4 as the count argument) caused the tape to position itself four files from its current position. For example, when current position was beginning-of-tape, the tape positioned itself in the gap preceding file 4, and when current position was file 3, the tape positioned itself in the gap preceding file 7.

3. RT-11 magtape handlers do not support the .DELETE and .RENAME programmed requests or the corresponding monitor commands.

4. RT-11 has new and improved error codes for hardware operations such as rewind, backspace, etc.

5. RT-11 magtape handlers allow file-structured .ENTER and .LOOKUP programmed requests.

3.18.1.2 Magtape Handler Restrictions -- The following restrictions apply to RT-11 magtape handlers.

1. Magtape handlers can read tapes written by V02C handlers but cannot write such tapes.

2. Magtape handlers do not accept a command that moves the tape forward after a write request unless that command is another write request. Accepting such a request results in unpredictable operation of the magtape.

3. When a file written in the file-structured mode encounters an EOT, it is your responsibility to close or remove any partially written file from the tape. The handler does not close   or   remove   the   file (see the RT-11 Advanced Programmer's Guide, Chapter 1).

4. The SET command allows you to specify the density and parity of the tape drive. It does not allow for the possibility that your system can have 9- and 7-track drives with tapes of different densities.

5. You cannot set density and parity characteristics with .SPFUN requests as you could in V02C. Change them by using the monitor SET command.

3.18.2 RK06/07 Disk Handler

The following changes and restrictions apply to the RT-11 RK06/07 disk handler (DM).

1. The DM handler supports bad block replacement. See the RT-11 System User's Guide, Chapter 8, for more information.

2. The monitor file cannot reside on a block that contains a BSE error if you are performing bad block replacement. If this condition occurs, a boot error occurs when you bootstrap the system. In this case, move the monitor so that it does not reside on a block with a BSE error.

3. If you did not cover BSE errors on an RT-11 V02C disk with .BAD files, you must now reinitialize the disk before you can use it.

4. RT-11 is distributed on RK06 disk.

3.19 PATCH

In previous versions of PATCH, you had to exit from PATCH with the E or F command to alter the size of an overlay region prior to patching it. RT-11 V03 PATCH does not have this restriction; rather, it has a special command (x;O) that informs PATCH that a value in the overlay handler or its tables is being modified. PATCH can then adjust its internal copy of the overlay information. PATCH has several new commands that help you modify ASCII and packed Radix-50 characters. PATCH also has a new checksum capability. See the RT-11 System User's Guide, Chapter 17, for more details on PATCH.

3.20 SYSTEM MACRO LIBRARY (SYSMAC)

The system macro library (SYSMAC) is changed so that it is now a true macro library file produced by the librarian.

3.20.1 SYSMAC Changes

1. The new MACRO assembler requires that macros requested via the .MCALL directive reside in a library file. Thus, RT-11 provides both the library file (SYSMAC.SML) and the library source file (SYSMAC.MAC).

2. RT-11 retains the ..V1.. and ..V2.. macros for compatibility; they cause system macros to generate the same expansions as in earlier releases of RT-11. Do not use

the ..V1.. and ..V2.. macros in new RT-11 programs. You need only specify system macros to be invoked with an .MCALL directive; there is no ..V3.. or equivalent.

3. Several problems in ..V2.. format expansions are fixed. The documentation stated that blank fields in a macro invocation are not set during execution of the expanded macro code. However, the length field on .ENTER, for example, was formerly set to zero if the argument was blank. For RT-11 V03, the length remains unset if the argument is blank. Specifying ..V2.. causes the same result as in RT-11 V02C, namely that the length is set to zero. For compatibility, the macros involved continue to expand in the same way as when you use ..V2... Macros affected by this are identified in the SYSMAC.MAC listing by the fact that they have an X as the fourth argument in a ...CM2 macro call. See Appendix B of the RT-11 Advanced Programmer's Guide for a listing of SYSMAC.

4. In many cases, the system macro expansions are significantly more efficient as far as number of words of code that the macro produces.

5. SYSMAC is expanded to support RT-11 V03 functionality, including support for multi-terminal macros, extended memory macros, device handler macros, .SCCA, .GVAL, .GTLIN, and .UNPROTECT.

6. A keyword argument, CODE, is now included in the form of macro expansions that include an AREA to give you greater flexibility in choosing how the macro will expand. Specifying CODE=SET or CODE=NOSET in the macro invocation allows you to choose whether or not the code byte (in the first word of the area) is set by the macro expansions. The CODE argument works whether or not you explicitly include an AREA argument in the macro invocation. Thus, you can load the address of the AREA into R0, omit the area argument from the macro invocation, and specify CODE=SET.

3.20.2 SYSMAC Problems Corrected

A problem is corrected that caused .DATE to fail if the current radix was decimal.

3.21 MACRO-11

RT-11 uses a new MACRO-11 assembler program. For complete details, refer to the PDP-11 MACRO-11 Language Reference Manual and the RT-11 System User's Guide, Chapter 10.

The following changes and restrictions apply to MACRO-11.

1. You can now define local symbols more than 128 words from the base of a local symbol block.

2. A problem is corrected that caused the LP:CR/C command string to generate an error message (OUTPUT DEVICE FULL). This problem occurred when the CREF file filled up and caused MACRO to halt.

3. A problem is corrected that caused macro arguments to always convert to upper case even when you enabled lower case.

4. MACRO permanently defines RETURN and CALL.

5. You can run MACRO-11 in 12K words of memory.

6. MACRO has a virtual (on disk) user symbol table.

7. MACRO supports user macro libraries.

8. The .REGDEF call is no longer necessary.

9. The following arguments are new for .ENABL and .DSABL:

    REG
    GBL
    CRF

10. You can define global symbols by either the :: or the == construction.

11. Local symbol names can range from 1$ to 65535$.

3.22 BASIC-11

Section 4.2.3 of the BASIC-11/RT-11 Installation Guide should read as follows:

   4.2.3 Foreground Execution -- You can run BASIC-11 in either the foreground or the background, but background is the only sensible choice. Using the foreground will tend to "hang" the system. If you must link BASIC to execute in the foreground with either the FB or XM monitor, type an F in response to the dialogue program's "Background or Foreground?" prompt.   To execute BASIC in the foreground, use the FRUN command with the /N option. Section 4.3 contains an example of linking BASIC and executing it in the foreground.

CHAPTER 4

RT-11 V03B

This chapter describes the RT-11 V03B release. RT-11 V03B is a minor
update from RT-11 V03. Its most significant feature is support of new
hardware devices. The following sections list the new features of
RT-11 V03B and detail the changes from Version 3.

4.1 RT-11 V03B SYSTEM OVERVIEW

The primary goals of RT-11 V03B are to support new hardware devices
and to introduce two new system utility programs. Some existing
system components have been improved as well, and some problems in
Version 3 have been corrected.   The following sections list the
components of RT-11 that are new.

4.1.1 New Device Support

RT-11 supports new mass storage devices and several hardware devices.
For your convenience, Table 4-1 lists the Radix-50 codes for all RT-11
devices.

Table 4-1
Radix-50 Codes for RT-11 Devices

RT-11 Device Name	Radix-50 Word
CR	012620
CT	012740
DD	014640 (new in V03B)
DL	015340 (new in V03B)
DM	015410
DP	015600
DS	015770
DT	016040
DX	016300
DY	016350 (new in V03B)
EL	020440
LP	046600
MM	051510
MT	052140
NL	054540
PC	062170
PD	062240 (new in V03B)
RF	070560
RK	071070

4.1.1.0 **RX02 Diskette** - RT-11 supports the RX211/RXV21 RX02 diskette as a storage medium and as a system device. The RX02 diskette is a single surface, double density flexible disk that offers approximately twice the storage capacity of the RX01 diskette. The new handler for the RX02 diskette is called DY.SYS for SJ and FB (and DYX.SYS for XM, if you select extended memory support).

The RX02 double density diskettes are upward compatible with RX01 single density diskettes. That is, the DY: handler can read and write single density diskette volumes that were created on the RX01 device. This is an automatic process and is completely transparent to users. Note, however, that FILEX does not support file transfers involving RX02 diskettes.

Additional .SPFUN programmed request codes are provided for the RX02. These codes permit you to read and write absolute blocks on the diskette and to determine the device size. The latter capability is useful when your program needs to distinguish between a single density and a double density diskette mounted in an RX02 drive.

RT-11 is distributed on RX02 diskettes. RT-11 does not support error logging for the RX02. See Section 4.1.2.2 for information on formatting diskettes for use with the RX02 device. If RX02 is your system device and you plan to create a DY monitor through the system generation process, read Section 4.2.24.

4.1.1.1 **RL01 Disk** - RT-11 supports the RL11/RLV11 RL01 disk as a storage medium and as a system device. The RL01 is a high-speed removable cartridge disk that offers approximately twice the storage capacity of the RK11/RK05 disk. DUP supports a replacement table for bad blocks on the RL01. You select this method of handling bad blocks by using the /R option with DUP (or by using the monitor INITIALIZE/REPLACE command). The new handler for the RL01 disk is called DL.SYS for SJ and FB (and DLX.SYS for XM, if you select extended memory support). An additional .SPFUN programmed request code allows programs to determine the size of a particular volume mounted in a device. RT-11 is distributed on the RL01. RT-11 does not support error logging for the RL01.

4.1.1.2 **RK07 Disk** - RT-11 supports the RK611/RK07 cartridge disk as a storage medium and as a system device. The RK07, a removable cartridge disk with twice the storage of the RK06, uses the same RK611 controller as the RK06 disk. The RK07 also uses the same handler, DM.SYS for SJ and FB (and DMX.SYS for XM, if you select extended memory support). An additional .SPFUN programmed request code allows programs to determine the size of a particular volume mounted in a device. DUP support of replacement bad blocks for the RK07 is the same as for the RK06. RT-11 supports error logging for the RK07.

4.1.1.3 **DECtape II Cartridge** - RT-11 supports the TU58 DECtape II Magnetic Tape Cartridge as a system device and as a random-access file-structured storage device. Each DECtape II cartridge stores 512 blocks of data and has a write-protect tab that slides to the right or left to write enable or write protect the cartridge. The permanent device name for DECtape II is DDn:.

The new handler for DECtape II is called DD.SYS. RT-11 is distributed on DECtape II cartridges. System generation is not a supported procedure on this device.

4.1.1.4 **PDT-11 Intelligent Terminals** - RT-11 supports both the PDT-11/130 Intelligent Terminal and the PDT-11/150 Intelligent Terminal. The PDT-11/130 Intelligent Terminal uses the TU58 DECtape II magnetic tape cartridge as the system device and storage device. The PDT-11/150 Intelligent Terminal, on the other hand, uses the RX01 single-density flexible diskette as the system device and storage device. The permanent device name for either PDT-11 device is PDn:. Directories and file data are interchangeable between DECtape II on the PDT-11/130 and DECtape II on the PDP-11. Likewise, directories and file data are interchangeable between single-density diskette on the PDT-11/150 and single-density diskette on the PDP-11.

4.1.1.5 **VT55 Terminal** - RT-11 supports the VT55 graphics display terminal. See the VT55 Programming Manual (Order No. AA-4949A-TC) for more complete information on how to program this terminal. Software for the VT55 consists of the following two files:

    PLOT55.OBJ
    TEST55.FOR

PLOT55 is a single FORTRAN-callable routine that provides access to the graphics features of the VT55 terminal. You can add PLOT55 to SYSLIB so that it is automatically linked into your programs at link time.

TEST55.FOR is a FORTRAN program that provides a simple but thorough demonstration of the VT55 graphics terminal in action.

To compile the test program and run the demonstration, you need the following files:

    FORTRA.SAV
    TEST55.FOR
    LINK.SAV
    PLOT55.OBJ
    SYSLIB.OBJ   (containing the FORTRAN OTS routines)

Enter the following commands:

    FORTRAN TEST55 (RET)
    LINK TEST55,PLOT55 (RET)
    R TEST55 (RET)
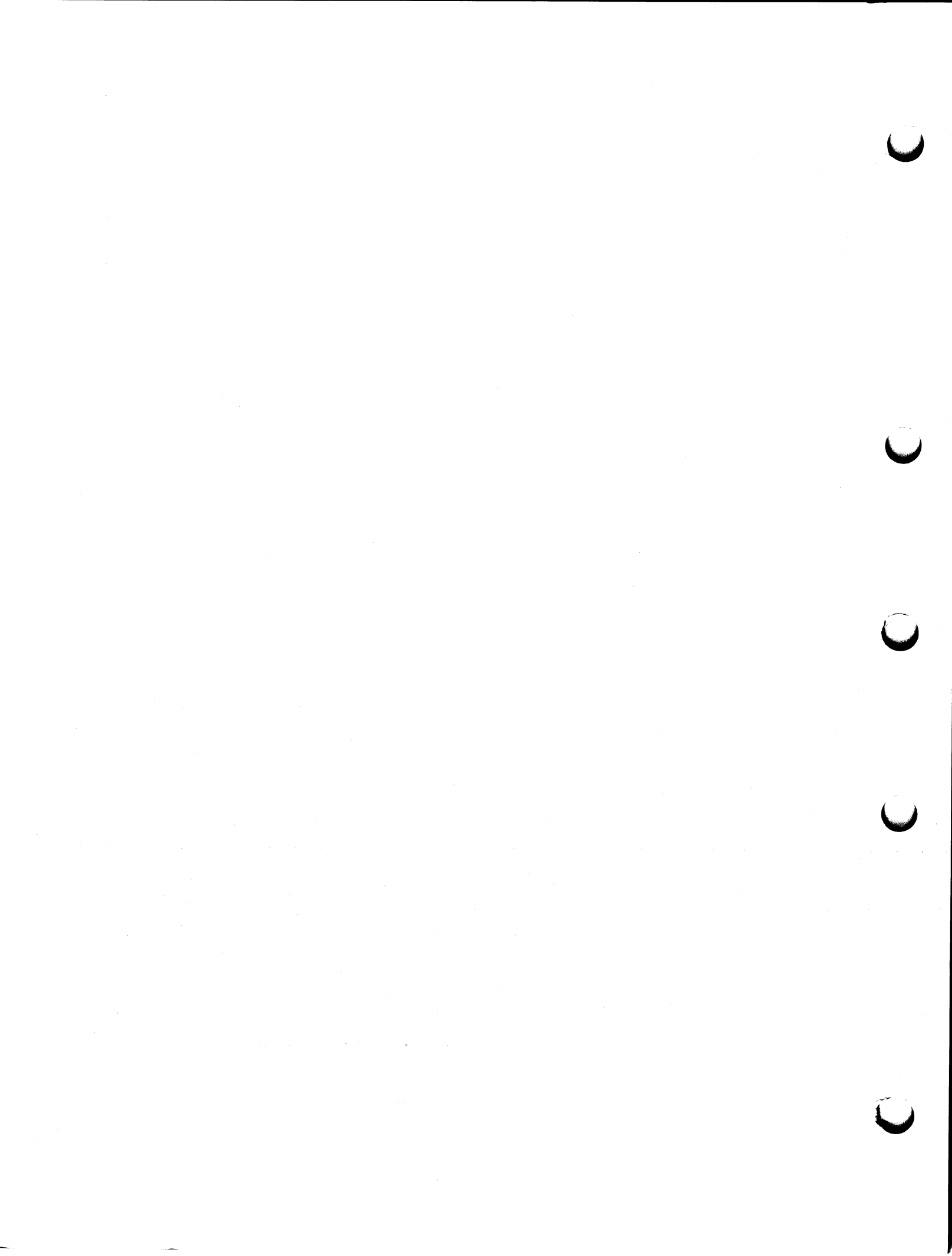
4.1.1.6 **Cache Memory for the PDP-11/34 Processor** - RT-11 supports cache memory (the KK11-A unit) on the 11/34 processor. Cache memory does not affect your system except to increase the speed of memory accesses. RT-11 error logging detects and reports cache memory errors.

4.1.1.7 **MSV11-DD Memory for the LSI-11** - RT-11 supports the MSV11-DD memory board unit for the LSI-11 microcomputer. This board enables you to access up to 30K words of memory with the SJ or FB monitor. The normal high limit is 28K words. A special jumper is required before you can use the extra 2K words of memory. This additional 2K words of memory is detected automatically at boot time; no action is required on your part. See the MSV11-DD hardware manual for details.

The REV11 bootstrap is not compatible with the 30K words option of the MSV11-DD board. The BDV11 bootstrap does not use any locations between 28K and 30K, as does the REV11 bootstrap, and is required with the MSV11-DD.

4.1.1.8 **New Line Interfaces** - RT-11 multi-terminal support includes support for the following line interfaces:

1. DLV11-J serial line interface (four DLV11s on a single board -- RT-11 supports two DLV11-Js)

2. DZV11 asynchronous four line multiplexor for the LSI-11 microcomputer (RT-11 supports one DZV11)

4.1.2 **New System Utility Programs**

Two system utility programs, RESORC and FORMAT, are new in RT-11. The following sections describe these programs, explain briefly how to use them, and reference the RT-11 documentation that describes them more fully.

4.1.2.1 **RESORC Utility Program** - The RESORC program examines the currently running RT-11 system and displays useful information about the status of the monitor and the system configuration. To access RESORC, use the keyboard monitor SHOW command with one of its three new options. See Section 4.2.1.4, which describes the changes to the SHOW command, for a list of the new options and a summary of the system resources each option displays. The SHOW command (if you specify any option) requires the file RESORC.SAV to execute properly.

4.1.2.2 **FORMAT Utility Program** - The FORMAT utility program sets media density marks for diskettes and writes headers for RK05 disks.

The diskette to be formatted can be mounted in any unit of the RX02 (DY:) device. You can format diskettes as single density (suitable for use with the RX01 device) or double density (for the RX02 device).

An RK05 disk to be formatted can be mounted in any unit of the RK05 (RK:) device. Formatting an RK05 disk takes approximately 25 seconds.

See Appendix C of the RT-11 System User's Guide for a more complete description of the FORMAT program.

4.2 CHANGES AND RESTRICTIONS

This section lists the system components that have been changed for RT-11 V03B. It also describes any restrictions that apply to the system components, and lists the problems from Version 3 that the V03B release corrects.

4.2.1 Monitor Changes

The RT-11 monitor has been changed to improve its performance and reduce the time required to assemble it. Two programmed requests have changes that reflect the new device support. Three of the keyboard monitor commands have changed slightly to reflect the new utility programs and to improve their performance. The following sections describe in more detail the changes to the monitor.

4.2.1.1 External Swap File - The RT-11 monitor has been reorganized to make it more modular. The swap blocks (that area within the monitor file on the system volume where sections of information from memory are temporarily stored) have been separated out of the monitor and placed in an external file called SWAP.SYS. This file is a logical part of the monitor file; it must always reside on the system volume along with the monitor. If for any reason you move SWAP.SYS, you must reboot the system.

Because the swap file is common to all monitors, you need to maintain only one copy of it on your system volume. The addition of this swap file represents a 24-block space saving for each additional monitor file you store on that same volume.

4.2.1.2 New Assembly Process - The assembly process for the monitor has been changed to simplify the process and to reduce the assembly time. Previously, the monitor was assembled into four separate components and then linked to form the monitor file:

1. BSTRAP.MAC

2. KMON.MAC, USR.MAC, RMONSJ.MAC (or RMONFB.MAC), and KMOVLY.MAC

3. System device handler

4. SYSTBL.MAC

Multi-terminal support was also assembled separately, if it was selected.

In RT-11 V03B, the components are assembled in five logical steps:

1. BSTRAP.MAC

2. EDTGBL.MAC, KMON.MAC, KMOVLY.MAC

3. EDTGBL.MAC, USR.MAC, RMONSJ.MAC (or RMONFB.MAC)

4. System device handler

5. SYSTBL.MAC

This arrangement significantly reduces overall assembly time for the monitor. Because the components are grouped differently, it is no longer necessary to reassemble KMON and KMOVLY when you make changes to RMON or the USR. A new file, EDTGBL.MAC, is required for monitor assembly. This file, which was previously part of KMON.MAC, contains the edit log, global definitions, system parameter definitions, and documentation on the monitor's data structures. It is no longer necessary to use a customized version of MACRO to do the assembly.

4.2.1.3 Programmed Requests - Two of the programmed requests are changed in V03B to reflect the new device support.

The .SPFUN request has a new special function code. The new function, 373, returns the medium size for devices that can support more than one kind of medium. It is valid for devices DL: and DM:.

In the .DSTATUS request, codes 5 and 23 in bits 0-7 of the status word identify a device as RL01 or RK06/07 respectively.

4.2.1.4 Keyboard Monitor Commands - Three of the keyboard monitor commands are changed in RT-11 V03B.

When you specify SHOW without any option, SHOW does not require the RESORC program. It prints a list of devices on the system and gives their current status, as it did in Version 3. This form of the command no longer displays USR SWAP/NOSWAP information.

The SHOW command has new options that allow you to access the RESORC program. The three new options to the SHOW command all call the RESORC utility program. These options are as follows:

1. CONFIGURATION
    Displays information about the monitor status, USR SWAP/NOSWAP, type of processor, special hardware, and SYSGEN options that are in effect.

2. DEVICES
    Displays device information, including current status and vectors.

3. TERMINALS
    Displays terminal information, including unit number, type, and SET options enabled for multi-terminal systems.

The DIRECTORY/VOLUMEID command calls the new DIR /V option, rather than the DUP /V option, as it did previously. It prints the volume identification and owner name as part of the header of a volume directory listing.

The syntax of the SET TT SCOPE command is the same as in Version 3. However, this command is now valid for the SJ monitor.

4.2.1.5 Distributed Monitors - RT-11 distribution kits do not include XM monitors or XM device handlers. You must perform a SYSGEN to create these files. Monitors created by the system generation process print (S) as part of the monitor identification message at bootstrap time.

The FB light pattern is not part of the distributed FB monitor.   This saves several words for users of the distributed FB monitor.

4.2.2 Monitor Problems Corrected

The following problems with the Version 3 monitor are corrected in Version V03B.

The V03 XM monitor failed when a large foreground program was run or when a number of large device handlers were made resident with the LOAD command. The problem occurred because the USR moved down into the area of memory mapped by the memory management page address register 1 mapping register (20000-40000).   The following message prints if this situation occurs:

        ?KMON-F-Not enough memory.

Two problems are fixed in the XM monitor that prevented virtual jobs from .CHAINing to another program or from passing command lines to the KMON in an .EXIT.

A patch is no longer required to make .DSTATUS work properly in the XM monitor.   In V03, a problem with .DSTATUS caused incorrect results when a program used extended memory programmed requests to manipulate program address windows.

The patches required in V03 for multi-terminal support are not necessary in V03B.

The .MTIN and .MTOUT programmed requests did not execute properly in V03 if the character count was greater than 127. V03B corrects this problem.

For RT-11 V03 monitors created by SYSGEN, there was a problem with the .TWAIT programmed request.   If a .CMKT request with an ID of 0 was issued during a completion routine while the program was suspended by .TWAIT, under some conditions the program never returned from the .TWAIT. This problem is corrected in V03B.

For V03 monitors created by SYSGEN, there was a problem with support for the 50 Hz system clock.

A problem with support for the KW11-P programmable clock is corrected.

A problem is corrected for FB and XM monitors that concerned the interrupt exit code. This problem occasionally left a fork element on the queue until the next interrupt occurred. The visible effect of this problem was that the LA180 line printer ran slower on PDP-11/03 systems when .WRITC requests were used.

A problem in the address checking code for XM monitors is corrected.

A minimum amount of patch space is now created in the monitor file. The bootstrap reclaims unused patch space when it loads the monitor.

A problem in the XM monitor for the .CRAW request is corrected.   The problem caused .CRAW with an implied .MAP request to permit a second window to overlap window 0.

A problem is corrected that caused systems with DL11-E remote units to crash.

A problem with the SET TT CONSOL command is corrected.

A problem with the FRUN /T command is corrected.

The FB and XM monitors handle the close brace and tilde characters correctly.

4.2.3 Monitor Restrictions

The SET FORTRAN and SET WILDCARDS commands accept only the first six characters of the words FORTRAN and WILDCARDS. Entering more than six characters causes an error and the system aborts the command. Abbreviate these two commands as SET FOR and SET WILD.

GT ON and GT OFF commands are illegal if multi-terminal support is included in a monitor created by SYSGEN. However, EDIT is still able to use a display terminal when you issue the ED command.

In a multi-terminal system, it is not possible to do a .READ, .READC, or .READW to the terminal. This is an unusual function and does not affect many users. Use EDIT or TECO to create a file instead of a .READx to TT:.

The following restriction applies to users with RK06, RK07, or RL01 disks. If you initialize your disk for automatic bad block replacement, a problem can occur when you copy the monitor to the disk. Since the bootstrap cannot support replacement bad blocks, the disk is not bootable if the monitor falls over a bad block. The following message prints on the console at boot time:

    ?BOOT-F-I/O error

To correct this situation, rename the monitor, giving it a dummy name such as TEMP.MON. Copy TEMP.MON to another location on the disk, calling the resulting output file by the monitor's correct name. Delete the temporary file. The space it occupied, including the bad block and its replacement, can be used by any other file. Place the monitor file as near as possible to the beginning of the disk so it will not move when a SQUEEZE command is issued. It is good practice when creating a system disk to copy the monitor, the bootstrap, and SWAP.SYS (and TT.SYS, if an SJ monitor) first. Then check to see if the volume is bootable before you copy any other files to it.

If a user program passes a command string to .CSIGEN that contains a syntax error, such as an illegal file name, the monitor .ENTERs the output file and returns an error to the user program. Immediately re-issuing the .CSIGEN causes the file to be closed with a length of 0. This problem can be avoided by issuing a .PURGE request for each of channels 0, 1, and 2 whenever .CSIGEN reports an error, and before re-issuing the .CSIGEN. This procedure eliminates the zero-length files.

The TRAP instruction is not available for programmers to use if FORTRAN is also being used. FORTRAN is considered to be "the user" of the TRAP instruction.

4.2.4 PIP Changes

The PIP system utility program is changed slightly in V03B. The /N
and /O options (and also the monitor COPY/NOREPLACE and COPY/PREDELETE
commands) now accept cassette and magtape files as valid input files.
Cassette and magtape files are not legal for output.

4.2.5 PIP Problems Corrected

A problem reading Version 2C magtapes is corrected.

4.2.6 DUP Changes

The DUP system utility program is changed slightly in V03B to support
replacement bad block handling for the RK07 and RL01 disks.   The /R
option (and also the monitor INITIALIZE/REPLACE command) is now valid
for RK07 and RL01.

DUP has also been restructured to provide increased buffer size for
multi-block transfers.   This change results in faster transfer rates
on all RT-11 systems. Because DUP is now overlaid, the volume from
which you run DUP must remain mounted unless you use the /W option.
See Chapter 8 of the RT-11 System User's Guide for details.

DUP now checks for SWAP.SYS on volumes it boots, and prints an error
message if the file is not present.

DUP checks the length of the volume identification and owner name when
it initializes a volume.   If the name you specify is too long, DUP
truncates the name to the correct length and prints a warning message.

4.2.7 DUP Problems Corrected

When the /C option was used to create a file and no length was
specified for the file, DUP supplied a file size of 0. This problem
is corrected in V03B. DUP uses one block as the default file size,
and supplies a null file type if you do not specify one.

4.2.8 DIR Changes

The DIR system utility program has a new option, /V. You can access
this new option through the monitor DIRECTORY/VOLUMEID command. This
option prints the device volume identification and owner name as part
of the header for a directory listing.

4.2.9 DIR Problems Corrected

In V03, a command of the following form (where the output device is
magtape or cassette and no file name is given) destroyed the directory
area of the output device:

    MT:=input-device-name:.*/option

In V03B, the same command generates the following output file specification:

    MT:input-device-name.DIR

A problem in reading Version 2C magtapes is corrected.

4.2.10 SYSGEN Changes

The SYSGEN program itself is changed in V03B. The program can now check the validity of responses to the dialogue questions. The ASK command (see Section B.3.6 of the RT-11 System Generation Manual) provides a check for valid responses based on the following data types:

    /A   Alphanumeric

    /D   Decimal number

    /O   Octal number

    /Y   Yes or No response

In addition, there are high limit and low limit options that define the valid range for numeric responses.   The #ASK command is also changed to allow two-line questions.

There are two new SYSGEN commands. The #IFGT command (see Section B.3.3 of the RT-11 System Generation Manual) tests the value of the associated conditional.   If the value is greater than 0,   the conditional is true.   If the value is less than or equal to 0, the conditional is false. The #DECR command (see Section B.3.13 of the RT-11 System Generation Manual) decrements the value associated with the specified conditional. You can use #DECR with #IFGT to allow a question to be repeated n times, where n is the value of a response to a dialogue question.

The RT-11 SYSGEN dialogue now offers RL01 and RK07 as system devices.

4.2.11 SYSGEN Problems Corrected

In V03, SYSGEN failed to enable the timer facility in SJ when error logging was also selected.   This problem is corrected in V03B. In addition, it is no longer necessary to patch the SYCND file.

4.2.12 PATCH Changes

It is no longer necessary to use /M to patch a monitor file.   The /M option now has no meaning to the PATCH system utility program.

Use /A for monitor patches.

4.2.13 BATCH Problems Corrected

In V03, BATCH printed the "?BATCH-F-Batch fatal error" message when BATCH was run under the XM monitor and the file BA.SYS was not present on the system device. This problem is corrected in V03B.

4.2.14 MACRO Problems Corrected

In V03, the .NARG directive returned the wrong argument count if a dummy argument was present. V03B corrects this problem.

4.2.15 EDIT Changes

EDIT now handles brace and tilde characters without requiring a patch.

4.2.16 EDIT Problems Corrected

A problem in EDIT that caused fatal errors on erroneous EB, ER, and EW commands under the XM monitor is corrected in V03B.

4.2.17 LINK Problems Corrected

Absolute program sections (.PSECT name, ABS) can now be used correctly. They do not allocate space in the image file. No information within an absolute program section is stored in the image file. The overlay (OVR) attributes work as defined for the sections, but the CON does not work because of a MACRO restriction.

LINK and FRUN correctly handle relocation information for an overlaid foreground program that is linked using library files.

The linker produced a bad load module (SAV, REL, or LDA) if an input module had an empty GSD record (that is, a GSD record containing no four-word Global Symbol Definition items). Typical symptoms of this problem included traps to 4 or 10, incorrect program execution, etc. Normally, object modules do not contain null GSD records, but if an object module producing program did create one, the resulting link was most likely invalid. This problem is corrected in V03B.

4.2.18 FILEX Problem Corrected

A problem with diskettes in interchange format (/U) caused transfers to fail when there was no system date issued.

It was possible to duplicate a file specification when transferring files to diskettes in interchange format (/U).

4.2.19 ODT Restriction

On-line Debugging Technique (ODT) does not interact properly with a multi-terminal system. It echoes twice, and its output is directed to the actual console terminal. If the multi-terminal monitor has time-out support, ODT does not work at all.

4.2.20 **MDUP Problem Corrected**

MDUP could work on only the RK05 disk. That problem is corrected in V03B.

4.2.21 **MACRO Restriction**

MACRO does not pass the correct information to the linker for certain .PSECT constructions.

The construction

```
.ASECT      .ABS
.PSECT      name, ABS, CON
```

is invalid.

The construction

```
.ASECT      .ABS
.PSECT      name, ABS, OVR
```

is valid.

4.2.22 **PAT Problem Corrected**

PAT (the object module patch program) gave a fatal error message if any of the input object modules had an empty record. Normally, object files do not contain null object records. This problem is corrected in V03B.

4.2.23 **FILEX Restriction**

FILEX does not transfer files to or from the RX02 diskette device (device DY:).

4.2.24 **RX02 Monitor Patch**

This information is important if you are creating a DY monitor through the system generation process. Disregard this section if you are using one of the distributed DY monitors.

When you run the SYSGEN dialogue you are asked for the RX02 control and status register (CSR). You can type an address in response to the dialogue question, thereby defining the symbol DY$CSR. Normally, you can also type just a carriage return in response to the question in order to select the default CSR, which is 177170 in this case.   When you select the default value by this method, SYSGEN leaves the symbol DY$CSR undefined so that it can be supplied later by the bootstrap routine.

At run time, the bootstrap uses conditional assembly directives to check symbols such as DY$CSR. If a symbol is undefined, the bootstrap gives it the default value by using a direct assignment statement. However, in the case of DY$CSR the bootstrap assigns the value 177164, which is incorrect. The observable result of this occurrence is that the system cannot run.

There are two ways of circumventing this problem. One way is to specifically type the RX02 CSR address when the SYSGEN dialogue program asks for it. The other way to solve this problem is to modify the bootstrap routine so that DY$CSR is correctly defined.

If you choose to modify the bootstrap, make the following changes to the file BSTRAP.MAC before you perform the system generation.

```
.R EDIT (RET)
*EBBSTRAP.MAC[57] (ESC) R (ESC) (ESC)
*FDY$CSR (ESC) V (ESC) (ESC)
.IIF NDF DY$CSR, DY$CSR == 177164
*G164 (ESC) =C170 (ESC) V (ESC) (ESC)
.IIF NDF DY$CSR, DY$CSR == 177170
*EX (ESC) (ESC)
```

The patch shown above does not alter the BSTRAP module's edit level, nor does it affect the monitor's version number.

4.2.25 RX01 Software Kits

RX01 software kits distributed after April 30, 1979 include all patches published through that date. You should not install any patches published before April 30, but you should install subsequent patches, as usual.

4.3 IMPORTANT MEMORY LOCATIONS IN V03B

Table 4-2 lists the specific memory locations referenced in RT-11 documentation. The locations are documented for the purposes of modification; therefore, they do not represent actual memory locations, but addresses in the disk file as accessed by PATCH. The symbols listed in the table are globals that will appear in a link map if a customized system is built with SYSGEN.

Table 4-2
RT-11 V03B Memory Locations

Location	Monitors		Other
	BL/SJ	FB	
ALTMDE	-	-	10755 (EDIT.SAV)
CONFIG	34300	34300	-
CSR1DY	-	-	2326 (FORMAT.SAV)
CSR2DY	-	-	2330 (FORMAT.SAV)
DSARG	-	-	17572 (EDIT.SAV)
GTVECT	34354	34354	-
LINPPG	-	-	22634 (MACRO.SAV)
LINPPG	-	-	4656 (MAC8K.SAV)
LINPPG	-	-	4324 (CREF.SAV)
MAXBLK	34314	34314	-
RP23	43026	53476	1116 (DP.SYS)
SYSLIB	-	-	7270 (LINK.SAV)

Location	Monitor (same for SJ and FB)										
	RK	DT	DX	DY	DL	DP	DS	RF	DM	DD	PD
BHALT	720	764	1002	774	1012	1002	724	712	756	766	744

	Monitors			
	RFMNSJ	RFMNFB	DSMNSJ	DSMNFB
SYDVSZ	34502	34570	34502	34570

March 1979

READER'S COMMENTS

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

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

_____
_____
_____
_____
_____
_____
_____
_____
_____
_____
_____
_____
_____
_____

Did you find errors in this manual? If so, specify the error and the
page number.

_____
_____
_____
_____
_____
_____
_____
_____
_____
_____
_____
_____
_____
_____

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

☐ Assembly language programmer
☐ Higher-level language programmer
☐ Occasional programmer (experienced)
☐ User with little programming experience
☐ Student programmer
☐ Other (please specify)_____

Name_____ Date_____

Organization_____

Street_____

City_____ State_____ Zip Code_____
                                                 or
                                            Country

Please cut along this line.