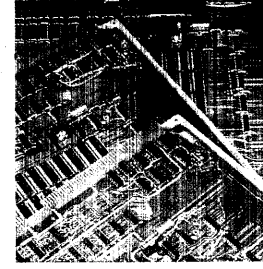
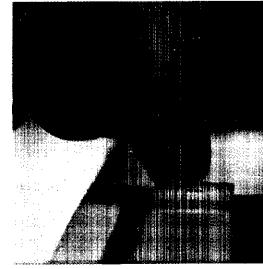
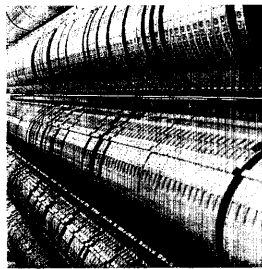
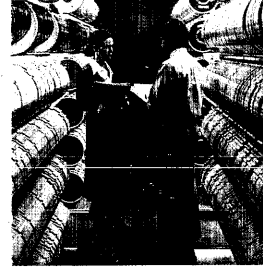
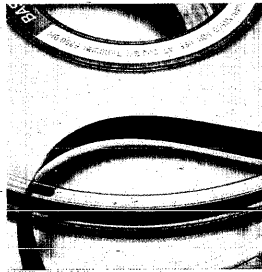


Prime Computer, Inc.

MRU4304-007
Software Release
Document
Rev. 18.4



Software Release Document

MRU4304-007

Rev. 18.4

by

Emily M. Stone

This document contains information on technical changes and enhancements made to Prime user software after Rev. 18.3 and released at Rev. 18.4.

**Prime Computer, Inc.
500 Old Connecticut Path
Framingham, Massachusetts 01701**

COPYRIGHT INFORMATION

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

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

Copyright © 1982 by
Prime Computer, Incorporated
500 Old Connecticut Path
Framingham, Massachusetts 01701

PRIME and PRIMOS are registered trademarks of Prime Computer, Inc.

PRIMENET, RINGNET, and THE PROGRAMMER'S COMPANION are trademarks of Prime Computer, Inc.

TELENET is a registered trademark of Telenet Communications Corporation.

MEDUSA is a trademark of Cambridge Interactive Systems Limited, Cambridge, England. The program MEDUSA is a copyrighted product of Cambridge Interactive Systems Limited, Cambridge, England © 1981.

HOW TO ORDER TECHNICAL DOCUMENTS

U.S. Customers

Software Distribution
Prime Computer, Inc.
1 New York Ave.
Framingham, MA 01701
(617) 879-2960 X2053, 2054

Customers Outside U.S.

Contact your local Prime
subsidiary or distributor.

Prime Employees

Communications Services
MS 15-13, Prime Park
Natick, MA 01760
(617) 655-8000 X4837

INFORMATION Systems

Contact your Prime
INFORMATION system dealer.

PRINTING HISTORY — SOFTWARE RELEASE DOCUMENT

<u>Edition</u>	<u>Date</u>	<u>Number</u>	<u>Documents Rev.</u>
	July, 1981	MRJ4304-005	18.2
	January, 1982	MRJ4304-006	18.3
	July, 1982	MRJ4304-007	18.4

SUGGESTION BOX

All correspondence on suggested changes to this document should be directed to:

Emily M. Stone
Technical Publications Department
Prime Computer, Inc.
500 Old Connecticut Path
Framingham, Massachusetts 01701

Contents

1	INTRODUCTION	
	Rev. 18.4 Software Release Document	1-1
	Overview of Rev. 18.4	1-2
	New Book Titles	1-4
	Installation of Rev. 18.4	1-7
2	SYSTEM ADMINISTRATOR	
	Booting From Magnetic Tape	2-1
	Notes on Disk Drives	2-13
3	LANGUAGES	
	BASIC/VM	3-1
	COBOL	3-9
	DBG (Source Level Debugger)	3-13
	FORTRAN (FTN)	3-19
	FORTRAN 77	3-23
	Pascal	3-25
	PL/I, Subset G	3-27
	PMA	3-31
	RPG	3-33
	VFINLIB	3-35
	VRPG	3-37
4	PRIMOS AND UTILITIES	
	PRIMOS	4-1
	Batch	4-9
	CPL	4-11
	Editor	4-15
	EMACS	4-17
	FUTIL	4-19
	LOGPRT	4-21
	MAGNET	4-23
	MAGSAV/MAGRST	4-29
	PHYSAV/PHYRST	4-33
	RUNOFF	4-35
	SEG	4-37
	Subroutines	4-41

5	DATA MANAGEMENT SYSTEMS	
	DBMS	5-1
	DBMS/QUERY	5-9
	MIDAS	5-11
	PRIME/POWER	5-13
6	FORMS	
	FORMS	6-1
7	COMMUNICATIONS	
	PRIMENET	7-1
	DPTX	7-3
8	THE FILE TRANSFER SERVICE (FTS)	8-1
9	FTS: FTR OPTIONS REFERENCE	9-1
10	FTS FOR SYSTEM OPERATORS	10-1
11	FTS FOR SYSTEM ADMINISTRATORS	11-1

OVERVIEW OF REV. 18.4

New Products

Rev. 18.4 introduces three major new products:

- VRPG, Prime's V-mode RPG II compiler
- The File Transfer Service (FTS), a utility for transferring files over a network
- EMACS, a customizable screen editor

VRPG: VRPG is significantly faster than RPG (the R-mode RPG II compiler), and provides improved error messages. The source level debugger (DBG) has been expanded to support VRPG. For further information, refer to the sections on VRPG and DBG in Chapter 3, and also to the new RPG II V-mode Compiler Reference Guide (IDR5040), which is described later in this chapter.

File Transfer Service (FTS): The new FTS utility transfers files between Prime computers over a network. Numerous command line options allow the user to monitor and control transfer requests. Full information on FTS is supplied in Chapters 8 through 11 of this book. This information will be incorporated into the PRIMENET Guide at Rev. 19.

EMACS: EMACS is a self-documenting, customizable screen editor that is compatible with ED, RUNOFF, and other Prime products that use text files. For more information, refer to the section on EMACS in Chapter 4 and to the three new EMACS manuals described later in this chapter.

Enhancements to Existing Software

Enhancements have been made to the following software products:

COBOL	Pascal
CPL	PHYSAV/PHYRST
DBG	PL/I, Subset G
DBMS	PRIMENET
DBMS/QUERY	PRIME/POWER
DPTX	PRIMOS
LOGPRT	SEG
MAGNET	Subroutines
MAGSAV/MAGRST	VFINLIB

Problems Fixed

Problems in the following products have been corrected:

BASIC/VM	LOGPRT
Batch	MAGSAV/MAGRST
COBOL	MIDAS
CPL	Pascal
DBG	PHYSAV/PHYRST
DBMS/QUERY	PRIMENET
DBMS	PRIME/POWER
DPTX	PRIMOS
Editor	RPG
FORMS	RUNOFF
FORTRAN	SEG
FORTRAN 77	Subroutines
FUTIL	VFINLIB

Documentation Changes

This publication contains documentation corrections, clarifications, and additions for manuals on the following products:

BASIC/VM	Pascal
CPL	PL/I, Subset G
DBMS	PMA
DBMS/QUERY	PRIME/POWER
DPTX	PRIMOS
FORTRAN	Subroutines
MAGSAV/MAGRST	VFINLIB
MIDAS	

Problems Outstanding

This document outlines problems outstanding in the following products:

DBG	FORTRAN 77
DBMS	MIDAS
DBMS/QUERY	PRIME/POWER
DPTX	SEG
FORTRAN	

NEW BOOK TITLES

The following new technical publications are now available.

RPG II V-mode Compiler Reference Guide (IDR5040)

This book is a comprehensive guide to Prime's V-mode RPG II. Designed to help RPG II programmers adapt quickly to Prime systems, the book assumes a knowledge of RPG II and of programming in general. Topics include the compiling, loading, and running of RPG II programs; conversion guidelines; and specification forms.

EMACS Primer (IDR6107)

The EMACS Primer introduces the EMACS screen editor and presents the basic EMACS commands in tutorial fashion. After reading this book, users will know enough EMACS commands to accomplish most editing tasks.

EMACS Reference Guide (IDR5026)

The EMACS Reference Guide is a reference manual for EMACS commands. This book describes all EMACS commands, and is intended primarily for users who have read the EMACS Primer.

EMACS Extension Writing Guide (IDR5025)

This book describes how programmers can create new editing commands and features for EMACS. The book describes the Prime EMACS Extension Language (PEEL), and teaches the reader how to incorporate user-written programs into the EMACS environment. Familiarity with the EMACS Reference Guide is assumed.

Magnetic Tape User's Guide Update, Rev. 18.4 (UPD5027-184)

This update to the Magnetic Tape User's Guide documents changes and enhancements to Prime's tape-handling software at Rev. 18.4. For summaries of the changes, refer to Chapter 4 of this MRU (specifically, to the sections on MAGNET, MAGSAV/MAGRST, and PHYSAV/PHYRST).

The following four books are new to the INFORMATION documentation collection as of Release 5.1 of INFORMATION software. (Release 5.1 of INFORMATION is based on Rev. 18.3 of PRIMOS.)

INFORMATION Administrator's Guide (IDR5131)

This guide documents the administration of INFORMATION systems as of Release 5.0 of INFORMATION software. It describes typical Administrator tasks and responsibilities, such as creating new user accounts and monitoring file efficiency. Major topics include procedures for system startup and shutdown, methods of account and file management, requirements for serial printers, and allocation of catalog space.

INFORM Reference Guide (PDR3905)

This new INFORM guide documents all the verbs and keywords associated with the INFORM processor. It is based on Release 5.0 of INFORMATION software. This book is a complete rewrite of the previous IDR version of the same title. It combines tutorial and reference material, contains many examples, and describes everything from defining a file with ENTRO to producing reports. The book is designed for users who are familiar with the basic terms and concepts described in the INFORMATION Highlights Companion (FDR5212).

INFORMATION Editor Reference Guide (PDR3904)

The INFORMATION Editor Reference Guide documents the Editor's features and commands as of Release 5.1 of INFORMATION software. This book is a complete rewrite of the previous IDR version. It describes what the Editor is, what to use it for, all of the associated terms and conventions, and all of the Editor's commands and special features. Like the INFORM guide, this book is a combination of tutorial and reference material, making it an easy-to-use guide for both new and experienced users.

INFO/BASIC Update, Release 5.0 (PIU2600-082)

The INFO/BASIC Update describes all the enhancements, additions, and modifications to the INFO/BASIC software since Release 2.5. It also includes corrections to the INFO/BASIC Reference Guide (PDR3903). This information is complete through Release 5.0.

INFORMATION System Update, Release 5.1 (MRU5964-001)

This Minor Release Update documents the changes, enhancements, and new additions to Prime INFORMATION software as of Release 5.0 and Release 5.1. Also included are corrections to the INFO/Basic Reference Guide (PDR3903), PERFORM Reference Guide (PDR3906), and INFORM Reference Guide (PDR3905), as well as descriptions of improvements of PRIMOS at Rev. 18.2 that are relevant to Prime INFORMATION.

The following six books document Release 2.1 of the Prime Office Automation System (OAS).

OAS Word Processing Guide (IDR5020) (for the PT45 terminal)

OAS Word Processing Guide (IDR5021) (for the PT65 terminal)

These books are intended for the beginning OAS user. No background in programming, computer operations, or word processing is required. Knowledge of the Management Communications and Support module of OAS is not required. A separate book is provided for each of the two types of terminals (PT45 and PT65).

The OAS Word Processing Guide provides complete instructions on the use of the Word Processing module of Prime's OAS. The Word Processing functions include creating, editing, and printing documents. This book also provides instructions for using the list processing function to create personalized form letters and special reports. In addition to tutorial material, the book includes an alphabetical reference section summarizing all functions.

OAS Management Communications and Support Guide (IDR5022)
(for the PT45 terminal)

OAS Management Communications and Support Guide (IDR5023)
(for the PT65 terminal)

These books are intended for the beginning OAS user. No programming or computer operations background is required. There are a few functions that may require knowledge of the Word Processing module of Prime's OAS. A separate book is provided for each terminal type (PT45 and PT65).

The OAS Management Communications and Support Guide provides complete instructions for using all functions of the Management Communications and Support module of Prime's OAS. These functions include sending, receiving, and filing electronic mail and maintaining an automated calendar of appointments.

Familiarity with the appropriate OAS Word Processing Guide will allow better utilization of all available MCS functions.

OAS Advanced Text Management Guide (PT65) (IDR4353)

This book is intended for the experienced word processing user at an installation that uses the Advanced Text Management module of Prime's OAS. Familiarity with the OAS Word Processing Guide (PT65) is assumed.

This book provides instruction for all functions of Advanced Text Management. These functions include proofreading, translating words from or into selected foreign languages, and searching documents in a database index. This book is written specifically for the PT65 terminal.

OAS System Administrator's Guide (IDR4354)

This book is intended for the OAS System Administrator who is familiar with all OAS functions but does not necessarily have a background in computer programming or operations. Familiarity with the OAS Word Processing Guide, the OAS Management Communications and Support Guide, and the OAS Advanced Text Management Guide is assumed.

This book provides information on all System Administrator functions for OAS, including creation and maintenance of the system calendar, the network directory, user records, and the document data base. It also gives instructions for monitoring system usage, troubleshooting printer problems, and performing other tasks related to the administration of an OAS.

INSTALLATION OF REV. 18.4

The Rev. 18.4 Master Disk is in complete Master Disk format, rather than in the update format that is normally used at minor (dot) revisions. The Rev. 18.4 Master Disk contains all Master Disk software, including the software that has not changed since the initial release of Rev. 18 (Rev. 18.1). Rev. 18.4 can therefore be installed without prior installation of Rev. 18.1. For example, Rev. 18.4 can be installed directly on a Rev. 17 system.

CHAPTER 2

SYSTEM ADMINISTRATOR

BOOTING FROM MAGNETIC TAPE

INTRODUCTION

The procedure for booting from magnetic tape has been improved at Rev. 18.4. This section contains:

- A summary of the new magnetic tape boot features
- Background information on booting from magnetic tape (including the meanings of the octal parameters you supply)
- Instructions for performing a standard magnetic tape boot
- Explanations of magnetic tape boot error codes

SUMMARY OF NEW FEATURES

The following changes have been made to the magnetic tape boot procedure:

- The new boot procedure offers a choice of ways to control relocation of the MIBOOT program. (After sizing memory, MIBOOT relocates itself out of the way of the program that is to be loaded, if necessary.)
- The new procedure makes it easier to boot from supervisor terminals that do not run at the default speed. (The default terminal speed for magnetic tape boots is now 30 characters per second [cps].)

The octal parameter that you supply during a boot has three newly-defined bit settings (bits 8, 9, and 13) that allow you to take advantage of the above improvements.

Relocation of the MTBOOT Program

There are now two ways to direct relocation of MTBOOT:

- If bit 13 is 0, then bits 8 and 9 control the relocation of MTBOOT. The following table shows how bits 8 and 9 determine the new ending address of MTBOOT:

<u>Bit 8</u>	<u>Bit 9</u>	<u>New Ending Address of MTBOOT</u>
0	0	End of physical memory
0	1	16K
1	0	32K
1	1	48K

- If bit 13 is 1, the new ending address of MTBOOT is determined by the A-register of the save vector of the program being read in. In this case, bits 8 and 9 must both be 0.

Booting from Non-30-CPS Supervisor Terminals

If your supervisor terminal does not run at 30 cps, the following steps must be taken:

1. The boot program (MTBOOT) must be modified to accommodate the alternate speed. Terminal speed is determined by the SOC and OPTION A control words of MTBOOT. You can either temporarily patch these control words during the boot, or change them permanently.

To patch the SOC and OPTION A control words temporarily, set bit 11 in the octal parameter before starting the boot procedure. This will cause MTBOOT to halt at '260 with halt code '21 after sizing memory. (The halt code is displayed in the data lights or at the supervisor terminal.) Then modify locations '220 - '222 for the appropriate speed, according to the following table:

<u>Speed</u>	<u>'220</u>	<u>'221</u>	<u>'222</u>
110 baud	110	27	74006
300 baud	1010	76	34006
1200 baud	2010	373	34006
9600 baud	3410	3735	34006

After performing the patch, select RUN and press START to continue.

To modify the SOC and OPTION A control words permanently, issue the following commands:

```
REST MIBOOT.SAVE
SA MIBOOT.SAVE 4/<value_1> <value_2> <value_3>
R IBOOT.SAVE
```

Choose value_1, value_2, and value_3 from the table of patch values above. Value_1 through value_3 correspond to the values for locations '220 through '222, respectively.

The program MAGRST>IBOOT.SAVE incorporates MIBOOT.SAVE into the MAGSAV runfile. That is, IBOOT.SAVE restores MAGSAV.SAVE, overlays MIBOOT.SAVE, and saves MAGSAV.SAVE.

2. If PRIMOS II (DOS) is to be loaded, then it must also be modified to accommodate the alternate speed. If you set bit 13 of the octal parameter, this modification is done automatically for you, as in the case of a disk boot. After DOS is loaded, but before it is started, the SOC and OPTION A control words from MIBOOT are copied into locations '1004 - '1006 of DOS. (Location '725 is always 0.) In addition, the save vector keys are copied into the save vector A-register.

If you do not set bit 13, you must modify DOS yourself. To accomplish this, set bit 10 of the octal parameter when PRIMOS II (*DOS64) is being loaded into memory. MIBOOT will then halt with halt code '000001 after loading in PRIMOS II. At this point, load '4000 into the A-register and set locations '1004, '1005, and '1006 to the appropriate values from the table of patch values, above. Finally, press START. PRIMOS II will reset the terminal to the desired speed.

BACKGROUND INFORMATION

The general sequence of events during a boot from magnetic tape is as follows:

1. Mount a tape that was saved by the MAGSAV utility and that contains the program(s) you wish to load into memory (for example, PRIMOS II [DOS], MAGRST, MAKE).

Note

It is now possible to boot from any magnetic tape produced by MAGSAV Rev 14.1 or later.

2. Initiate the boot. If your computer has a Virtual Control Panel (VCP), do this by typing the command BOOT followed by an octal number. If your machine does not have a VCP, initiate the boot by setting your control panel sense switches to an octal number, turning the rotary switch to LOAD, and pressing

START. In either case, the octal value you supply determines the nature of the boot (as described below).

3. A short bootstrap program that resides in the computer's Programmable Read-Only Memory (PROM) takes control. This bootstrap starts up the tape drive, ensures that the tape is set at a loadpoint, and reads in the program MTBOOT from tape. (MTBOOT was placed on the tape by MAGSAV.)
4. The MTBOOT program takes over. Its major functions are to size memory and to read in a file (for example, DOS) from tape. (You specify which file MTBOOT reads in, as described below.)
5. When the requested file is found, the first data block is read and the save vector examined. The file must have a starting address above '6207 or an ending address below the memory size minus '5423 (for example, '172355 for a 64K system). If necessary, MTBOOT relocates itself out of the way of the program being loaded (thus adjusting effective memory size).
6. If MTBOOT encounters an error when reading from the tape, it halts with the status word in the data lights or displayed at the supervisor terminal. These halt codes are explained at the end of this section.

The Octal Parameters

On VCP-equipped computers, the octal number that you type in after the BOOT command consists of up to 6 digits. (BOOT xx is interpreted as BOOT 0000xx.) On non-VCP-equipped computers, each of the 16 control panel sense switches represents one bit of the octal value. A sense switch in the up position represents a bit that is set (1), and a sense switch in the down position represents a bit that is reset (0). To translate between the sense switches (bits) and octal digits, use the following correspondence:

Sense Switches (Bits)															
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
	└──────────┘			└──────────┘			└──────────┘			└──────────┘			└──────────┘		
1	2			3			4			5			6		
Digits															

The leftmost digit can be 0 or 1; the other digits can range from 0 through 7. For example, the octal number '014114 would be entered on a non-VCP machine by setting sense switches 4, 5, 10, 13, and 14.

The meanings of the octal values for magnetic tape boots are as follows:

Bits 1-7 specify the number (on the logical tape) of the file to be loaded, the first file being number 1. Calculating the position of a file on tape is easiest if the tape has been indexed by MAGSAV. The

numbers of the files correspond to the order of files in the index. Note that if subdirectories have been saved, each subdirectory itself has a file on the tape.

If bits 1-7 are 0, `MTBOOT` produces the prompt:

```
TREENAME=
```

and waits for you to supply the pathname of the file to be loaded. The pathname you supply must be a series of up to eight valid file system names, with no embedded blanks, separated by ">" signs. If you enter a null line, a question mark, or any character whose value is less than '240 (for example, a control character) the prompt is repeated. The file to be loaded must be in standard save-file format.

Note

To load from a logical tape other than the first, you must skip over the first logical tape. Do this by specifying a non-existent file number or pathname. When `MTBOOT` halts with code '000005 displayed at the supervisor terminal or in the data lights, reset the switches (if necessary) and restart at '400.

Bits 8 and 9 control the relocation of `MTBOOT`, as described in the summary of new features, above.

Bit 10 determines whether or not the program loaded in from tape is automatically started. If bit 10 is 0, `MTBOOT` starts the program as soon as it is loaded. If bit 10 is 1, `MTBOOT` halts with a code of '000001 after loading in the program, and you must start the program yourself. (The halt code is displayed at the supervisor terminal or in the control panel data lights.)

Bit 11 allows you to patch the `MTBOOT` program to accommodate a non-30 cps supervisor terminal. If bit 11 is 1, `MTBOOT` halts at '260 before prompting for the pathname of the file to be loaded. You can then patch the SOC and OPTION A control words for alternate terminal speeds, as described in the summary of new features, above.

Bit 12 selects between 7-track and 9-track drives. If bit 12 is 0, a 9-track drive is assumed; if bit 12 is 1, a 7-track drive is assumed.

Bit 13 determines how relocation of `MTBOOT` will be controlled, as described in the summary of new features, above.

Bits 14, 15, and 16 should always be 1, 0, 1. (That is, the rightmost digit of the octal parameter should always be 5.) This setting specifies that the boot is from magnetic tape.

Note

If your computer has sense switches rather than a VCP, switches 8-10 and 12-13 must be left in position until the selected file has been loaded. The other switches may be reset any time after the pathname prompt has been issued or the tape search has started.

THE STANDARD MAGNETIC TAPE BOOT

The instructions provided below are for standard magnetic tape boots. In particular, the instructions given here assume that you are booting from a 9-track tape drive and that your supervisor terminal runs at 30 cps. If either assumption is false, you will need to use different octal values than those given in these instructions.

Three sets of instructions are supplied here:

- How to create a tape that can be used to boot the system
- How to boot from magnetic tape on a computer with a control panel
- How to boot from magnetic tape on a computer with a VCP

Creating a Tape for Booting the System

If your master disk is intact, all you need on your boot tape is PRIMOS II. However, if you must remake and restore your master disk, you will need to have MAKE, MAGRST, and possibly a few other utilities on your tape. The following is one scenario for creating a tape from which to boot the system and remake the master disk.

```

OK, /* Save the necessary files in a scratch directory
OK, CREATE WORK
OK, A *>WORK
OK, FUTIL
[FUTIL rev 18.4]
> FROM DOS
> COPY *DOS64
> FROM CMDNC0
> COPY MAKE
> COPY MAGRST
> COPY COPY_DISK
> COPY PHYRST
> Q
OK, AS MT0 /*Assign the tape deck
Device MT0 assigned.
OK, /* Save the contents of the WORK directory. Take an index
OK, /* so that you will know which file numbers correspond

```

```

OK, /* to which files.
OK, MAGSAV
[MAGSAV Rev. 18.4]
Tape unit (9 Trk): 0
(Tape not at load point)
Enter logical tape number: 1
Tape name: BOOT
Date (MM DD YY):
Rev no: 18.4
Name or command: $I 7
Name or command: *
*** Start of Save ***
*DOS64 (sam)
MAKE (sam)
MAGRST (sam)
COPY_DISK (sam)
PHYRST (sam)
***End of Save ***

```

To minimize the risk of read errors, save the tape with the minimum number of required utilities. The whole of PRIMOS could be saved in a separate logical tape, or PRIMOS could be restored from another backup tape.

Booting from Tape on a Computer with a Control Panel

Use the procedure described below to boot from tape if your system has a control panel rather than a VCP.

Load PRIMOS II Into Memory:

1. Mount the magsaved tape on drive 0. The tape should be at load point or beyond.
2. Put the tape drive on line.
3. Set the sense switches to '15 (switches 13, 14, and 16 up; all others down).
4. Set the rotary switch to LOAD.
5. Press START. When the system types "TREENAME=", enter the pathname of PRIMOS II exactly as it appears on the index of the magtape. Typically:

```
TREENAME= *DOS64
```

The system responds with:

```

PRIMOS II REV. 18.4 02/17/82 (AT 170000)
OK:

```

6. If your master disk is intact, respond to the PRIMOS II "OK:" prompt by starting up the disk and attaching to its MFD:

OK: STARTUP nnn
 OK: A MFD xxxxxx

(nnn is the disk pack's physical device number)

If you need to remake and restore your master disk, continue with Step 7 when PRIMOS II issues the "OK:" prompt.

Load MAKE Into Memory:

7. Set the rotary switch to STOP.
8. Press MASTER CLEAR.
9. Set the sense switches to '505 (switches 8, 10, 14, and 16 up; all others down).
10. Set the rotary switch to LOAD.
11. Press START. When the system types "TREENAME=", enter the pathname of MAKE exactly as it appears on the index of the magtape. Typically:

TREENAME= MAKE

The system halts with a '1 in the data lights:

HALTED AT 072427: 000001

Run MAKE:

12. Set the rotary switch to STOP.
13. Press MASTER CLEAR.
14. Set the sense switches to '170000 (switches 1 through 4 up, all others down).
15. Set the rotary switch to LOAD.

16. Press START. PRIMOS II responds with the prompt "OK:". Enter "S 1000" to start MAKE, and enter the physical disk number when MAKE requests it:

```

OK: S 1000
GO
*** MAKE *** <Rev. 18.4>
BUILDING NEW PARTITION
PHYSICAL DISK: nnn
:
:
DISK CREATED
OK:

```

Load MAGRST Into Memory:

17. Set the rotary switch to STOP.
18. Press MASTER CLEAR.
19. Set the sense switches to '505.
20. Set the rotary switch to LOAD.
21. Press START. When the system types "TREENAME=", enter the pathname of MAGRST exactly as it appears on the index of the magtape. Typically:

```
TREENAME= MAGRST
```

The system halts with a '1 in the data lights:

```
HALTED AT 072427: 000001
```

Run MAGRST:

22. Set the rotary switch to STOP.
23. Press MASTER CLEAR.
24. Set the sense switches to '170000.
25. Set the rotary switch to LOAD.

26. Press START. PRIMOS II will respond with the prompt "OK:". Enter the STARTUP command with the disk device number of the master disk, and then attach to the MFD on that disk and start at location '1000:

```
OK: STARTUP nnn
OK: A MFD XXXXXX
OK: S 1000
[MAGRST Rev. 18.4]
.
.
.
*** Restore Complete ***
OK:
```

Booting from Tape on a Computer with a VCP

Use the following procedure to boot from tape if your system is equipped with a VCP:

1. Mount the magsaved tape on drive 0. Make sure the tape is at or beyond load point.
2. Boot PRIMOS II:

```
CP> SYSCLR
CP> BOOT 15
TRENAME = *DOS64
```

```
PRIMOS II REV. 18.4 02/17/82 (AT 170000)
OK:
```

3. If your master disk is intact, respond to the "OK:" prompt by starting up the master disk and attaching to its MFD:

```
OK: STARTUP nnn
OK: A MFD xxxxxx
```

(nnn is the physical device number of the disk you wish to start up.)

If you need to remake and restore your master disk, continue with Step 4 in response to the "OK:" prompt.

4. Press the ESCAPE key twice to return to control panel mode:

```
CP> <ESC><ESC>
```

5. Load MAKE:

```
> STOP
CP> SYSCLR
CP> BOOT 505
TRENAME = MAKE
```

HALTED AT 072427/000001

6. Restart PRIMOS II:

```
CP> BOOT 170000
```

7. When the PRIMOS II prompt appears, start MAKE:

```
OK: S 1000
*** MAKE *** <Rev. 18.4>
BUILDING NEW PARTITION
PHYSICAL DISK: nnn
:
:
DISK CREATED
OK: <ESC><ESC>
```

8. Load MAGRST:

```
>STOP
CP> SYSCLR
CP> BOOT 505
TRENAME = MAGRST
```

HALTED AT 072427/000001

9. Restart PRIMOS II:

```
CP> BOOT 170000
```

10. When the PRIMOS II prompt appears, start MAGRST:

```
OK: STARTUP nnn
OK: A MFD xxxxxx
OK: S 1000
[MAGRST Rev. 18.4]
:
:
***Restore Complete***
OK:
```

The contents of the boot tape can now be restored, or a backup tape can be mounted and restored. To run another R-mode program from tape, repeat the procedure indicated in Steps 5-7 (for MAKE) and 8-10 (for MAGRST).

TAPE BOOT ERRORS

The following halt codes may appear in the data lights during a boot from magnetic tape:

- 000001 (Halt at '635 or '175175) The requested program has been loaded and auto-start has been suppressed (bit 10 of the pre-boot octal parameter set). On a machine without a VCP, pressing START and typing "START <address>" starts the loaded program. On a VCP machine, perform an auto-start bootstrap and type "START <address>". (At the halt, A B, X, and so on have been loaded from the program's save vector.)
- 072642 (Halt at '503) The pathname record for the requested file was not followed by a data record.
- 000003 (Halt at '641) The requested program cannot be loaded by the boot program; its starting address is less than '6207 and its ending address is above the memory size minus '5423.
- 000005 (Halt at '440) The end of the logical tape was encountered before the requested program was found.
- 000021 (Halt at '260) Bit 11 of the pre-boot octal parameter was set, causing the boot to halt so that the SOC and OPTION A control words can be patched for a non-30 cps terminal.
- xxxxxx (Halt at '1) A tape read error or unexpected status condition occurred. The data lights display the last status word read from the controller. Status bit definitions are as follows:

100000	Parity error
040000	Runaway tape
020000	CRC error
010000	LRC error
004000	Low DMX range
002000	Permanent error
001000	Read-after-write error
000400	File mark detected
000200	Ready (OK)
000100	Online (OK)
000040	End of tape detected
000020	Rewinding
000010	Beginning of tape
000004	Protected (OK)
000002	Overrun
000001	Rewind complete

NOTES ON DISK DRIVES600MB Disk Drives

The 4490 Series of 600MB disk drives operate under several restrictions at Rev. 18.4. These restrictions include the following:

- A drive of a different type must be available for paging, as there is no paging on 4490 disks.
- Disk-to-disk physical copy is limited to like devices.
- PHYSAV requires a software patch to operate properly. Consult your Prime field engineer to obtain the patch.
- There is a slight capacity loss because of unaddressed cylinders and the nature of badspot handling.
- The maximum partition size is 30 heads (460 MB); the other 10 heads (154 MB) must form one partition.

For additional details, consult your Prime field engineer.

160MB Disk Drives

On the 4480 Series of 160MB disk drives, four-head partitions are available for paging at Rev. 18.4. Thus, partitions of up to 64MB can be used for paging. At Rev. 18.4, it may not always be possible to use larger partitions for paging, because PRIMOS handles a maximum of 16 badspots per paging partition. Four-head paging partitions should be adequate for most installations. For additional details, consult your Prime field engineer.

CHAPTER 3

LANGUAGES

BASIC/VM

PROBLEMS FIXED

MARGIN Statement: Large margins were causing unpredictable program errors, including problems with assigning values to 0th array elements. The MARGIN statement has been modified to accept a range of 1 to 1000 instead of 1 to 32767 (POLERS #14445, 22252, 31240, 32468, 33524).

OLD Command: If the OLD command was used with a nonexistent filename as argument, the foreground file was deleted. This problem has been corrected (POLER #23481).

Assignment Statements: The compiler was accepting LET statements with constants on both sides of the equal sign (for example, LET 34 = 56). Such statements are no longer accepted (POLERS #24782, 33288).

Statements such as LET A = 5 and A = 5 are now interpreted as equivalent, as they should be (POLER #30491).

Writing to BINDA files: Records written to BINDA files sometimes contained portions of previously-written records because buffers were not being cleared. This problem has been corrected (POLER #25085).

Subscript Problems: Floating point array subscripts are now evaluated correctly (POLER #25478).

BASICV now accepts assignment statements with function array subscripts on the left hand side of the equal sign (POLER #22250).

Extraneous DIM statements: The compiler now flags multiple DIM statements for the same array (POLER #27202).

Abnormal Program Halts: BASICV was causing abnormal and inconsistent program halts. This problem has been corrected (POLER #28791).

SPA Modifier: Using SPA with negative arguments caused program errors. Negative arguments are now treated as zero (POLER #28793).

ELSE GOTO/ELSE GOSUB Statements: The compiler was interpreting ON GOSUB...ELSE GOTO as ON GOSUB...ELSE GOSUB. Both ON GOSUB...ELSE GOTO and ON GOSUB...ELSE GOSUB are now supported. The latter pair has the form:

```
ON [num-expr] GOSUB [lin-num-1,...lin-num-n]
    ...ELSE GOSUB [lin-num] (POLER #28895).
```

LIST Following RESEQUENCE: The LIST command following a RESEQUENCE command now works correctly (POLER #28914).

Rounding Consistency: The INT function now rounds decimal numbers correctly (POLER #29528).

LIN Modifier: LIN(0) now produces a carriage return without a line feed, as it should (POLER #27884).

LIN modifiers in WRITE statements were causing all output to be sent to the terminal (rather than to a specified output file). The compiler no longer allows LIN modifiers in WRITE statements (POLER #14440).

RESEQUENCE Command: The RESEQUENCE command was changing some characters in program lines as it resequenced. This problem has been corrected (POLERS #31144, 36867, 37252).

The RESEQUENCE command was deleting lines, producing lines with duplicate line numbers, and using incorrect increments. These problems have been corrected (POLER #28792).

MAT Statement Error: The illegal statement MAT A = MAT B * MAT C now elicits the error message "Illegal syntax in a MAT statement" rather than the message "Internal compiler error in cmlmt" (POLER #32467).

ATTACH Command: The ATTACH command of BASICV now accepts pathnames with passwords (POLERS #24783, 33441).

Repeated Runs: The MARGIN counter, cursor position, and other printing parameters were not being reset between consecutive RUN commands. This problem has been corrected (POLERS #33525, 34918, 82480).

CVT\$\$ Function: Mask 1 (parity bit off) of the CVT\$\$ function now works correctly (POLER # 33526).

AND/OR/NOT Clauses: In certain IF statements with complex AND, OR, and NOT clauses, BASICV was misinterpreting parentheses. This problem has been corrected (POLERS #33529, 34394, 36052).

Exponential Function: BASICV now correctly handles large negative exponents (POLER #34237).

The FNEND Statement: Extraneous FNEND statements in a program were causing a halt with error ACCESS_VIOLATION\$. The compiler now flags extraneous FNENDS (POLERS #34821, 41405).

Breakpoints: Setting a breakpoint to a nonexistent line caused unpredictable results. This problem has been corrected (POLER #35153).

"IF NOT NOT..." Statements: Statements such as "IF NOT NOT A THEN GOTO 100" were causing program halts. The compiler no longer allows such statements (POLER #35335).

MIDAS: Sequential reads now work correctly after deletion of the current record (POLER #35889).

TAB Function: The TAB function was giving erratic results. This problem has been corrected (POLER #41408).

File Writing Errors: Under certain circumstances, writing a file from a BASICV program caused unpredictable results (for example, failure of the BREAK key). This problem has been corrected (POLER #41409).

Spurious blank records were being written to ASCDA files. This problem has been corrected (POLER #36059).

Problems with Large Programs: Very large programs sometimes do not compile because the compiler's constant table is exhausted. Patches are now available upon request (POLERS #42933, 81488).

Spurious error messages were being generated during the editing of large BASICV programs. This problem has been corrected (POLER #35435).

Scratch Files: ASCDA scratch files are now automatically deleted when closed, as they should be (POLER #81646). BASICV was sometimes deleting program file(s) instead of the scratch file. This problem has been corrected (POLERS #44146, 80864).

Erroneous Error Messages: The FILE command was producing the error message "File in use. Tree search error" when there was no error. This problem has been corrected (POLERS #82214, 27836).

BASICV was producing the error message "File open on delete" when there was no error. This problem has been corrected (POLERS #33438, 33503, 33558).

RUN Command Problem: RUN commands with statement numbers did not work correctly in programs with more than one DIM statement. This problem has been corrected (POLER #82706).

DOCUMENTATION CORRECTIONS AND ADDITIONS

The following corrections apply to the Rev. 18.3 Software Release Document (MRU4304-006):

On page 3-3, under String Function Formats, the entry for INDEX should read: INDEX (X\$, Y\$ [,Z]).

On page 3-4, the pathname in Step 4 should be BASICVSRC>BASICV.BUILD.CPL.

On the last line of page 3-4, the phrase "Step 1 is not necessary" should be replaced with "Steps 1 and 3 are not necessary."

On page 3-5, Steps 4 through 6 should read as follows:

4. Run BASICV>BASICV.BUILD.CPL with the RESUME command.
5. Run BASICV>BASICV.INSTALL.COMI with the COMINPUT command.
6. Run BASICV>BASICV.SHARE>COMI with the COMINPUT command. This step can only be performed from the supervisor terminal.

It is recommended that Steps 1-5 be performed at a terminal other than the supervisor terminal, because these steps are time-consuming. However, Step 6 can be performed only at the supervisor terminal.

The following corrections apply to the BASIC/VM Programmer's Guide (FDR3058). These and other corrections will be incorporated into the next update of that book (BASIC/VM Programmers's Guide Update, COR3058-002).

Reading Data Lists: At the bottom of page 5-2, the programming example should read as follows:

```

10 DATA 5,10,15,10
20 DATA 2,7.2
30 READ X,A1,A2,A3,B,C
40 PRINT 'PARTIAL SUM =' ; A1+A2
45 PRINT 'PARTIAL SUM=;A3+B+C
50 PRINT 'AVERAGE=' ; (A1+A2+A3+B+C)/X
60 END
>RUNNH
PARTIAL SUM =25
PARTIAL SUM=19.2
AVERAGE=8.84

```

Formatting String Fields: On pages 5-10 and 5-11, the sections on angle-brackets should read as follows:

Left angle-bracket (<): The primary function of the left angle-bracket is to left-justify text in a character string. In a string, a left bracket by itself causes the leftmost character to be printed:

```

>PRINT USING '<', 'UGANDA'
U

```

Other format characters in the format string dictate how many other characters will be printed, as well as the field positions they will occupy:

```
>PRINT USING '<##', 'UGANDA'
UGA
```

Right angle-bracket (>): The primary function of the right angle-bracket is to right-justify text in a character string. In a string, the right angle-bracket by itself causes the rightmost character to be printed in the first character position in the print zone:

```
>PRINT USING '>', 'UGANDA'
A
```

Depending on the other format string characters, the item may begin printing in the first character position of the print zone, or forced to other print positions as shown below:

```
>PRINT USING '>#####', 'YES'
YES
```

Placing more than one left or right angle-bracket in a format string is not useful. All of the format characters following the second bracket are ignored:

```
>PRINT USING '<##<###', 'UGANDA'
UGA
```

A bracket that falls anywhere but the first character position is not useful. Everything is processed up to the bracket, and everything to the right of the bracket is ignored:

```
>PRINT USING '###>###', 'UGANDA'
UGA
```

FOR Statement: At the bottom of page 6-11, "6" should be changed to "I" in line 20 of the example and in the short paragraph following the example. At the top of page 6-12, line 100 of the example should read, "100 FOR I=1 STEP 1 UNTIL J=10E4". In the next paragraph, "I 10" should be changed to "I*10".

ALTER Command: On page 7-2, in the section on the ALTER command, the reference to Section 14 should be to Section 13.

File Types: On line 19 from the top of page 8-1, "seven file types" should read "seven other file types".

READ AFTER WRITE Error: On page 8-8, line 30 of the second programming example should be deleted.

MIDAS Access Statements: In the table of terms and definitions in the middle of page 8-22, KEY0 and PRIMKEY should both be defined by, "Represents the primary key." The definition of SAMEKEY is, "Positions to or returns datum only if next key matches current one."

MATREAD* Statement: On page 9-15, line 100 of the example should read, "100 PRINT 'SECOND MATREAD*'",. In the output for the example, the third line should be changed from "SECOND MATREAD" to "SECOND MATREAD*".

INT Statement: At the bottom of page 10-1, "INT(.99989) = 1" should be changed to "INT(.99989) = 0".

Evaluating Numeric Expressions: At the bottom of page 11-2, the one-line expression following the phrase "is interpreted as" should have an additional ")" appended. At the top of page 11-3, the following changes should be made:

Line 3: "(E^F)G" should read "(E^F)^G"
 Line 4: "(E(FG))" should read "((E^F)^G)"
 Line 5: "AB" should read "A+B"
 Line 7: "ABC" should read "A^B^C"

SINH: In Table 10-1 (page 10-3), the definition of SINH(X) should read: $(\text{EXP}(X) - \text{EXP}(-X))/2$.

COMINP: The COMINP statement accepts a filename, rather than a pathname, as an argument. At the bottom of page 6-15, "pathname" should be changed to "filename". This change should also be made on pages 13-2 and 14-2, under COMINP.

Typographical Errors: In the table on page 8-1, the braces should be positioned as follows:

$$\text{ON } \left\{ \begin{array}{l} \text{ERROR} \\ \text{END} \end{array} \right\} \text{ GOTO}$$

In the CREATK example on page 8-25, in the line "DATA SIZE = : 64", the colon should be printed in dark brown.

In the first example on page 9-6, the numbers 10, 20, 30, 40, 50, and 60 as input by the user should be printed in dark brown.

On the third line from the top of page 9-9, "dimenof" should be changed to "dimension of".

In the middle of page 10-13, item 2 should read: "Line 110 changes the variable Y to equal X^2 ."

On page 4-4, the symbol "><" in the left-hand column of the table of operators should be removed.

In Table 10-2 (page 10-7), the commas that precede the option brackets in the INDEX, SUB, and VAL formats should appear inside the brackets.

ENTER: The ENTER statement can now take a numeric expression as an argument. In the top line of page 5-5, "time-limit is expressed" should be changed to "numeric expression time-limit is expressed". This change should also be noted on page 14-3, under ENTER.

TAB, SPA, and LIN: The TAB, SPA, and LIN modifiers can now take numeric expressions as arguments. In the paragraphs describing these modifiers on pages 5-7 and 5-8, the words "parenthetical value" should be changed to "numeric expression in parentheses". In the descriptions of TAB and SPA, it should also be noted that negative arguments are treated as 0.

STEP: STEP can now take a numeric expression as an argument. On page 6-3, in the paragraph under the format example, "incr" should be changed to "incr, a numeric expression". This change should also be noted on page 14-3, under FOR.

MATINPUT, MATPRINT: On page 9-6, the following sentence should be inserted after the first example: "Multiple matrix variables are allowed in MATINPUT and MATPRINT statements."

MARGIN: The change in the range of MARGIN (previously 1 to 32767, now 1 to 1000) should be noted on page 5-12 (in the first paragraph under "Changing output line length") and on page 14-7, under MARGIN. (See PROBLEMS FIXED, above.)

COBOL

NEW FEATURES

New Error Messages: Two new error messages have been added to the COBOL compiler:

- The compiler no longer aborts with INTERNAL ERRORS when more than 35 values are used on a level-88 data name. Instead, the error message "NUMBER OF VALUES IN LEVEL 88 EXCEEDS 35" is generated (POLERS #29465, 37613).
- Since no conversion is done when an elementary numeric item is moved to a group level item, the warning message "MOVE DONE WITHOUT CONVERSION" is now issued. (For example, this warning is issued if a PIC 9(5) COMP-3 field is moved into a group level item that is made up of numeric DISPLAY fields.)

Note

The COBOL compiler no longer allows non-numeric figurative constants to be moved to numeric target fields, as in the statement "MOVE SPACES TO PIC-9-FIELD". Such statements are now flagged as errors by the compiler. This change has been made to conform to ANSI standards.

PROBLEMS FIXED

PROCEDURE DIVISION: Compilation no longer aborts if a legal paragraph name does not immediately follow the PROCEDURE DIVISION header (POLERS #14651, 25058, 25773, 29983, 33284, 35961, 36466, 40110, 40869, 41604, 46852, 82062).

Minor errors preceding the PROCEDURE DIVISION header no longer cause the compiler to abort (POLERS #21255, 40070).

Condition Names: Subscripted numeric level-88 condition names were not producing correct results. For example, IF TEST(I)... would not always evaluate properly if the data type was numeric. This problem has been corrected for decimal tests (POLERS #20475, 32159, 41857).

BLANK WHEN ZERO: The BLANK WHEN ZERO clause no longer forces the suppression of leading zeros (POLER #21793).

PIC: Moving a PIC S9(5) field to a group item defined as PIC X or PIC 9(5) no longer causes a character to be lost (POLER #22077).

If a blank was the last character in a signed PICTURE item, the NUMERIC class test would evaluate to TRUE. This problem has been corrected (POLER #30265).

Paragraph Names: The compiler no longer aborts if a paragraph has the same name as a variable in the program (POLERS #27069, 44216).

The compiler no longer aborts if a paragraph name is a reserved COBOL word (POLER #40544).

INTERNAL ERROR Messages: Setting an indexed data name to an index (or vice versa) no longer generates an INTERNAL ERROR 116 or 106 (POLER #28617).

An illegal move to a subscripted identifier no longer causes INTERNAL ERROR 117 (POLERS #29336, 32701).

Paragraphs containing 330 statements no longer generate INTERNAL ERROR 106 (POLER #31848).

DISPLAYing a field with more than 131 characters no longer causes INTERNAL ERROR 116 (POLER #34804).

INTERNAL ERROR 116 is no longer generated by a COMPUTE statement with a subscripted result and an ON SIZE ERROR clause (POLERS #36021, 82846, 40719).

A MOVE CORRESPONDING with more than 11 subscripted correspondences no longer causes an INTERNAL ERROR 117 (POLER #41909).

The compiler no longer generates INTERNAL ERROR 101 if no paragraph name is found after the PROCEDURE DIVISION header and the program uses one file (POLER #48079).

I-O Statements Within IF Statements: Prime's current COBOL compiler does not support conditional I-O clauses on I-O statements that are contained in IF statements. This means that an I-O statement inside an IF statement may not contain an INVALID KEY or AT END clause. Any I-O statement having one of these clauses inside an IF statement is now flagged as an error. For example, the statement:

```
IF A = B READ FILE-NAME AT END GO TO END-PARA.
```

is now flagged as an error. However, the statement:

```
IF A = B READ FILE-NAME.
```

is not flagged. It is suggested that a PERFORM statement be used to accomplish this kind of logic; for example,

```
IF A = B PERFORM MY-READ-PARA.
```

File Description: Placing an "01" where "FD" should appear no longer causes serious compilation problems (POLERS #28815, 36282).

MIDAS: When a MIDAS file was read sequentially by primary key and then a DELETE was issued, subsequent READ statements would read by primary key. This problem has been corrected (POLER #29211).

COMPUTE: The compiler can now handle COMPUTE statements with more than 12 subscripted variables (POLER #29474).

Source Line Counter: The compiler's internal source line counter now works correctly for up to 9999 lines (POLER #29488).

MOVE: A MOVE statement between two ACCEPT...FROM DATE statements no longer generates incorrect object code and incorrect run-time results (POLERS #30266, 35441).

Numeric Arrays: READING a file into a numeric array now works correctly (POLER #32157).

READ Syntax: The compiler now flags a "READ <filename> NEXT RECORD KEY IS <data name>" statement as an error, since the NEXT RECORD and KEY IS clauses conflict in this statement (POLER #32354).

STRING: The STRING statement no longer automatically takes the ON OVERFLOW clause (POLERS #32618, 35212, 40353).

RELATIVE Files: The compiler no longer generates a fatal error when a RELATIVE file is assigned but never opened (POLER #32678).

File Assignments: Repeating file assignments at run-time no longer causes two files to be assigned to the same file-id (POLER #36368).

ADD and SUBTRACT: All ADD statements following a bad SUBTRACT statement were being flagged as errors until a good SUBTRACT statement was found. This problem has been corrected (POLER #36658).

COPY: A COPY statement without a period is now flagged as a syntax error (POLER #82847).

Notes

Users of COBOL should note that the "DEPENDING ON <data name>" clause is not supported in the DATA DIVISION. If used there, this clause may cause unpredictable results at compile-time.

When a RELATIVE file is created with CREATK and defined in a COBOL program, the record length specified in the program must match the record length specified in the CREATK dialog.

DBG (SOURCE LEVEL DEBUGGER)NEW FEATURESVRPG

DBG now supports VRPG, Prime's new V-mode compiler for the RPG II language. A description of DBG's VRPG support follows. For general information on VRPG, refer to the section on VRPG later in this chapter.

Variables: All VRPG variables are either alphanumeric or numeric. The DBG "type" of an alphanumeric variable is alphanumeric (n), where n is the field length of that variable. The DBG "type" of a numeric variable is trailing overpunch (m [, n]), where m is the field length of the variable, and n is the number of decimal positions. (n is omitted if it is zero.)

The print format of an alphanumeric variable is xxxx, where x is any alphanumeric character. The print format of a numeric variable is [-]dddd[.dddd], where d is any digit. A decimal point is printed if the number of decimal positions is greater than zero. Leading zeroes are suppressed left of the decimal point; however, for fractional values, the decimal point is preceded by a single leading zero. DBG always prints the specified number of digits to the right of the decimal point, including trailing zeroes. If the value is negative, the trailing overpunch is printed as a leading separate minus sign.

Indicators: The DBG "type" of an indicator is fixed binary (15) external. The print format of an indicator is d, where d is either 0 or 1.

Arrays and Tables: The DBG "type" of an array or table is "1 dimensional array: (lb:ub)", where lb is the lower bound and ub is the upper bound. The print format of an array or table is a sequence of lines of the form a(i) = b, where a is the array or table name, i is the current subscript, and b is the value of the ith element of a. Each element is printed on a separate line, with i ranging from the lower bound to the upper bound. The print format of each of the elements is that of an ordinary alphanumeric or numeric variable (depending upon the data type of the array or table).

Accessible Variables: The following VRPG variables are accessible inside DBG:

- Indicators: Indicators are accessed via the name IND\$xx, where xx is a VRPG indicator; for example, IND\$L3 represents the L3 indicator.
- Date fields: The date fields are UDATE, UDAY, UMONTH, and UYEAR.
- Page fields: The page fields are PAGE and PAGE1 through PAGE7.
- User fields: The user fields are all of the fields defined in Input or Calculation specifications.
- User arrays: The user arrays are all arrays defined in Extension specifications.
- User tables: The user tables are all tables defined in Extension specifications.

Program Control: VRPG labels (as defined in TAG and ENDSR statements) and subroutines (as defined in BEGSR statements) are available inside DBG. The DBG "type" of a label is "label constant", and the print format is its procedure frame address. The DBG "type" of a subroutine is "entry constant", and the print format is the address of its ECB.

The entire VRPG program is considered to be inside the scope of an external procedure named RPG\$MAIN. The DBG "type" of RPG\$MAIN is "entry constant external".

Reserved Identifiers: The reserved identifiers are as follows:

RPG\$MAIN	Name of external procedure including entire RPG source program.
IND\$xx	Name of an indicator, where xx is any legal VRPG indicator (for example, IND\$L3 for the L3 indicator).
IX\$zzz	Name of RPG table index (explained below), where zzz are the last three letters of the corresponding table name. (For example, the index for the table TABABC is IX\$ABC.)

Setting Variables to New Values: The DBG LET subcommand allows you to assign new values to VRPG variables. All of the variables listed in the above section on accessible variables may be updated with the LET command, although it is not recommended that you update the date fields or page fields.

Updating the values of tables is slightly more difficult than updating the values of fields or array elements. For each table, VRPG keeps a table index field that contains the currently selected element from the

last LOKUP operation. The table index field is a variable named IX\$yyy (where the table is named TAByyy). The table index is of DBG "type" fixed binary(15). You may update the currently selected element in a table by assigning a new value to the table's index field.

Similarly, indicators are of DBG "type" fixed binary(15), and always contain either 0 or 1. While it is possible to assign other values to indicators, doing so will cause erroneous results.

Other New Features

Pascal: Value tracing is now supported in Pascal evaluation mode.

New SOURCE Subcommand: A new DBG SOURCE subcommand has been added to reset the default source file name for a given program block. The syntax of this command is:

```
SOURCE RENAME filename [-BLOCK program-block-name]
```

(abbreviation: SRC RN)

This command replaces the default (compiler-supplied) source file name of the indicated program block with filename. If program-block-name is not specified, the program block corresponding to the evaluation environment is assumed. If the indicated program block is the same as the current block, the current source file is changed to file-name (POLER #26947).

Value Tracing: Value tracing performance has been improved by a factor of 2 to 3, depending on program size (POLER #81498).

STATUS Command: The STATUS command now displays the type of value tracing selected (entry_exit or full).

Command Length: The maximum length for a DBG command line has been increased from 256 characters to 512 characters. The maximum length for a single command (separated by semicolons) within a command line has been increased from 120 to 128 characters. This change affects the DOCUMENTATION ADDITION on page 3-11 of the Rev. 18.3 Software Release Document (MRU4304-006).

PROBLEMS FIXED

Access Violation: Initialization of more than one external module in a single file no longer causes an access violation. Use of the FULL_INIT option is no longer required in such cases (POLER #25450).

A problem that was causing occasional access violations during evaluation of arrays has been corrected.

External Variables: Variables declared outside the scope of a procedure are now visible to DBG (POLERS #25450, 35852).

PL/I Replaced Symbols: Replaced symbols in PL/I, Subset G now apply to the entire file rather than only the first procedure (POLERs #25450, 35852).

Negative Constants: It is now possible to evaluate or assign large negative constants (POLER #28711).

Subscripted Labels: A warning message is now displayed on an attempt to evaluate subscripted labels as an array (POLER #32661).

File Units: File units were not being handled correctly by DBG on systems configured for fewer than 62 file units. This problem has been corrected (POLER #34352).

RESUBMIT Command: The RESUBMIT command now works correctly with separators inside action lists (POLER #34956).

Following a command error, the previous command line is no longer truncated, and is still available for RESUBMIT.

Pascal: The ordinal value of a Pascal character with bit offset is now computed correctly (POLER #35849).

FORTRAN: Substring references are now permitted on the left-hand sides of FORTRAN assignment statements (POLER #45443).

FORTRAN 32-bit octal constants are now evaluated correctly.

CALL Command: The CALL command now correctly passes alternate return labels to FORTRAN modules.

Strings: The string built-in functions (including INDEX) now work correctly.

A problem that was causing a stringrange error on some literals has been corrected.

Table Overflow: A warning message is now printed if the breakpoint table overflows.

Alternate Entry Blocks: Full initialization of alternate entry blocks is now done correctly.

Special Characters: Special characters are now handled correctly in action lists.

Error Message: An appropriate error message is now supplied for a command line that is too long.

HELP: The HELP command now works with the argument "BREAKPOINT-IDENTIFIER".

PROBLEMS OUTSTANDING

Exponentiation: The exponentiation operator produces incorrect results (POLER #35832).

FORTTRAN: Large real numbers in FORTRAN do not print (POLER #40066).

Note

DBG uses segments '4037 and '4036, and allocates temporary segments not occupied by the user program downward from '4037.

FORTRAN (FIN)

PROBLEMS FIXED

Source Files: A source file pathname can now include blanks so that passwords can be specified (POLERS #33990, 40592, 40991).

Relational Operators: An overflow problem that occurred when relational operators compared integers has been corrected (POLERS #35339, 82303, 82614, 25845).

Error Messages: The compiler now reports an error when an array name is used as a statement function dummy argument (POLER #82976).

The "Parameter is better" warning message was occasionally inaccurate. This problem has been corrected (POLER #12484).

A program size overflow message has been clarified.

ENCODE: The modification of an array by an ENCODE statement is now reflected in the cross-reference listing and in the "Never given a value" warning of debug mode (POLER #12490).

Format Specifiers: The compiler now accepts a variable as the format specifier in an I/O statement (POLER #21112).

Continuation Lines: The compiler now uses two source lines, if necessary, to determine the type of a FIN statement. This enhancement corrects some problems that were caused by continuation lines (POLERS #27520, 82273, 36472, 32652, 29567, 45138, 37287).

Floating Point: A problem with floating point comparison in 64R mode has been corrected (POLER #33631).

DMIN1 Function: The DMIN1 function in 64V mode no longer rejects argument lists with more than four arguments (POLER #34908).

Cross-Reference Listings: The line numbers where equivalenced variables were specified are no longer specified (POLER #21197).

External names with "\$" as the second character are no longer omitted from the cross-reference listings (POLER #37636).

The cross-reference listings for certain intrinsic functions have been corrected (POLER #82611).

Rounding: A problem with the rounding of real and double-precision literals has been corrected, improving arithmetic precision.

Array Syntax: Statements with illegal syntax involving arrays compiled without error messages, producing incorrect program results. The errors are now detected (POLERS #81994, 40709).

Insert Files: Using \$INS instead of \$INSERT neither generated a compile-time error nor inserted a file into the program. This problem has been corrected (POLER #30130).

Indirect Errors: Indirect errors were sometimes occurring in the object code and being detected by SEG. This problem has been corrected (POLER #36980).

DOCUMENTATION CORRECTIONS

On page 3-13 of the Rev. 18.3 Software Release Document (MRU4304-006), the entry under NEW FEATURE should read as follows:

Additional Logical Units: Two more logical units were added to the IOCS system. The units are 140 and 141 for printer units 0 and 1, respectively. FORTRAN units may reference these units.

On page 6-11 of the FORTRAN Reference Guide (FDR3057), the following caution should be inserted just before the section called "CONTROL STATEMENTS" (POLER #40089):

Caution

Negative numbers cannot be raised to real-number powers in FIN. An attempt to do so may result in program failure. However, negative numbers can be raised to integer-valued powers.

PROBLEMS OUTSTANDING

FIN vs. F77: A program that performs a great deal of single-precision arithmetic may produce different results, depending upon whether it is compiled by FIN or by F77 (POLER #28703).

XREFS Option: When compiling a particular large program with the XREFS option, FIN produces a "program overflow" error message and eventually halts abnormally (POLER #31241).

XREFL Option: When a program is compiled with the XREFL and UNCOPT options, the XREFL listing incorrectly places some statement labels at the end of a do loop instead of within the loop (POLER #45192).

Arithmetic: A particular arithmetic expression with mixed integer and real operands evaluates incorrectly (POLER #43819).

Subtractions of double precision variables that have exponents greater than 17 are sometimes evaluated inexactly (POLER #43820).

Precision: Under certain circumstances, FIN with the -FRN option gives the same rounded single precision value to both a single and a double precision variable (POLER #37414).

FORTRAN 77 (F77)

PROBLEMS FIXED

Runtime Library Changes: Changes in the runtime library have increased the following three limits:

- The number of arguments that can be passed to and from subroutines (now up to 254 in most cases).
- The number of entries that may be present in a NAMELIST block (now up to 247 in most cases).
- The number of separate target labels permitted in a computed GOTO statement (now up to 254).

There are exceptions to the first two limits indicated above. The exceptions occur particularly when CHARACTER data is involved. For example, if all arguments to a subroutine were of type CHARACTER, only 127 arguments would be allowed. However, if arguments were of mixed data types (CHARACTER and other types), the maximum would be between 127 and 254, and would depend upon the positions occupied by the CHARACTER data in the argument list. Similar restrictions hold for NAMELIST blocks.

Data Type Conversion: The INT, INTS, and CHAR intrinsic functions now function correctly (POLERS 27719, 31868).

Output Format: The unimplemented output formats Z (hexadecimal) and O (octal) now result in the compiler error message, "Unexpected character in format" (POLER #36544).

The non-ANSI FORTRAN omission of a delimiter (such as a comma) after an X format descriptor now results in compiler error 449 (POLER #36500).

Arrays: The source level debugger (DBG) now correctly evaluates the type of an assumed-size array. Previously, an access violation occurred when the assumed-size array type was requested. Note, however, that information on the correct dimensions of such arrays is unavailable to DBG. (POLER #29734).

A new error message (#447) prohibits the use of assumed-size arrays in I/O lists (POLER #35846).

Adjustable-dimensioned arrays are now passed correctly to subroutines (POLER #40064).

Register Values: All levels of optimization now preserve register values. For example, the contents of the X register are preserved across all calls to LOG, EXP, SIN, etc. (POLER #32927).

PROBLEMS OUTSTANDING

Exponents: The compiler cannot yet handle double-precision constants where the exponent has more than 2 digits, as in 1.0D200 (POLERS #29335, 32927).

Floating Point Precision: Floating point constants evaluated within F77 should be more precise (POLER #36539).

DOCUMENTATION CORRECTION

On page 3-15 of the Rev. 18.3 Software Release Document (MRU4304-006), the entry Additional Logical Units should read as follows:

Additional Logical Units: Two more logical units were added to the IOCS system. The units are 140 and 141 for printer units 0 and 1, respectively. FORTRAN 77 units may reference these units.

Note

The build file for F77 is now F77SRC>F77.BUILD.CPL.

PASCAL

NEW FEATURE

Real Numbers: The minimum field width for real numbers printed out in E format (scientific notation) has been increased to eight characters, in accordance with the most recent ANSI Pascal standard.

PROBLEMS FIXED

DBG: The source level debugger (DBG) now recognizes variables defined in external procedures as external references (POLER #35852).

DBG also now recognizes the standard identifiers INPUT and OUTPUT (POLER #35847).

Compiler Listing: In the listing file that is generated at compile time, the attribute for P\$AINP is now INPUT EXTERNAL, and the attribute for P\$AOUT is now OUTPUT EXTERNAL. (Previously, FILE EXTERNAL was printed for both.)

Disk Organization: The files A\$KEYS.PASCAL, ERRD.PASCAL, and KEYS.PASCAL now reside in the SYSCOM UFD (POLER #35855).

WITH Statements: A WITH statement did not work properly for a record that spanned a segment. This problem has been corrected (POLER #29463).

Optimization: Compiling a program with the DEBUG option now automatically turns off optimization (POLER #35851).

A problem that was causing the compiler to abort during optimization of a valid program has been corrected (POLER #35858).

BAD UNIT NUMBER Error: A BAD UNIT NUMBER error was generated at compile time if segment 4001 was not initialized to zero before compilation. This problem has been corrected (POLER #43164).

Real Numbers: When printing, PASCAL now rounds real numbers instead of truncating them (POLER #27635).

Real numbers with more than 15 digits are now written correctly (POLER #41961).

DOCUMENTATION CORRECTIONS

The following corrections apply to the Pascal Reference Guide (IDR4303):

The sentence "This includes the program name" should be added to the Note at the bottom of page 4-5 (POLER #33344).

On page 5-1, in the paragraph beginning "The keyword PROGRAM must be the first...", the second sentence should read as follows (POLER #33344):

"It is followed by an identifier, which is the name of the program (maximum of eight characters), and an optional file-identifier-list, which is a list of files (separated by commas) used by the program."

PL/I, SUBSET G

NEW FEATURES

Number of Records in a MIDAS File: MIDAS files may now contain up to 32767 records, rather than only 16383 records.

Note

The above change involved a modification of the file control block. For this reason, programs written in pre-Rev. 18.4 PL/I must be recompiled before they are run at Rev. 18.4 or later Revs.

New Command Line Options: A new pair of command line options is now available for PL/I, Subset G. The `-OVERFLOW` option allows for detection of zero-divide and fixed binary overflow conditions. The `-NO_OVERFLOW` option (which is the default) suppresses checking for these conditions. It should be noted that specifying the `-OVERFLOW` option significantly increases code generation time for programs that use arrays.

When `-OVERFLOW` is specified, the built-in function `DIVIDE` now returns an error message if a program attempts to divide a binary value by zero (POLER #24308).

When `-OVERFLOW` is specified, fixed binary values greater than 32767 now create an overflow and generate an appropriate error message (POLER #30110).

New Option for DECLARE ENTRY: The `DECLARE ENTRY` statement has a new option, `OPTIONS(SHORTCALL)`, that allows a procedure to be called using the `PMA` instruction `JSXB` instead of the `PCL` instruction. This option is described below under `DOCUMENTATION CORRECTIONS AND ADDITIONS`.

DOCUMENTATION CORRECTIONS AND ADDITIONS

The following changes apply to the PL/I Subset G Reference Guide (IDR4031):

A-Format Input Conversion: The first paragraph of page 8-12 should read as follows:

A-Format Input Conversion: For input conversion, when `w` is specified, the result is a character string containing the next `w` characters from the input-stream file. If `w` is not specified, characters are read up to the end-of-line or until the character string is filled, regardless of whether the string is character-varying or not.

PL/I Subset G Program Structure: On page F-2, the third item under Program Structure should read as follows:

The following options of the PROCEDURE and ENTRY statements are not supported:

ORDER*, IRREDUCIBLE*, REDUCIBLE*, REORDER*

Iterative-do Statement: The Note at the bottom of page 9-10 should be changed to read as follows:

If both the TO option and the WHILE option are omitted, the do-group is executed once.

Character Data Alignment: The last three lines of page 3-7 should be replaced with the following line:

CHARACTER(n) [VARYING] [ALIGNED]

On page 3-8, the paragraph beginning "Nonvarying character string variables..." should be replaced with the following two paragraphs:

The ALIGNED attribute aligns nonvarying character data along word boundaries. If ALIGNED is not specified, nonvarying character data is byte-aligned. The ALIGNED attribute has no effect on varying character data, which is always byte-aligned.

Nonvarying character-string variables always occupy exactly n bytes of storage. As elements of arrays or members of a structure, they begin on the next available byte if they are byte-aligned. In this case, an array of nonvarying characters can be stored and accessed as if it were a single string. See Section 4 for a discussion of storage sharing.

On page 5-7, the paragraph on ALIGNED should read as follows:

ALIGNED is an optional part of the bit-string and character string specifications. Its presence allows an implementation to align data on a convenient storage boundary (word boundary, for nonvarying character data). Specifying ALIGNED may cause a bit string to use more bits of storage than are specified by its declared length. See Section 3 for discussions of bit-string and character data.

The following two entries should be added to the list on page 11-2:

Character(n) Aligned	word	n bytes
Character(n) Varying Aligned	word	(n+3)/2 words

Also on page 11-2, the last sentence before the section on TTY input/output should read as follows: "The ALIGNED attribute applied to character(n) varying data has no effect on the alignment of the data."

On page C-7, the entry labelled Alignment should read as follows: "CHARACTER VARYING data is always byte-aligned. Nonvarying CHARACTER data is byte-aligned unless ALIGNED is specified, in which case it is aligned along word boundaries."

ROUND Function: On page 10-12, the second sentence in the section on ROUND(X,K) should read as follows: "The result is always fixed decimal, and is the value of X rounded such that the Kth position of X is expressed to its nearest integer."

DECLARE ENTRY Statement: The following text should be added to page 5-11, just before the section entitled EXTERNAL:

The OPTIONS(SHORTCALL) form of the DECLARE ENTRY statement is useful for calling procedures with the PMA instruction JSXB instead of with the more common PCL instruction. Using JSXB to call a procedure is faster than using PCL; however, if OPTIONS(SHORTCALL) is to be used, the called procedure must be written to expect a JSXB-type call. At Rev. 18, no system subroutines can be declared in this way.

The format of the OPTIONS(SHORTCALL) declaration is as follows:

```
DECLARE procedure-name ENTRY OPTIONS(SHORTCALL) [stack-size];
```

where stack-size specifies the extra space needed for the calling procedure's stack. The default size is 12.

A subroutine that is called using OPTIONS(SHORTCALL) cannot require arguments. The call does not generate a new stack for storage, as does a call that uses PCL. The calling procedure's stack space is used. Thus, it may be necessary to specify stack size in the declaration in order to enlarge the calling stack. If stack size is not large enough, the return from the subroutine will cause unpredictable error messages.

A user-written procedure to be called using OPTIONS(SHORTCALL) should not include the ARGV declaration. Arguments, if any, must be passed through a pointer in the L register. See the Assembly Language Programmer's Guide (FDR 3059).

PMA

DOCUMENTATION CORRECTIONS

The following corrections apply to the Assembly Language Programmer's Guide (FDR 3059):

Device Options: In Figure 3-1 (page 3-3), the last item in the DEVICE OPTIONS list should read "7=DISK (default)".

Typographical Errors: On page 11-4, the first line in the section on CGT (Computed GOTO) should begin: "If $1 \leq A < n$, then..."

On page 11-9, in Table 11-1, the first item in the column headed "Positive" should be "0,space ('240),+,{ " rather than "0,+{ ". Also, the first item in the column headed "Negative" should be "-,{ " rather than "-,{ ".

On page 11-45, the first line in the section on JDX (Jump and Decrement X) should begin "X=X-1;if X not= 0, then..."

On page 11-46, the first line in the section on JIX (Jump and Increment X) should begin "X=X+1;if X not= 0, then..."

IFM, IFP, IFZ, IFN: On page 16-9, in the section on:

[label] {IFM | IFP | IFZ | IFN} absolute-expression

the following phrase should be inserted before the two paragraphs of explanation: "where IFM = if minus, IFP = if plus, IFZ = if zero, and IFN = if not zero."

INT, INTA, and INTL: On pages 11-21 and 11-22, the descriptions of INT, INTA, and INTL should read as follows:

- INT: Convert the double precision number in the double precision floating point register to a 31-bit integer. Store the result in register A and bits 2-16 of register B. Force bit 1 of B (bit 17 of the result) to 0. This instruction ignores the fractional portion of the floating point number. If the value in the floating point register is less than $-(2^{*30})$ or greater than $(2^{*30})-1$, a floating point exception occurs and the C-bit is set. If bit 7 of the keys contains a 0, a floating point exception fault occurs. MODES=V, FORMAT=GEN, OPCODE=140554 C=exception, L=unspecified, CC=unspecified.
- INTA: Convert the double precision number in the double precision floating point register to a 16-bit integer and store the result in register A. The fractional portion of the

floating point number is ignored. If the value in the floating point register is less than $-(2^{**15})$ or greater than $(2^{**15})-1$, a floating point exception occurs and the C-bit is set. If bit 7 of the keys contains a 0, a floating point exception fault occurs. MODES=V, FORMAT=GEN, OPCODE=140531, C=exception, L=unspecified, CC=unspecified.

- INIL: Convert the double precision number in the double precision floating point register to a 32-bit integer, and store the result in register L. The fractional portion of the floating point number is ignored. If the value in the floating point register is less than $-(2^{**31})$ or greater than $(2^{**31})-1$, a floating point exception occurs and the C-bit is set. If bit 7 of the keys contains a 0, a floating point exception fault occurs. MODES=V, FORMAT=GEN, OPCODE=140533, C=exception, L=unspecified, CC=unspecified.

RPG

PROBLEMS FIXED

Spacing 0: Spacing 0 before and after on output now work correctly (POLER #33310).

Fetch Overflow: Fetch overflow is now properly coordinated with page eject (POLER #33311).

DIVIDE Operation: DIVIDE now works correctly with a subscripted array as result field (POLER #33481).

Matching Fields: Sequence checking now works with packed decimal matching fields (POLER #35157).

VFINLIB

NEW FEATURES

I/O Improvement: Binary F77 I/O now works 20% faster.

NAMelist: NAMelist Read/Write blocks can now handle up to 247 variables. Refer to the first "Problem Fixed" in the section on FORTRAN 77 for more information.

NAMelist support for FORTRAN 77 is now unshared.

IOCS: Two more logical units have been added to the IOCS system for use by PL/I. The units are 140 and 141, for printer units 0 and 1, respectively. FORTRAN and FORTRAN 77 programs may reference these units as well.

PROBLEMS FIXED

F\$INQF: F\$INQF no longer closes any command input file that the user has open at the time of inquiry (POLER #11985).

F\$MIN: F\$MIN now gives correct results if the arguments are within legal bounds (POLERS #22050, 40357).

F\$IO77: F\$IO77 does correct outputs, with leading zeros only when necessary (POLER #27250).

F\$IO77 now accepts B-format statements with surrounding blanks (POLER #29621).

F\$IO77 now handles non-word-aligned character output from internal formats correctly (POLERS #31488, 32927).

SQRT: SQRT now allows more accurate comparisons between returned results (POLER #28524).

DEXP: DEXP now returns 1.0 if its argument is 0.0 (POLER #29556).

F\$IOFIN: F\$IOFIN now operates properly on multiple, internal, sequential commas (POLER #30249).

F\$IOFIN now handles encodes of non-word-aligned characters correctly (POLER #32166).

NAMEQ\$: NAMEQ\$ now checks for lower case a (POLER #31198).

CABS: CABS no longer overflows if the argument is within legal bounds (POLER #32724).

F\$CLOS: F\$CLOS can now delete a scratch file from a passworded UFD, provided that the program is executed by a user with owner status (POLER #40213).

Error Codes: System error codes are now appropriate for the types of errors generated, assuming that VFINLIB is built using a Rev. 18.4 PRIMOS.

DOCUMENTATION CORRECTION

On page 3-35 of the Rev. 18.3 Software Release Document (MRU4304-006), the entry IOCS System should read as follows:

IOCS System: Two more logical units were added to the IOCS system for use by PLIG. The units are 140 and 141 for printer units 0 and 1, respectively. FORTRAN 77 units may reference these units.

VRPG

INTRODUCTION

VRPG is Prime's new V-mode compiler for the language RPG II. It offers a significant increase in execution speed over RPG, Prime's R-mode compiler for RPG II. VRPG also provides superior error messages and the capability to interface with both DBG (Prime's interactive source level debugger) and EMACS (Prime's screen editor).

For general information on VRPG, refer to the new RPG II V-mode Compiler Reference Guide (IDR5040). For information on DBG's support of VRPG, refer to the discussion of DBG earlier in this chapter. The VRPG/EMACS interface is described briefly below and more fully in the EMACS Reference Guide (IDR5026).

Command Line Formats: VRPG supports the following command line formats:

1. VRPG pathname [-options]
2. VRPG [-options] -INPUT pathname [-options]
3. VRPG [-options] -SOURCE pathname [-options]

where pathname is the pathname of the VRPG source program. The pathname must appear either immediately after the VRPG command (1), or as an argument to the -INPUT option (2) or the -SOURCE option (3).

VRPG supports the following options (asterisks indicate defaults):

-SOURCE	precedes pathname of source file
-INPUT	precedes pathname of source file
-LISTING	precedes pathname of listing file
* -BINARY	precedes pathname of binary file
* -ERRITY	prints error messages on the user terminal
-NOERRITY	suppresses error messages on the user terminal
-ERRLIST	prints error messages in the listing file
* -NOERRLIST	suppresses error messages in the listing file
-XREF	prints cross reference in listing file
* -NOXREF	suppresses cross reference in listing file
-BANNER	prints column index banner in listing file
* -NOBANNER	suppresses column index banner in listing file
-OBDATA	prints generated object data in listing file
* -NOOBDATA	suppresses generated object data in listing file
-SEQCHK	performs sequence checking during compile time
* -NOSEQCHK	suppresses sequence checking during compile time
* -STATUS	prints status message during compile time
-NOSTATUS	suppresses status message during compile time
-EXPLIST	synonym for -OBDATA

- * -NOEXPLIST synonym for -NOOBDATA
- DEBUG generates information for DBG
- * -NODEBUG suppresses DBG information
- STATISTICS prints compile-time statistics
- SILENT suppresses level 1 (warning) error messages

Note

The default options indicated above reflect changes that took effect since the release of the RPG II V-mode Compiler Reference Guide.

EMACS Interface: A special VRPG interface with EMACS (Prime's screen editor) simplifies the process of entering VRPG source code. EMACS provides a set of templates (one for each specification form) that label the column numbers on the screen. In addition, you can compile your source code from within EMACS, and then view errors in one window while the lines that produced the errors are displayed in another window. For more information on the VRPG/EMACS interface, refer to the EMACS Reference Guide (IDR5026).

Runtime Support: The runtime support for VRPG is in the system library VRPGLB. This library should be included when linking and loading programs compiled by the VRPG compiler.

Shared Segments: VRPG uses shared segments '2144, '2145, and '2146. If you are using these segments for another purpose, you will have to make changes so that they can be used by VRPG.

PROBLEMS OUTSTANDING

Compile time tables and arrays must have records of 80 characters or less. If a table or an array record is greater than 80 characters, unexpected results may occur.

The RPG delimiter for end of input is /*. This causes some confusion within a CPL program, which treats /* as a comment delimiter. A conflict arises if an RPG program is executed within a &DATA - &END construct that also contains input data for a console file. If /* is one of the data entries, CPL treats it as a comment and does not pass that line of data to the RPG program.

CHAPTER 4

PRIMOS AND UTILITIES

PRIMOS

NEW FEATURES

The following new features are available with Rev. 18.4 PRIMOS:

- A new subroutine (SEM\$OU) for opening named semaphores
- A new ASSIGN command line option (-DENSITY) for specifying tape density
- A new choice of ways to set tape density on Model 3 1600bpi/6250bpi-capable magnetic tape drives
- New instructions for the T\$MT and T\$CMPC subroutines
- Additions to the HELP data files

Named Semaphores: A new subroutine for opening named semaphores is available at Rev. 18.4. The SEM\$OU subroutine opens a set of named semaphores and associates them with the file that is open on a particular file unit. SEM\$OU is similar to the SEM\$OP subroutine, which opens a set of named semaphores and associates them with a specified filename. However, in addition to using file units rather than filenames, SEM\$OU also initializes semaphore counters to a specified non-positive value. (SEM\$OP always initializes semaphore counters to zero.) For information on SEM\$OP and named semaphores in general, refer to the Prime Technical Update for Rev. 18 subroutines (PTU2600-078).

Before a named semaphore may be used, it must be opened by either SEM\$OP or SEM\$OU. Both subroutines assign the calling process a set of semaphores from a global pool of semaphores, and return the numbers (IDs) of the semaphores assigned.

The calling sequence for SEM\$OU is:

```
CALL SEM$OU (funit, snbr, ids, init_val, code)
```

funit (fixed bin) is a file unit number on which a file is already open. The semaphores that are opened in this call will be associated with that file. (The connection between the semaphores and the file is explained below.) The calling process must have at least read access rights to the file.

snbr (fixed bin) is a number that specifies how many semaphores are to be opened by this call.

ids ((snbr)fixed bin) is an array of semaphore numbers; one number is returned for each semaphore that was opened successfully. (Returned by SEM\$OU.)

init_val (fixed bin) is the value to which the counter of each semaphore being opened for the first time will be initialized. This value must fall between -32767 (decimal) and 0, inclusive; it cannot be positive.

code (fixed bin) is a success/failure code returned by SEM\$OU. The codes and their meanings are as follows:

0	Successful call.
E\$BPAR	An invalid value was supplied for SNBR.
E\$UNOP	There is no longer a file open on the specified file unit.
E\$IREM	The specified file resides on a remote disk.
E\$BUNT	The specified file unit was invalid.
E\$FUIU	Either the user has all available file units opened, or there are no available named semaphores.

It is also possible that code will be set to any of the error codes that can be returned by the SRCH\$\$ routine.

If access is granted to the semaphores, the call will return an array of semaphore numbers in the ids parameter. One number will be returned for each semaphore requested in snbr, assuming enough semaphores exist in the system pool. A semaphore number of zero will be returned if a semaphore could not be assigned. In addition, code will be non-zero if one or more semaphore numbers could not be assigned. The values returned in ids should be examined to determine which semaphores were opened (non-zero value returned) and which were not (zero value returned).

The semaphore numbers returned should be used in all other semaphore calls as the semaphore number parameter.

A process that calls SEM\$OP and a process that calls SEM\$OU will receive the same semaphore numbers if they specify the same file (whether by name or unit). Thus, multiple processes of a subsystem can reference common semaphores. However, only the first process to open a semaphore can initialize the counter; subsequent calls do not alter the counter's value.

If a call to SEM\$OU or SEM\$OP specifies the same file as a previous call, and a larger number of semaphores is requested, then new semaphores are acquired from the system pool to make up the difference between the number currently open under the file and the number requested in the call. However, another process cannot use these newly assigned semaphores without explicitly opening them via a call to one of the two routines.

A named semaphore to be opened may only be associated with a file that resides on the local computer system. Attempts to associate a remote file with a named semaphore during the semaphore open operation will result in failure; no semaphore numbers will be assigned and code will be set to E\$IREM.

ASSIGN: The tape density options of the PRIMOS ASSIGN command for magnetic tapes have changed at Rev 18.4. The options -800, -800BPI, -1600, -1600BPI, -6250, and -6250BPI are no longer recognized. Attempts to use them will result in an error message. These options have been replaced by the new -DENSITY option, which requires a tape density in bpi (bits per inch) as an argument. The format of the -DENSITY option is as follows:

$$\text{-DENSITY } \left\{ \begin{array}{l} 800 \\ 1600 \\ 3200 \\ 6250 \end{array} \right\}$$

Currently only 800, 1600, 3200, and 6250 are legal arguments to the -DENSITY option. Note that the ASSIGN command does not actually check that the drive being assigned is capable of the desired density.

Tape Drive Operation: The tape density on a Model 3 1600bpi/6250bpi-capable magtape drive can now be set either by means of the T\$MT routine or by means of the density select switch on the front panel of the drive. When the density is set via a software call to T\$MT (or via an ASSIGN command, which itself calls T\$MT), the front panel switch is disabled from having any effect on the tape density. The switch can only be reenabled by another call to T\$MT, with the enabling instruction specified. (See the entry under T\$MT, below.)

Under PRIMOS, density setting is handled at magtape ASSIGN time as follows: If no density setting option is given in the magtape ASSIGN command line, the ASSIGN command calls T\$MT to enable the front panel switch. If a density setting is given in the magtape ASSIGN command line, ASSIGN calls T\$MT to set the drive to the desired density, and the front panel density switch is disabled. In neither case is operator intervention necessary; intervention is therefore not requested at the supervisor terminal, as it is with other magtape drive models.

If PRIMOS crashes and a tape dump is to be taken, the act of hitting the "MASTER CLEAR" switch causes the Model 3 drive to revert to its initial state of 1600 bpi (with the front panel switch disabled). The dump is therefore made at this density.

Under PRIMOS II (DOS), Model 3 1600 bpi/6250 bpi-capable drives are initialized with the front panel density select switch enabled.

T\$MT: The instruction listed below is now available for the T\$MT subroutine. (instruction is one of the arguments in a call to T\$MT.) For general information on T\$MT, see pages 19-24 of the Subroutines Reference Guide, PDR3621.

Octal	Hexadecimal	Meaning
100140	8060	Enable front panel density select switch (Version 3 controller only)

T\$CMPC: The status returned by the read status instruction (octal 100000) reflects only the status of the input buffer in PRIMOS. A new instruction has been added: octal 100001 returns the actual hardware status (POLER #27073). See also the entry under T\$CMPC in DOCUMENTAION CORRECTIONS, below.

HELP: VRPG, FIS, FTR, FTOP, FTGEN, and EMACS have been added to the HELP data files.

PROBLEMS FIXED

ADDISK: If the RENAME option of ADDISK fails, the system lock is now released correctly (POLER #41564).

AMLC: The setting and clearing of the LWORD bits now work correctly when AMLC lines are assigned and unassigned (POLER #32702).

The performance of T\$AMLC has been improved in cases where the OUTPUT BUFFER FULL condition occurs frequently.

COMO: Superfluous ATTACH commands were sometimes being issued internally when command files were executing. This problem has been corrected.

Disk Drives: A problem that was causing incorrect header check disk errors has been corrected.

Floppy Disks: PRIMOS can now access track 0 of IBM-format floppy disks.

Installation: C_INSTALLALL and C_INSTALLSTD have been updated to include VRPG.

MAKE: After entering badspots through the MAKE command, the user can now make corrections by answering NO to the PARAMETERS OK? prompt. A response of NO causes MAKE to disregard all badspot entries made so far, and to prompt for a new set of badspots. Previously, new entries were simply appended to old ones, rather than replacing them (POLER #45409).

Message: The MESSAGE command was sometimes losing the last character of a message. This problem has been corrected.

Network: The Node-Node password is now tested correctly (POLER #45343).

A "BAD PASSWORD" error was being produced when a process first attached to a remote disk with an MFD password other than "XXXXXX", then ran an external command. This problem has been corrected (POLER #43290).

Remote File Access: Opening a remote como file and logging out without closing it resulted in an error that could "hang" the user terminal. This problem has been corrected (POLER #45343).

A problem in the initialization of network slaves has been corrected (POLER #45392).

Passwords: Lowercase UFD passwords now function correctly (POLER #27618).

Phantoms: Phantoms will now clean up and log out if fatal errors occur during login (POLER #41622).

PRIMOS now produces an error message if a phantom tries to access a file to that the user has no access rights (POLER #41521).

Phantom jobs doing I/O over the network were not always correctly logged out by a forced logout. This problem has been corrected (POLER #44296).

Network phantoms (slaves) are no longer able to log out other slaves (or any other processes). A slave can only either log itself out or be logged out from the supervisor terminal (POLER #45340).

PRERR: PRERR has been removed from the list of gated PRIMOS routines. The routine PRERR\$ performs a similar function. A problem which caused the system to crash when PRERR was supplied with a single non-positive argument has been corrected (POLER #43324).

Prime 850: The Prime 850 can now be warm started properly.

Printers: XOFF is now processed in time to avoid buffer overruns in NEC spinwriter printers. Output is resumed correctly if XON/XOFF is disabled while output is suspended (POLERS #32445, 42687).

RDLIN\$ and WILIN\$: Negative count arguments in calls to RDLIN\$ and WILIN\$ no longer cause network problems. WILIN\$ now recognizes a disk full condition and no longer writes over a previously-written portion of the file (POLER #45589).

SMLC: Users with user numbers greater than 63 are no longer prevented from assigning synchronous lines (POLER #34096).

T\$CMPC: Reading cards in binary no longer takes twice as long as reading them in ASCII (POLER #27971).

T\$LMPC: The read status instruction was returning a status of ON-LINE regardless of the state of the line printer. This instruction now returns the hardware online status bit from the controller (POLER #33480).

Tape Drive Operation: The current limit for magtape reads and writes is six pages. PRIMOS now detects an error if the user attempts to transfer more than 32K words with a buffer in the low part of a segment.

Magnetic tape ASSIGN commands entered at the supervisor terminal and requesting operator action were hanging the terminal because it waited for a reply from itself. This problem has been corrected. However, the ASSIGN MTX -ALIAS MTn form of the ASSIGN command can no longer be entered at the supervisor terminal, because in this case a reply from the operator is required.

It is now possible to set the density to 6250 bpi from the front panel of the Model 3 controller/magnetic tape drive (POLERS #29261, 31851, 35688). (See the entries under ASSIGN and Tape Drive Operation in the NEW FEATURES section.)

Tape Dumps: An error in the tape dump routine has been corrected, and error recovery has been added. In addition, a tape dump may now be sent to any tape unit (POLER #40048).

UNASSIGN: An error message for the magnetic tape UNASSIGN command that described proper usage of the ASSIGN command has been corrected (POLER #33448).

Versatec Plotter: When one process unassigned the plotter and the next process assigned it, sometimes a line of data was lost. Also, when a blank line was to be printed, sometimes the second character of the previous line would be printed instead. These problems have been corrected (POLER #45555).

Version Stamp: The program VERSION_STAMP.CPL now displays the version of PRIMOS currently running. The program VERSION.CPL extracts the necessary information from COLD.COM1 to create VERSION_STAMP.CPL.

DOCUMENTATION CORRECTION

AMLC: The following correction applies to the System Administrator's Guide (PDR3109). On page 16-6, the following sentence should be added to the description of Bit 2 of the LWORD parameter: "This bit is meaningful only if Bit 1 is set."

BATCH

PROBLEMS FIXED

CPL Jobs: CPL jobs with arguments were not executing successfully if the BATCHQ>CIFILE directory had a non-blank password. This problem has been corrected (POLER #41773).

Error Messages: When a fatal error occurs, the error code number sent to the system console is now correct. (Previously, only the tens digit was sent.)

The "<text> seen where end-of-line expected" message is now correctly provided when necessary.

The error message "Command file required as first argument on submission" has been changed to "Command or CPL file required as first argument on submission."

BATGEN: The BLOCK, UNBLOCK, and DELETE ALL commands in BATGEN now correctly append a comma to the end of each line of queue names if there are more names left.

Note

If a job is submitted from a passworded directory (including your own), the -HOME option must be used. The password must be included in the pathname, and the pathname must be enclosed in quotation marks. For example:

JOB filename -HOME 'dir-name password'.

CPL

NEW FEATURES

Global Variables: The DEFINE_GVAR command now accepts an "-OFF" option, which turns off the current global variable file.

The new command function GVPATH, invoked by [GVPATH], returns the pathname of the current global variable file, if one is defined, or returns "-OFF" if no global variable file is active.

The GET_VAR function now returns \$UNDEFINED\$ if the value of a global variable is requested and no global variable file is active.

WILD: The WILD function now accepts the options -FILE, -DIR, and -SEGDIR. The plural forms of these options still work at Rev. 18.4, but are officially obsolete.

-TTY Option for QUERY and RESPONSE: A new -TTY option has been added to the QUERY and RESPONSE functions to allow these functions to force on terminal input. See DOCUMENTATION CORRECTIONS AND ADDITIONS, below.

PROBLEMS FIXED

CALC: The CALC function now correctly handles + and - operations following parentheses (POLER #37520).

Error Messages: An error message is now supplied if the CPL command fails (POLER #40711).

Echoing: When echoing is enabled, comments and null lines will no longer be echoed. Null lines will be inserted into &DATA temporary files, but comments will not.

&ARGS: The &ARGS directive now explicitly rejects numeric option arguments (for example, those of the form "-123").

DOCUMENTATION CORRECTIONS AND ADDITIONS

The following corrections apply to the CPL User's Guide (IDR4302):

QUERY and RESPONSE: On page 5-1, in the description of the RESPONSE function, the sentence "RESPONSE accepts any character string the user types in" should be replaced by the sentence "RESPONSE accepts any character string as an answer."

On page 5-2, the form of the QUERY function should be changed to:

```
[QUERY {text} {default} {-TTY}]
```

The next full paragraph (beginning "When the QUERY function is encountered...") should read as follows:

When the QUERY function is encountered, CPL prints text at the user's terminal, following it with a question mark. QUERY then reads the command input stream for a response, and returns either TRUE or FALSE, as described below. If the -TTY option is specified, QUERY only accepts input from the terminal. Even if the command input stream is a command file, -TTY causes QUERY to wait for a reply from the terminal. Once the user replies, the normal command stream resumes.

At the bottom of page 5-3, the form of the RESPONSE function should be:

```
[RESPONSE {text} {default} {-TTY}]
```

The first sentence on page 5-4 should read: "This function returns the text string accepted from the command input stream (up to 1024 characters)." In the next paragraph, the sentence "(Like the QUERY function, the RESPONSE function cannot be used in files designed for Batch or phantom execution.)" should be replaced with the following sentence: "The -TTY option for RESPONSE has the same effect as for QUERY."

&ARGS Directive: The third example in the middle of page 13-1 should read as follows:

```
&ARGS CHARM:CHAR; TR_FLAG:-TR TRUTH:DEC;~
  BE_FLAG:-BE BEAUTY:TREE=A_UFD>FILE
```

The same change should be made on page A-1, in the section on &ARGS. The reason for the correction is that control arguments must follow positional arguments in the &ARGS directive (as stated on page 13.5).

Typographical Errors: On page 7-8, line 17 should read:

```
THEN &SET_VAR COMPILER := [RESPONSE 'Please specify compiler']
```

On page 7-12, line 8 from the bottom should read:

```
SET_VAR .OBJ := [WILD @.PLIG -BF 05/30/80]
```

On page 10-5, line 6 should read:

```
&DO I := 1 &TO 5
```

The first item of the list on page 12-10 should read:

-FULL 80-10-21.13:24:48.Tue

On line 10 from the bottom of page 13-5, -no binary should be changed to no_binary.

EDITOR

PROBLEMS FIXED

CHANGE Command: The C/ //G command now works correctly when applied to a line that ends with a space (POLERS #29369, 40212, 82493).

OOPS Command: The command sequence D;N;OOPS now works correctly. That is, the OOPS returns the pointer to the position it was in before the N (POLERS #33936, 34683, 34837).

Extraneous Lines: Moving the pointer around a file with lines containing many blanks no longer causes ED to create extraneous lines (POLERS #37202, 45579).

MODE Command: Typing MODE with an illegal parameter now results in a BAD MODE error message (POLERS #33128, 80479).

WHERE Command: A problem that was sometimes causing the WHERE command to abort the Editor has been corrected (POLER #82205).

EMACS

INTRODUCTION

EMACS is a customizable screen editor which is compatible with ED, RUNOFF, and other Prime products that use text files. EMACS has a built-in help facility that provides information on the available editing commands. An abbreviations facility and a macro facility allow users to customize and extend the standard commands. EMACS also has special language modes that let users compile programs without leaving the editing environment.

An online tutorial is available for users who are unfamiliar with EMACS. To find out about it, type:

```
EMACS EMACS*>TEACH-EMACS_INSTRUCTIONS
```

In addition, the following new manuals are available for EMACS users:

```
EMACS Primer (IDR6107)
EMACS Reference Guide (IDR5026)
EMACS Extension Writer's Guide (IDR5025)
```

These books are described in Chapter 1.

FUTIL

PROBLEMS FIXED

Fatal Errors: FUTIL now returns a severity code in the event of a fatal error.

Erroneous FROM Commands: When a FROM command causes FUTIL to report a syntax error, FUTIL now correctly resets the FROM-name to the current directory (*). Subsequent LISTF commands produce the correct names at the top-level BEGIN and END points (POLER #21145).

Passwords: FUTIL does not automatically convert passwords to uppercase. A warning to that effect is now issued the first time the user enters a password containing lowercase characters during a FUTIL session. The warning message is as follows:

Note: a password with lower-case characters will not be converted to upper-case by FUTIL.

This warning is for informational purposes only; the command that elicits it is still executed (POLER #22245).

CLEAN Command: The CLEAN command no longer resets protection on files below the current level to 7 0. Protection is reset to the original values (POLER #24563).

File Units: FUTIL no longer closes any file units when invoked. This correction allows FUTIL commands to be issued from within nested command files. Instead of closing file units, FUTIL finds units that are free and allocates them for itself. If FUTIL does not find at least six free file units, it issues the message, "All file units in use. I need at least 6 (FUTIL)." Under PRIMOS II, the corresponding message is, "ERROR CODE 41. I need at least 6 (FUTIL)." In any case, FUTIL uses at most 14 file units (POLER #15842).

COPY Command: The file types of all copied files are now preserved correctly (POLER #14942).

Comments: FUTIL now treats any command beginning with either * or /* as a comment.

FROM, TO, and ATTACH Commands: FUTIL now finds the correct volume when a volume name is included in a FROM, TO, or ATTACH command.

PRIMOS II: FUTIL no longer elicits a PRIMOS II "Beginning of File" error when sizing a DAM file under PRIMOS II.

LOGPRT

NEW FEATURES

At Rev. 18.4, three new LOGREC types and five new NETREC types are available for the LOGPRT command. The -TYPE option now accepts the following new arguments:

System Event Types (LOGREC Types):

<u>Type</u>	<u>Meaning</u>
MCHECK	Machine checks (not including memory parity)
QUIET	PRIMOS entering QUIET machine check mode (happens after 1024 ECC errors since cold start)
BADENT	Bad LOGREC entries

Network Event Types (NETREC Types):

<u>Type</u>	<u>Meaning</u>
NPXTHR	Error on transmit or receive
NPXRCV	Unanticipated receive status detected
NPXCLR	Calling process's circuit cleared with an unexpected clearing cause
NPXSEQ	Sequence error received in bounce detect
NPXCON	Unexpected circuit status received in call setup

Note

When network events are specified, the -NET option must precede the -TYPE option in the LOGPRT command line.

PROBLEMS FIXED

DELETE Option: The DELETE option now correctly spools the output file before deleting it.

HELP Display: The HELP display now uses screen scrolling. Typing any character(s) except uppercase or lowercase q, qu, qui, or quit continues the display.

DSWPARTY: In messages about DSWPARTY, what used to appear as "D board" now appears as "J board".

DSWPARTY checks are now performed for 850 processors as well as 750.

MAGNET

INTRODUCTION

A completely new version of MAGNET was made available as a qualified release product at Rev. 18.3. This new MAGNET is now fully released as part of the Rev. 18.4 Master Disk. Rev 18.4 MAGNET contains all the functionality of Rev. 18.3 MAGNET, as well as several new features.

For full information on MAGNET, refer to the Magnetic Tape User's Guide (DOC5027-183) and the Magnetic Tape User's Guide Update (UPD5027-184).

The following are the major additions to MAGNET between Rev. 18.3 and Rev. 18.4:

- Tape-to-spooler and disk-to-spooler support
- BREAK key handling
- Three new subcommands (CLOSE, NOISY, and SILENT)
- Nine new subcommand options (AT, CONTROL, COPIES, DEFER, FORM, ITEMS, LIKE, LINENOS, and SPOOL)
- New values for the PREACTION and POSTACTION options, for handling of spool files
- New error messages and a new error message format
- A new PAD option for specifying the padding character that is used to fill a logical record

These features are described below.

TAPE-TO-SPOOLER AND DISK-TO-SPOOLER SUPPORT

Tape and disk files can now be spooled from within MAGNET. The new SPOOL option can be used with the DECLARE and MODIFY subcommands to define spool objects. Many of the MAGNET functions that previously applied only to tape and disk objects now apply to spool objects as well. For example, the MOVE subcommand can be used to move logical files to and from spool objects as well as tape and disk objects. The new AT, CONTROL, COPIES, DEFER, FORM, and LINENOS options can be used to specify where, when, and how a spool object is printed.

Spool objects may be included in chains of objects, as defined by the PREVCHAIN and NEXTCHAIN options. In addition, the PREACTION and POSTACTION options now accept values that apply to spool objects (see below).

BREAK KEY HANDLING

MAGNET is now capable of handling BREAK key (CONTROL-P) interrupts by the user. Typing CONTROL-P or pressing the BREAK key returns the user to MAGNET subcommand level. From there, the user can enter more subcommands (including QUIT, to exit to PRIMOS). The new CLOSE subcommand (see below) can be used to close any disk, tape, or spool object that may have been left open by a BREAK (or any other abnormal condition).

NEW SUBCOMMANDS

The following are the subcommands added to MAGNET at Rev. 18.4:

- **CLOSE**: Closes a tape, disk, or spool object that may have been left open by an abnormal condition (for example, use of the BREAK key). The LIST command now displays the open status of declared objects. (See the description of the ITEMS option, below).
- **NOISY**: Enables the display of severity 1 messages at the terminal. This is the default mode for MAGNET, provided the -SILENT option is not specified on the MAGNET command line and the SILENT subcommand is not issued.
- **SILENT**: Suppresses the display of severity 1 messages at the terminal. Equivalent to specifying the -SILENT option on the MAGNET command line.

NEW SUBCOMMAND OPTIONS

The following are the subcommand options added to MAGNET at Rev. 18.4:

- **AT**: Specifies the location at which to print a spool file (DECLARE, MODIFY subcommands).
- **CONTROL**: Specifies the type of line printer carriage control desired (DECLARE, MODIFY subcommands).
- **COPIES**: Specifies the number of copies to be printed (DECLARE, MODIFY subcommands).
- **DEFER**: Specifies the time at which a spool file is to be printed (DECLARE, MODIFY subcommands).

- FORM: Specifies the forms to be used for printing (DECLARE, MODIFY subcommands).
- ITEMS: Specifies which types of objects should be listed (LIST subcommand).
- LIKE: Identifies an object whose options, option values, and translation edit tokens are to be duplicated for the object currently being defined (DECLARE, MODIFY subcommands).
- LINENOS: Specifies whether line numbers should be printed on output (DECLARE, MODIFY subcommands).
- SPOOL: Defines a spool file (DECLARE, MODIFY subcommands).

NEW VALUES FOR THE PREACTION AND POSTACTION OPTIONS

The PREACTION subcommand option now has a new possible value, SUPPRESS, for spool objects. SUPPRESS suppresses the printing of a banner page when a spool object is printed. (The banner normally shows the name that appears as the SPOOL option's value in the definition of the object.)

The POSTACTION subcommand option also has a new possible value, NOEJECT, for spool objects. NOEJECT suppresses the page ejection that normally occurs after printing is completed.

ERROR MESSAGES

The following changes have been made to the MAGNET error messages at Rev. 18.4:

- Some existing error messages have been modified to mention spool objects.
- Error messages have been added for the new subcommands and options.
- A new type of message (severity 0) has been created. Severity 0 messages are simply informational, and do not signal error conditions. These messages generally indicate that an operation has been completed or that the BREAK key has been used. The SILENT and NOISY subcommands do not affect the display of severity 0 messages; these messages cannot be suppressed.
- MAGNET error message format has been changed. The new format is as follows (nnn is the message identification number):

```
** Message nnn, Severity = n [, Object-name = name ]
   message
```


PAD OPTION

A new option (PAD) for tape objects has been included in Rev. 18.4 MAGNET. When files are written to tape, PAD specifies the padding character that is used to fill a logical record.

The default padding character in new-style operations is an industry-standard blank character (octal 040). In old-style operations, the padding character selected is either a PRIME ASCII blank character (octal 240), an EBCDIC blank (octal 100), a BCD blank (octal 000) or, for BINARY files, zero (octal 000), depending on the user's reply given to the "ASCII, EBCDIC, BCD or BINARY" question asked by MAGNET.

The PAD option is specified in the following manner:

```
PAD=('ccc'T)
```

where ccc specifies one or more characters and T is the type indicator. Note that the apostrophes must be specified.

There are eight padding types that can be selected, including industry-standard ASCII, PRIME ASCII, BCD, and EBCDIC, as well as BINARY, DECIMAL, OCTAL, and HEXADECIMAL numbers. These types are indicated by single letters as follows:

A	industry-standard ASCII
B	BCD
D	decimal
E	EBCDIC
H	hexadecimal
I	binary
O	octal
P	PRIME ASCII

Thus, to specify an EBCDIC blank as a padding character, one could use any of the following:

```
PAD=(' 'E)           (for EBCDIC)
PAD=('01000000'I)   (for binary)
PAD=('100'O)        (for octal)
PAD=('064'D)        (for decimal)
PAD=('40'H)         (for hexadecimal)
```

The type indicators (A, B, D, E, H, I, O, P) may be in either uppercase or lowercase. If A, B, E, or P is specified as a type, then only one character must appear between the quotes. If I is specified, eight digits must appear between the quotes. If D or O is specified, three digits must appear between the quotes. If H is specified, only two digits must appear between the quotes.

Note that if A, B, E, or P is selected, uppercase and lowercase characters are significant within the quotes. That is:

```
PAD=('K'E)
```

is not equivalent to:

```
PAD=('k'E)
```

The first example results in a padding value of octal 322, while the second results in a padding value of octal 222.

Because BCD is only a 64-bit code, lowercase characters specified map to uppercase pad characters.

Note

The code to read, write, and verify magnetic tape labels is contained within Rev. 18.4 MAGNET. However, this facility has not yet been fully tested, and the labels facilities should therefore not be used. If you specify LABELS= on the DECLARE or MODIFY subcommand line, a warning is displayed on your terminal.

MAGSAV/MAGRST

For information on new magnetic tape boot procedures, refer to Chapter 2.

NEW FEATURES

-TTY Option: The new -TTY option forces the following questions to wait for responses from the terminal (rather than from the current input stream, which may be a command file):

- MAGSAV's "Do you want to rewind tape?" (asked if the second or subsequent physical reel(s) of a save are not at load point)
- MAGRST's "Do you want to rewind tape?" (asked if the second or subsequent physical reel(s) of a restore are not at load point)
- MAGRST's "Continue with this reel?" (asked if the user mounts a continuation reel whose reel number is not in sequence)

New Option for Incremental Saves: A new command line option, -SAVE_UFD (abbreviated -SUFDF), is now available for MAGSAV. If -SUFDF is specified on an incremental save, an entry is saved for each UFD, regardless of whether any files in the UFD have been changed since the previous save. The -SUFDF option thus preserves tree structure so that the incremental save can be restored into an empty UFD without first restoring the original save.

If a UFD is saved on an incremental save, but none of the UFD's files were changed since the previous save, then an empty UFD will be created on restore.

The -SUFDF option can only be specified on incremental saves, that is, when -INC is also specified (POLER #41766).

Error Handling: Tape error handling in MAGSAV has been changed. Versions 2 and 3 of the tape controller (Integrated Formatter and GCR Formatter) now use the ERASE command to recover from tape write errors. Versions 0 and 1 (Kennedy Formatters) continue to use file marks for recovery.

Error recovery on labels has been improved. Error recovery will now be applied for every label instead of only for labels on the first logical tape. The number of retries during label writing has been increased from 5 to 20. After a label is successfully written after recovery has taken place, the number of retries is printed. If the label cannot be written, the reason for failure is printed. New error messages for label recovery are as follows:

- UNABLE TO WRITE FILE MARK
- UNABLE TO BACKSPACE
- UNABLE TO FIND FILE MARK
- UNABLE TO ERASE TAPE
- RECOVERY RETRIES EXCEEDED

PROBLEMS FIXED

Specifying Tape Units: The erase and kill characters can now be used when a tape unit number is input from the terminal.

If the specified tape unit is not assigned, the message "Device not assigned. Type 'S' to continue" is displayed, and the user is returned to PRIMOS. Typing 'S' causes the "UNIT NUMBER" prompt to be repeated so that the save/restore process can continue.

"Wrong Tape" Message: A multi-reel tape can now be searched for a logical tape that is not on the first reel without eliciting the "Wrong tape" error message.

Date Time Stamp: MAGRST now correctly sets the DIM on files restored from tape. It also sets the DIM and protection on UFDs if they do not already exist.

Open UFDs: MAGRST no longer leaves the current UFD open after checking to see if it is the MFD (POLER #40596).

Write Error Handling: If an unrecoverable write error occurs in MAGSAV, a message is now produced to inform you of the error recovery action taken.

Error recovery in MAGSAV consists of restarting the save at the last "checkpoint". Checkpoints occur:

- At the beginning of the logical tape
- At the beginning of the continuation reel
- At the last "NAME OR COMMAND" prompt

MAGSAV can only recover the save to the closest checkpoint. Thus, if more than one answer has been given to the "NAME OR COMMAND" prompt since the start of the current reel or logical tape, then MAGSAV cannot recover the whole save, but can only restart using the last response to "NAME OR COMMAND".

After an unrecoverable error, MAGSAV produces one of three error recovery messages, depending on which type of checkpoint is being used:

- If the reel number is 1 and there is only one "NAME OR COMMAND" answer since the start of the logical tape, then the message is "Restarting current logical tape".
- If the reel number is greater than 1 and there is only one "NAME OR COMMAND" answer since the beginning of the current reel, then the message is "Restarting current reel (reel n)", where n is the current reel number.
- If there has been more than one answer to "NAME OR COMMAND" during the current reel, then the message is:

****WARNING****

Unable to recover to beginning.

Restarting at name <answer to last prompt>

If you continue, you will have to keep the Reel which failed.

In this case, the failed reel will contain some of the save, and thus must be kept even though it is not complete. An alternative is to restart the entire save again.

\$A Command: The default action of the \$A command is now to search all disks rather than just logical device 0 (POLER #37054).

DOCUMENTATION CORRECTIONS

\$A Command: On page 7-12 of the System Administrator's Guide (PDR3109), under the NAME OR COMMAND request, the entry for the \$A command should read as follows:

\$A UFD [password] [ldisk] [key] changes the home UFD to the specified UFD on the disk ldisk (use password if required). To specify a sub-UFD, ATTACH first to its UFD, and then attach to the sub-UFD, giving the ldisk number and the key "2". \$A does not accept pathnames.

On page 2-47 of the PRIMOS Commands Reference Guide (FDR3108), the format for the \$A command should read: \$A UFD [passwd] [ldisk] [key].

PHYSAV/PHYRST

NEW FEATURES

-TTY Option: PHYSAV and PHYRST now both accept the -TTY option on the command line. If -TTY is specified, the magnetic tape unit number is requested at the user terminal, even if the current input stream is a command file.

Logical Tape 0: Logical tape number 0 is now legal. It is interpreted as the current (or next) logical tape number.

Expanded Writing Capabilities: PHYSAV can now save partitions of up to 40 heads. It can also now write more than one logical tape when used with an Integrated Formatter.

New EOT Handling: If EOT (end-of-tape) is detected while PHYSAV is erasing tape after a recoverable write error, the current block is not written to the current tape, but becomes the first block on the new tape. PHYSAV attempts to write trailer labels and continue on the next reel. The error message:

EOT DETECTED WHEN DOING ERROR RECOVERY SAVE WILL CONTINUE

is displayed.

PROBLEM FIXED

PHYSAV now correctly handles the REN command when a user QUITs after the question "WRITE NEXT LOG.TAPE (YES/NO)?" (POLER #32189).

RUNOFF

PROBLEMS FIXED

Table of Contents: Long headings in a table of contents are now underlined correctly (POLER #23242).

.DD: .DD statements without headings now work correctly (POLER #29153).

File Units: RUNOFF no longer uses file unit 2 instead of prompting for an output filename (POLER #29971).

Underlining: Underlining was not always working correctly when certain combinations of .INDENT and .PARAGRAPH commands were in effect. This problem has been solved (POLERS #31245, 34264, 35560).

Underlining was not occurring when RUNOFF found one or more blanks before the end underline delimiter. This problem has been corrected (POLER #34257).

Underlining now works correctly at the end of an insert file that comes at the end of the entire input file (POLER #34588).

Adjustment: A single word on a line is no longer split and "adjusted" (POLER #40187).

.COLUMNS Command: On pages with two columns, when the number of lines to be output matched the page length (as set by .LENGTH), RUNOFF was omitting the final page. This problem has been corrected (POLER #81527).

.DRESET Command: The .DRESET command now undents text correctly (POLER #82095).

SEG

NEW FEATURES

SPLLIB: SEG now automatically loads the SPL library (SPLLIB) whenever the pure FORTRAN library (PFINLB) is loaded by means of the LI or PL command.

Note

Because of the above change, Rev. 18.4 SPLLIB must be installed on your system if Rev. 18.4 SEG is to be run.

SPLIT Command: When the SPLIT command is used, the default stack address is now '4000/'150000. If you have a program which uses this range, you can restore the stack to its old address. To do so, invoke the SPLIT command as follows:

```
SP 100000 4000 170000 ext_stack_segno
```

The ext_stack_segno argument can be any segment that you are not currently using, with the exception of segment '4035.

PROBLEMS FIXED

Start_address: The start_address is now correctly initialized to the null pointer ('177777/0) rather than to 0/0 (POLER #29718).

File Units: SEG no longer dedicates any file units except unit 13, which is used for the maps (POLERS #20795, 37978).

Error Messages: SEG now reports an error if the user loads a SEG file in the VLOAD subprocessor (POLER #36524).

-LOAD Option: SEG no longer leaves a temporary file in the current directory when the -LOAD command line option is specified (POLER #34205).

Redefining Common Blocks: SEG now issues a warning if the user redefines a common block that is shorter than a segment to one that is longer than a segment (POLER #32068).

Stack Overflow and Extension Stacks: Stack overflow and extension stacks now work correctly (POLER #27021).

MODIFY Command: All reported problems with the MODIFY command have been corrected (POLERS #32187, 30102).

Old Runfiles: SEG now runs Rev. 15 and Rev. 16 runfiles correctly (POLER #41718).

NOTES

The following are responses to reported problems.

DELETE Subcommand: If the DELETE subcommand is issued from a command file at the beginning of a SEG session, and the file to be deleted does not exist, then SEG aborts the command file. However, the DELETE command is not needed at the beginning of a SEG session, since SEG truncates an existing file before writing to it. Thus, the DELETE command should be removed from the beginnings of command files (POLER #34484).

Start_address: When the SPLIT command is used, the start_address value indicated in the map has no meaning, and should be ignored (POLER #32730).

ECB Location: ECBs should be placed in the link frame rather than the procedure frame (POLER #36423).

DOCUMENTATION CORRECTIONS

The text in this section replaces the information on CMDSEG in:

- The System Administrator's Guide (PDR3109), pages 3-16 through 3-19 (in the section called Segmented Runfiles Saved by SEG's Loader).
- The LOAD and SEG Reference Guide (PDR3524), pages 7-7 and 7-8.

V-mode Segmented Runfiles

A segmented program cannot be run directly from UFD CMDNCO because PRIMOS's command processor cannot directly handle the SEG runfiles. The segmented program may be invoked by means of a non-segmented interlude program in CMDNCO.

The procedure for creating an interlude is:

1. Create the desired SEG runfile.
2. Copy the SEG runfile into the directory in which it will actually reside. The directory SEGRUN* is provided for system software; however, other directories may be used if so desired.
3. Run the CPL program SEG>CMDSEG using RESUME; it will ask for the SEG runfile pathname. This CPL program will create the interlude program with the runfile base name and the .SAVE suffix. You must be an owner of the directory to which you are attached while executing SEG>CMDSEG.

4. Copy the interlude program into CMDNCO.

Example

1. Extensions to the FARLEY utility described above make it desirable to compile and load it as a segmented program:

OK, <u>FTN FARLEY -64V</u>	Compile in 64V mode
0000 ERRORS [<u><.MAIN.>FTN-REV18.4</u>]	
OK, <u>SEG -LOAD</u>	Invoke SEG's loader
[<u>SEG rev 18.4</u>]	
\$ <u>LO FARLEY</u>	Load object file
\$.	Load other modules, if needed
\$.	Load other libraries, if needed
\$ <u>LI</u>	Load system library
<u>LOAD COMPLETE</u>	
\$ <u>SA</u>	Save runfile - FARLEY.SEG
\$ <u>QU</u>	Return to PRIMOS

2. Copy the SEG runfile into the directory in which it will reside during its invocation. In this example, the SEGRUN* UFD is used.

```
OK, FUTIL
[FUTIL rev 18.4.1]
> TO SEGRUN*
> TRECPY FARLEY.SEG
> QUIT
```

3. The CPL program SEG>CMDSEG.CPL creates the interlude program:

OK, <u>R SEG>CMDSEG</u>	Run CPL program
[<u>CMDSEG version 18.4</u>]	
Seg pathname: <u>SEGRUN*>FARLEY.SEG</u>	Enter runfile pathname
0000 ERRORS [<u><.DATA.>FTN-REV18.4</u>]	
[<u>SEG rev 18.4</u>]	
# <u>vload ENK\$RGWFKVZBCKT.SEG.T</u>	
\$ <u>co abs 4000</u>	
\$ <u>mi</u>	
\$ <u>sz 4000</u>	
\$ <u>s/li share4 130000 4000 4000</u>	
\$ <u>xp 1 2</u>	
\$ <u>sy map 4000 126000</u>	
\$ <u>s/lo b_ENK\$RGWFKVZBCKR.FIN.T 0 4000 4000</u>	
\$ <u>au 3</u>	
\$ <u>d/lo seg>cmdlib.bin</u>	
\$ <u>au 0</u>	
\$ <u>d/li vapplb</u>	
\$ <u>au 1</u>	
\$ <u>d/li</u>	

```

LOAD COMPLETE
$ re
# sh
TWO CHARACTER FILE ID: KT
CREATING KT4000
# delete
# q

```

4. The interlude program (here FARLEY.SAVE) is copied into the command UFD.

```

OK, FUTIL
[FUTIL rev 18.4.1]
> TO CMDNCO
> COPY FARLEY.SAVE
> QUIT

```

When FARLEY is entered at the user terminal, the FARLEY interlude program in CMDNCO is executed. This program executes the segmented runfile SEGRUN*>FARLEY.SEG.

If the SEG runfile requires only one segment of loaded information (procedure, link frames, and initialized common) in user space (segment '4000 and above), it is possible to include the interlude in the SEG runfile. This is discussed in the LOAD and SEG Reference Guide.

PROBLEMS OUTSTANDING

Old Runfiles: SEG does not run Rev. 14 runfiles (POLER #40634).

Segment '4035: SEG uses segment '4035 for its symbol table (POLER #36475).

Note

The Rev. 18.4 version of SEG.BUILD.CPL does not operate correctly with Rev. 18.2 SEG.

SUBROUTINES

Information on the following subroutines is provided in the section on PRIMOS earlier in this chapter: RDLIN\$, SEM\$OU, T\$CMPC, T\$LMPC, T\$MT, WILIN\$.

NEW FEATURES

The subroutine PHNIM\$ is now available. This subroutine allows a process to start up a CPL file or a command input file as a phantom. The usage is as follows:

```
dcl phntm$ entry (bit(16) aligned, char(32), fixed bin,
                fixed bin, fixed bin, fixed bin, char(128), fixed bin)
```

```
[call] phntm$ (cplflg, file-name, name-len, funit, phant-user,
              code, args, argsl)
```

where the arguments have the following meanings:

cplflg	Source of process. If true ('1'b), a CPL program is being started as a phantom. If false ('0'b), a command input file is being started as a phantom. (BIT(16) aligned = LOGICAL.)
file-name	The name of the file to be started as a phantom.
name-len	The number of characters in <u>file-name</u> .
funit	The file unit on which to open the phantom file.
phant-user	The user number of the phantom (returned).
code	A return code; zero means no error.
args	The arguments for a CPL phantom. A dummy argument must be given for non-CPL phantoms.
argsl	The number of characters in <u>args</u> .

This subroutine replaces the obsolete PHANT\$ subroutine.

PROBLEMS FIXED

PHNIM\$: The PHNIM\$ routine now always closes the input file by unit rather than by name, in case the user has the file open on another unit.

A\$FLOW: A missing key, A\$FLOW, has been inserted into SYSCOM>A\$KEYS.INS.PLL (POLER #29597).

UNIT\$A: UNIT\$A now returns FALSE on a bad unit number (POLER #33282).

DOCUMENTATION CORRECTIONS

The following correction applies to page 8 of the Prime Technical Update for Rev. 18 Subroutines (PTU78): The argument lists for the GV\$GET and GV\$SET routines should indicate that the VAR-NAME array must contain the size of the variable-name in the first word, and the name itself in the remaining words.

The following corrections apply to the Subroutines Reference Guide (PDR3621).

CSTR\$A: The following sentence should be added to the paragraph describing CSTR\$A on page 11-5: "However, if the actual lengths of the two strings (as specified in the argument list) are not equal, the function will be .FALSE.."

IFIX and INT: In Table 7-1 (page 7-3), the descriptions of IFIX and INT should be corrected. Both IFIX and INT return:

(sign of argument) * (largest integer <= absolute value of argument).

EDAT\$A: On page 11-20, in the second sentence of the description of EDAT\$A, the form of the date should be 'DD.MM.YY' instead of 'DD/MM/YY' (POLER #80457).

SPOOL\$: The Rev 18.3 Software Release Document (MRU4304-006) provided an update to the description of SPOOL\$ in the PRIMOS Subroutines Reference Guide (PDR3621). As a further update, the user should note that info(1), the first word of the array info, is now obsolete. (This word used to specify a temporary file unit.)

CHAPTER 5

DATA MANAGEMENT SYSTEMS

DBMS

NEW FEATURES

New DBUTIL Editor Command: A new DBUTIL editor command, DEN <entry-number>, has been created. This command deletes all but the last entry from the current leaf node of a set. (The command cannot be used to delete entries in index nodes, nor can it delete the last entry in a leaf node.) This command is useful in deleting entries in leaf nodes that have bad DBKs (data base keys).

New DBUTIL Facility: DBUTIL has acquired a new facility whose function is to mark as deleted any invalid record pieces within an area. (Invalid record pieces are those that are not connected to any current records.) A subsequent PACK will then reclaim the space. This utility need only be run once on any area. It is invoked as the second option (FIX2) of the "FIX AREA" command in DBUTIL.

New IDBMS Option: A new command line option, -NFCBS, is now available for IDBMS, the shared memory initialization routine for DBMS. This option allows a reduction in the initial allocation of file control blocks in shared memory. Such a reduction increases the number of different schemas an installation may access concurrently. Specify the -NFCBS option as follows:

```
SEG DBMSLB>IDBMS.SEG -NFCBS fcb-number
```

where fcb-number is an integer between 100 and 750 (inclusive) that specifies the desired number of file control blocks. If the -NFCBS option is not specified, or if any errors are encountered while processing the option, the number of file control blocks allocated defaults to 750.

The -NFCBS option need only be used if more than seven different schemas will be invoked concurrently.

New Error Message: Error message nn30F (-nn30), where nn is the number of the verb in execution, is now generated when too many arguments appear in the DMLCP call. The effective limit is currently 20 arguments (POLERS #41442, 41617).

FTRACE: Successive FTRACE dumps during the same day are now appended to the PDBMS>FTRACE file.

RECORD CURRENCY

DBMS maintains currency information within transactions only, not across transactions. Although an END TRANSACTION statement does not cancel currencies established by that transaction, reliance on that currency in a subsequent transaction produces unpredictable results. This is because of the possibility of an intervening transaction updating the data base before the original user's next transaction can begin. When an application program operates on a current record, that currency should be established in the same transaction as the operation using the currency.

For example, user A retrieves record WIDGET XYZ to establish the record as current, then ends the transaction. (See 1 below.) User B then begins an update transaction that fetches record WIDGET XYZ and subtracts 10 widgets before ending the transaction. (See 2 below.) Next, user A begins an update transaction to remove 10 widgets from record WIDGET XYZ. (See 3 below.) However, User A relies on the fact that record WIDGET XYZ is already current from the previous transaction and still contains the values that were in it then. In effect, since user A has no knowledge of the update by user B, user A's update cancels user B's previous update. The company could possibly be shipping non-existent widgets, or at best, think they have 10 more than actually exist.

USER A	USER B
1. START TRANSACTION, RETRIEVAL FETCH RECORD WIDGET XYZ END TRANSACTION	
2.	START TRANSACTION UPDATE FETCH RECORD WIDGET XYZ . . . Ship 10 widgets MODIFY RECORD WIDGET XYZ END TRANSACTION
3. START TRANSACTION, UPDATE . . . Ship 10 widgets MODIFY RECORD WIDGET XYZ END TRANSACTION	

PROBLEMS FIXED

DMLCP: DMLCP now establishes correct set currency to access NEXT/PRIOR in a set subsequent to the deletion of a record occurrence in that set.

Thus, a DELETE can be followed by a FIND NEXT or a FIND PRIOR (POLERS #15456, 41436, 45089, 24259, 37971).

The DMLCP command line -VALIDATE option (formerly the VERIFY option) is now able to find member records sorted by data base key (POLER #48084).

It is no longer necessary to open a file on unit 45 before using the DMLCP -VALIDATE option. If -VALIDATE is specified, DBMS asks for the name of a file to which the debug information will be written. This file is opened on a dynamically allocated unit.

DMLCP now checks that the run-unit table does not exceed a maximum size of 32,000 words (POLER #37380).

As of Rev. 18.3, retrieval transactions are no longer written to the after-image file. The screening process noted above is still valid, however, in order to allow for after-image recovery using after-image files created prior to Rev. 18.3.

If an owner is stored using a subschema that does not include all owned sets, a subsequent FIND OWNER using a subschema that does include all owned sets now updates set currency correctly (POLERS #40557, 45202).

After deletion of a record that is the last member of an ORDER NEXT set occurrence, the user no longer receives an internal fatal error when a subsequent record of the same type is stored and inserted into the same set occurrence (POLER #41492).

An error code of 26 (NO RECORD SATISFIES FIND) is now returned when a FIND is performed on an empty set occurrence whose owner was stored using a subschema not including the target set (POLERS #40529, 40537).

Within multi-member sets, successive calls to FIND NEXT or FIND PRIOR specifying different record types now either find the record or result in the appropriate non-fatal error (POLER #37463).

Mapping of bit strings to INTEGER*2 now moves all bits (POLER #34469).

DMLCP now correctly marks the status of deleted record partitions (POLER #47650).

As a schema is being closed, a check is now made to see that the user control block has not already been deleted.

An #EXIT DBMS following an Internal Fatal Error will now validate that the user control blocks and their links are correct.

The minimum amount of space required for a bucket to remain on the available space chain has been increased from 8 to 100 words. The effect of this reduction is to cut down on record fragmentation, thus reducing the number of disk accesses required and improving performance. This change may require that an EXPAND AREA be performed from within DBACP should the area become full (POLERS #47656, 40182, 40183).

DBUTIL: DBUTIL can now operate on a schema table of less than or equal to 32,767 words in size.

DBUTIL may now open up to 128 files.

When the VERIFY command encounters an inconsistency in a set file, it now displays the error and continues processing. It also validates the owner data base key (DBK) in the owner directory entries of a set file. A bad owner DBK is indicated by a -1 in the entry-number of the display.

A record dump now checks the status of record fragments and ignores those which have been deleted.

DBACP: DBACP now packs record fragments from most recent to oldest within a bucket, ensuring that the current fragment is chosen (POLER #47650).

DBACP EXPAND and PACK now reflect the change (from 8 to 100) in the number of words required to remain on the available space chain (POLERS #47652, 40183, 40182).

DBACP now rechains a bucket in which some records have been deleted, but none of whose remaining partitioned records can be packed.

DBACP now checks the validity of directory space pointers and record space pointers in VERIFY, EXPAND, and PACK AREA. All bucket pointers are checked for internal consistency.

A schema restore now updates all schema table information correctly when one or more subschemas reside on one volume and other data base files reside on another volume. A multi-volume restore proceeds as follows:

After restoring the schema table but before restoring any other data base files, DBACP scans the names of all volumes used by the saved schema and checks that all are available. For each volume that is not available, DBACP asks for a substitute. Only when all substitute volumes have been supplied and verified as available does the restore of the rest of the data base continue. If the user enters "NO" instead of a substitute volume name, the restore is halted. (POLERS #48007, 34245, 32190).

If the before-image file is on Volume A while the schema table is on Volume B, then VERIFY SCHEMA correctly reports the volume on which the schema table resides. In addition, a saved version of this schema will now also indicate the correct volume (POLER #31024).

If a subschema resides on one volume and all non-subschema files reside on a different volume, the subschema may now be deleted using the DBACP "DELETE SUBSCHEMA" command (POLERS #28882, 32706).

An area file may be expanded onto another volume, even when another data base file already exists on that volume (POLER #36992).

If a data base resides on volume A with the exception of some CALC files on volume B, and the data base is restored on a system where volume B is not available, DBACP is now able to restore all CALC files to a different specified volume (POLER #12002).

Although update transaction numbers as a group and read transaction numbers as another group are unique, update and read transaction numbers considered together are not unique. After-image recovery has been corrected to screen out retrieval transactions, thus ensuring valid roll forward capability.

SCHEMA: SCHEMA now generates a fatal error when the schema table size exceeds the maximum of 32,767 words (POLER #37380).

SCHEMA now flags as an error an attempt to declare a record more than once as a member record of the same set (POLER #34475).

SCHEMA now flags a bit string that exceeds the maximum length of 1056 bits (POLER #34475).

SCHED: The descriptions of data items output by SCHED now correspond to those in the schema. For example, S9(8)V99 in the schema is now described by SCHED as 99999999V99 (POLER #37614).

The size of an internal table has been increased so that adding a record with a large number of items is very unlikely to cause an overflow (POLER #34480).

CSUBS: If a virtual record section follows a set section in which only certain sets are copied (i.e. not 'COPY ALL SETS'), that virtual record section is now compiled (POLER #41616).

CSUBS now generates a fatal error if it detects that a subschema is too large for DBMS run-time to handle (POLER #37380).

Nested naming groups are now chunked appropriately (POLER #34490).

FSUBS: FSUBS now generates a fatal error if it detects that a subschema is too large for DBMS run-time to handle (POLER #37380).

Logical arrays are now flagged if they exceed the maximum size, currently 66 (POLER #34470).

A data item of type DATE is now chunked if the item is declared in the subschema as INTEGER*4 (POLER #33510).

CLUP: CLUP now closes only those file units (excluding l27, comoutput unit) which have the pathname prefix "<some_volume_name>PDBMS". In particular, this allows CLUP to be run from a CPL file.

Note

If the user opens files between the time the DBMS program ends and the time CLUP is run, CLUP may not have enough available file units to do the cleanup.

If CLUP is run from a CPL file that resides in the PDBMS UFD, that CPL file will also be closed (POLERS #40184, 35558).

PROBLEMS OUTSTANDING

CDML and FDML: CDML and FDML do not handle ON ERROR clause paragraph names that begin with numbers (POLER #12613).

A nondescriptive error message is returned when the user attempts to open a specific area (with only one record type) that has not been included in the subschema (POLER #41434).

A table overflow error is received when a fetch is attempted via current of set using nine data items (POLER #32202).

The preprocessor does not close the listing file after an error (POLER #20712).

The preprocessor does not close the output file if the input file contains no DML statements.

CLUP: If run after forced logout, CLUP may generate a RAM error.

DBACP: DBACP does not consistently close all opened files (POLERS #34472, 36602, 37900).

DBACP has no retry facility after a SAVE or RESTORE SCHEMA hits a bad spot on a tape. DBACP does not indicate whether a SAVE SCHEMA with tape errors has lost records (POLERS #34481, 36464, 43922).

If the after-image restore file treename includes passwords and is entered incorrectly, a pointer fault is returned to the user (POLER #32211).

DBACP may expand set files more than anticipated (POLERS #41637, 44226).

PACK and EXPAND may cause high I/O usage (POLERS #32867, 32869).

DBACP does not report the actual size of the data base (POLERS #40934, 41637).

Allocation of an empty area causes an error of 33 EOF (POLER #46238).

DBACP does not allow a user to verify an after-image tape and still turn after-image on after the verify (POLER #20531).

DBACP does not always restore the data base properly if the data base was saved without the after-image file (POLERS #34481, 35201, 40760).

DBACP gives nondescriptive error messages when it fails to open the DB# file (POLER #33542).

The DELETE SCHEMA command sometimes causes DBMS errors because the SD# entry was left around (POLERS #37900, 46653, 81975).

DMLCP: A delete generates an internal fatal error if the subschema does not contain member records of a set of which the deleted record is owner (POLER #44173).

DMLCP may not find all records contained within a subschema when the subschema is a subset of the entire schema (POLERS #48256, 45233, 40537).

DMLCP cannot reference subschemas larger than 64KB (POLERS #37381, 37380).

START TRANSACTION UPDATE is not flagged when the areas are opened in retrieval mode (POLER #34499).

An IF SET EMPTY statement always takes the EMPTY branch if the member record is not declared in the subschema (POLER #41635).

The IF MEMBER statement erroneously tests a record as being a member of a set from which it has already been removed (POLER #41436).

FETCH or FIND USING with an invalid database key gives an undocumented error message with minor code of 10. It should give a minor code of 12: "DATABASE KEY NOT AVAILABLE" (POLER #45820).

DMLCP gives an internal fatal error if the size of a search, sort, or calc key exceeds the limit of 30 words (POLER #43242).

CSUBS/FSUBS: When the area-id data base dataname is placed at the end of the subschema, the area-id does not function (POLER #40690).

CSUBS may produce the error message "INDEX OUT OF RANGE" if the schema table size exceeds 32K words and no RENAMING SECTION is used. A temporary solution is to include an empty RENAMING SECTION (POLER #44173).

If the phrase "set section" is followed immediately by a period, an error message is printed but no sets are copied (POLER #48256).

FSUBS does not indicate the line number of a duplicate element name (POLER #36876).

Use of data base datanames may result in errors if a record has some items not included in the subschema (POLER #41420).

CSUBS signals errors on the terminal only, not in the listing file (POLER #41449).

A subschema does not compile if the area section is followed by a period (POLER #82153).

RLIB: The transaction bit map may overflow if a user is hung or is very slow (POLERS #29298, 36006).

SCHED: If there is not enough space in the home bucket and record size is less than four bytes, the record may be overwritten with zeros (POLER #48051).

SCHED fails to stop after-imaging. Any after-images made after the edit session will not correctly roll forward the data base (POLER #48540).

SCHEMA: A data type of decimal or picture does not permit check range clause usage (POLER #23841).

A sign character in a picture clause in a schema has no effect. DBMS relies on the subschema in COBOL to enforce the presence or absence of a sign (POLER #34766).

DOCUMENTATION CORRECTIONS

The following corrections apply to the DBMS Administrator's Guide (PDR3276):

On page 9-39, the correct abbreviation for SUBSCHEMA should appear as SUB rather than SUBS (POLER #43213).

On page 11-6, the following text should be inserted just before the section on error messages:

Item Initialization

A new item added to a record description is initialized according to the description of the item. A numeric item is initialized to zeros; a non-numeric item is initialized to blanks.

DBMS/QUERY

NEW FEATURE

HELP Data Base: The DBMS/QUERY HELP data base has been expanded to provide more information on DBMS/QUERY commands (POLER #40185).

PROBLEMS FIXED

BREAK Key: If a user pressed the BREAK key during a Report Generator sort, the table being sorted was destroyed, and the files created by the sort routines were left open and not deleted. This problem has been corrected.

PICTURE Clause: The Report Generator no longer prints items of data type PICTURE "V999" one column too far to the right during a tabular display.

DBMS/QUERY no longer prints a null picture string when asked to describe an item of data type PICTURE "V999" or to describe a table containing an item of that data type.

DBMS/QUERY no longer generates a fatal error when displaying a PICTURE "V9999" field (POLER #40933).

Editor LOAD Function: The DBMS/QUERY editor LOAD function now works correctly (POLERS #48035, 48044).

File System Errors: DBMS/QUERY no longer displays the complete pathname (including passwords) of a file in the PDBMS UFD when a file system error occurs during file opening (POLER #37621).

File system errors during SAVE commands no longer close all DBMS/QUERY file units, causing INTERNAL ERRORS.

Footer Position: In list and block detail, page ejects no longer cause footers to be misplaced.

Footer position is now properly checked when margins are specified.

Block Detail: Block detail now allows use of position 1 of the detail line (POLER #45577).

PRINT Command: The PRINT command sometimes issued the error message, "An error occurred while trying to open the spool file" when a complicated data base was being accessed on a system with fewer than the default number of user segments configured. This problem has been corrected (POLER # 48075).

Justification: Zero replacement strings (strings following the ZERO option) are now right-justified instead of left-justified.

Any justification specifications that occur in list or block detail now correctly override default justification. Thus, to left-justify all items in a list (including numeric items, which are right-justified by default), the user must specify LEFT LIST detail.

Spurious Blanks: A problem that caused spurious blanks between the page heading and the first line of block detail has been corrected (POLER #44283).

DOCUMENTATION CORRECTION

On pages 11-30 and 11-31 of the DBMS/QUERY Reference Guide (IDR4607), the following should be added to the discussion of justification:

For a TABULAR DETAIL, default justification can be overridden on a field-by-field basis. However, for a LIST DETAIL, a justification can only be specified for all fields.

PROBLEM OUTSTANDING

If the SAVE command with the TO option is entered without a treename, all commands following the resulting syntax error are ignored until the user presses the BREAK key.

MIDAS

PROBLEMS FIXED

CREATK's MODIFY Option: The MODIFY option of CREATK now correctly modifies synonym (duplicate) support as specified by the user. ASCII key sizes are also changed correctly (POLER #32195).

GDATA\$: The GDATA\$ routine was returning an EOF error when it reached the end of the first data segment, instead of attempting to open a succeeding segment. This problem has been corrected (POLER #43079).

GDATA\$ now correctly returns a truncated record when the buffer size (specified in bytes) is between one-half and one times the actual data record length.

DOCUMENTATION CORRECTIONS

The following corrections apply to The MIDAS User's Guide (IDR4558):

On page 6-16, in the list of subroutines, "NEXT" should read "NEXT\$".

On page 6-32, in Table 6-7, the entry for FL\$FST should begin, "Tells FIND\$ to position to first entry..."

On page 6-39, under Argument Settings, the first sentence should read, "The index argument must be set to -1 regardless of the search method used."

On page 6-46, in the second paragraph, "FL\$PLN" should read "FL\$PLW".

On page 12-1, at the end of the section called EXAMINING THE TEMPLATE, the reference to Section 14 should be to Section 15.

In the Note at the bottom of page 12-8, the reference to Section 14 should be to Section 15.

On the top of page 15-8, the second sentence reads, "The user is asked to supply the segment directory length in words and..." This sentence should read, "The user is asked to supply the segment directory length in number of segment subfiles and..."

On page 15-9, in the section on RECLNT, the third sentence should read, "Users may change this value depending on what type of disk they are using." The following paragraph should be inserted between the two existing paragraphs in the RECLNT section:

The RECLNT parameter must be large enough to ensure that each index block can fit two key/pointer entries plus the 10 control words required in the last block level. However, RECLNT should not be given a value greater than 4095.

PROBLEMS OUTSTANDING

DISK FULL: MIDAS leaves a file in an inconsistent state when a DISK FULL error arises on a call to the file system. MIDAS aborts the process.

FIND\$, NEXT\$: MIDAS ignores the request to return data shorter than the actual record length in FIND\$ and NEXT\$ calls. It returns the full record instead.

File Unit Usage: When MIDAS receives the "All file units in use" error on calls to the file system, it aborts the process because it is unable to multiplex its file unit usage.

ADD1\$: ADD1\$ calls for direct access files can erroneously extend file allocation.

Concurrency: Under certain circumstances, the MIDAS semaphore shows user number = 0, yet no MIDAS processes are able to run.

KBUILD: In some cases, KBUILD is not able to add duplicate entries to an already populated index. A patch is available for this problem; consult your Prime field engineer (POLER #41858).

PRIME/POWER

NEW FEATURE

The *REORG and *LOG utilities (in POWERPLUS>TOOLS and POWERPLUS>POWER*) have been converted from R-mode to V-mode programs, and have been renamed to REORG.SEG and LOG.SEG, respectively.

PROBLEMS FIXED

Procedures: Variable names in procedures are now checked for valid characters. (The same restrictions apply as in the case of descriptor names.) Procedures with variables for UFD, PASSWORD, and FILENAME for CREATE commands now substitute variables correctly (POLER #27517).

"*" is now accepted in procedures to indicate the current UFD for the CREATE command (POLERS #37794, 35908, 32653).

CREATE: Field names defined by CREATE ADD sequences as synonyms for indexed fields now function properly in FINDs (POLER #45074).

Head of Form: POWER was failing to generate a head of form on the first page of a printout. This problem has been corrected (POLERS #41486, 31134).

Linked Fields: Data in linked fields is now displayed correctly (POLER #31542).

When a report was run on linked fields, blank lines were inserted for every owner record, whether or not member records were present. This problem has been corrected (POLER #34642).

A DISPLAY with six linked fields and left side headers now works correctly (POLER #35655).

Computed Fields: Computed fields (NL...Nx) now compare correctly to decimal file values (POLER #29583).

Reports: The report writer now prints descriptors that are two links away from the selected file correctly and in entirety (POLER #36575).

The existence of a mixture of security levels in a file no longer prevents the calculation of variables in a report (POLERS #27491, 32049).

Errors in display condition are now recognized during the report create dialog (POLER #34630).

Linked file descriptors are now used correctly during report creation (POLER #40130).

DISPLAY: DISPLAY USING FORM no longer aborts with an illegal access violation (POLER #24267).

Scrolling: Scrolling is no longer disabled after a SCREEN function on the PT65 (POLER #43029).

*REORG: The *REORG on the Master Disk had to be rebuilt before it could be used. This problem has been corrected (POLER #33285). (See also the section on NEW FEATURES, above.)

DOCUMENTATION CORRECTION

The following correction applies to the PRIME/POWER Guide (PDR3709):

On the sixth line of text from the bottom of page 9-25, "CREATING REPORTnn" should read "REPORT CREATE". (POLER #48062)

PROBLEMS OUTSTANDING

Reports: No page number appears on the last page of a report (POLER #42800).

REPORT CREATE gives incorrect member field prompts (POLER #37327).

REPORT CHANGE does not print old title lines correctly (POLER #41912).

DUMP: Descriptors with more than 60 characters are truncated during DUMP (POLER #42801).

FIND with DATE: FIND with DATE gives incorrect results (POLERs #45070, 45148).

CHANGE: "TO" is not optional in the CHANGE command (POLER #42799).

PRINT: The last two characters of a MIDAS record are sometimes truncated on PRINT ALL (POLER #45075).

Descriptor Names: Descriptor names are unnecessarily restricted. For example, a descriptor cannot be named E, El, D, or Dl (POLER #48172).

FIND: FIND sometimes selects a record twice (POLER #48171).

SUM: The SUM of PIC 9(8) (or larger) fields displays a spurious "\$" (POLER #41609).

Keywords: No error message is issued for a keyword of more than 20 characters (POLER #41850).

COMBINE: COMBINE of two large sets produces incorrect results (POLER #48162).

Audit Files: An overlay of a POWER audit file aborts (POLER #42797).

Vertical Display: The heading format is not used in vertical display.

Error Messages: The error message on CHANGE USING SCREEN is not erased.

EDIT: An EDIT on KEYWORD loses the last character of KEYWORD.

TERM USER: TERM USER does not work if the user's POWER* directory is not on Disk 0.

Command Priority: Individual command priority does not work if the user's name contains a period.

DECIMAL Descriptors: An internal routine error occurs during display of a DECIMAL descriptor with more than 24 characters.

CHAPTER 6

FORMS

FORMS

PROBLEMS FIXED

Field Attributes: Changing a field from protected to unprotected at run-time now works correctly.

NODISPLAY INPUT fields may now be used.

Function Keys: Function keys 11 and 12 now work correctly. The CLEAR key now returns number 14 when function keys are enabled. The ##FKEYS ON directive now works correctly.

3271 Support: FORMS can now be run on plug-compatible 3271 units.

Note

The default build of FORMS now uses the "_DD" versions of FM\$TCB. These versions cause FORMS to ignore the TCB* directory and go directly to FORMS* instead. To make use of TCB*, you must change the files FAP>C_SUBS, RUN>C_FMV, and RUN>C_FMR to use the version of FM\$TCB without the "_DD" suffix.

CHAPTER 7
COMMUNICATIONS

PRIMENET

NEW FEATURES

X\$KEYS: A PL/I version of the X\$KEYS insert file (named X\$KEYS.INS.PLL) has been added to SYSCOM.

PDN Parameters: For all supported Public Data Networks (PDNs), the NETCFG utility now queries each line's number of virtual circuits, default window size, and default packet size (POLERS #27592, 27602).

PROBLEMS FIXED

FAM I: FAM I now checks to see that there is a node configured for it to communicate with when the phantom starts up. If not, it prints the following message:

No nodes are enabled for FAM I
The FAM phantom is not needed and is automatically logging out

Note

PSS/IPSS: In the NETCFG utility, all references to PSS have been changed to IPSS.

DPTX

NEW FEATURE

Rev. 18.4 introduces a new way to optimize the performance of DPTX/DSC in heavy polling environments. The following is a description of DPTX/DSC optimization in general, and the Rev. 18.4 enhancements in particular.

Optimizing DPTX/DSC Performance in Heavy Polling Environments

DPTX/DSC is designed to operate in environments in which the number of polls (general and specific) being received is on the order of 2 to 6 per second. At installations where DPTX/DSC is heavily polled (20 to 30 times per second), significant performance problems can arise on the Prime CPU running DPTX/DSC. This problem is increased when the line involved is half duplex (HDX).

Should this performance problem arise at a DPTX/DSC installation, the System Administrator should first attempt all of the following:

- Switch from Specific Polls to General Polls if Specific Polls are being used exclusively. General Polls use the communications line more efficiently.
- Have the host site reduce the number of polls being sent per second by adjusting the communications front end.
- If the line is half duplex, have it changed to full duplex.

If these changes do not alleviate the problem, there is one other possibility. For Rev. 18.2, 18.3, or 18.4 systems, the MAXPOL parameter in DPTX-xxx>SOURCE>BSCGO.FIN can be modified so that BSCMAN will send EOTs to General Polls (not Specific Polls) on its own, without consulting the emulator.

The MAXPOL setting is an 8-word array (one word per line) passed by BSCGO at start-up. Each setting indicates the number of polls BSCMAN should answer (on the corresponding line) before passing a General Poll back to the emulator, EM3270. The default setting is 0. When BSCMAN answers a General Poll on its own, the emulator is never informed (and thus cannot respond with data bound for the host).

When the MAXPOL setting is raised, CPU usage by BSCMAN also increases, especially if the line is HDX. However, the emulator's CPU usage decreases, and the DPTX system misses fewer polls.

Note

Once BSCGO is modified, all of DPTX-xxx must be rebuilt and reinstalled.

Caution

In all cases, if MAXPOL is to be modified, DPTX-DSC must be configured for only one control unit on the troublesome line. At multi-drop installations, MAXPOL may only be modified if:

- DPTX is emulating only one control unit address on a line that is multi-dropped with other control units.
- The one control unit address being emulated is specified in the CUADDR array in DPTX-xxx>SOURCE, as described below.

Modifying MAXPOL at a multi-drop installation where these two conditions do not hold would cause DPTX to answer polls for other control units.

Additional Optimization at Rev. 18.4

At Rev. 18.4, additional tuning capability is available to allow modification of the MAXPOL parameter at multi-drop installations. The array CUADDR in DPTX-xxx>SOURCE can be modified so that BSCMAN will ignore all General Polls except those addressed to exactly one control unit. Set the CUADDR entry corresponding to the line that DPTX/DSC uses to the chosen control unit address (in EBCDIC). If the CUADDR entry remains 0 or if an incorrect control unit address is specified, the feature is disabled.

The CUADDR feature speeds up DPTX/DSC processing of General Polls on a multi-drop line where exactly one control unit address is being emulated. All Selects and Specific Polls are still processed by BSCMAN and the emulator. However, the overall use of CPU cycles by BSCMAN and the emulator is significantly reduced in the typical multi-drop installation. Furthermore, DPTX misses fewer polls. This is because there are fewer cases in which a General Poll arrives for DPTX while the emulator is still processing the previous General Poll (addressed to a different control unit). Thus, there are fewer cases in which a General Poll arrives for DPTX while the hardware receiver is off.

PROBLEMS FIXED

PA Keys: DPTX/TCF would not accept any PA key as the exit indicator. This problem has been corrected (POLER #42967).

OWLDSC Command: The OWLDSC command no longer causes an "Invalid Screen Address" error on a call to BD\$OUT (POLER #46862).

The OWLDSC command no longer leaves the keyboard locked when it receives a "transmission failed" message from BD\$INP (POLER #35639).

PROBLEMS OUTSTANDING

PT45DSC Command: As part of the DPTX/DSC product, the PT45DSC command transmits only data that has been modified. In certain cases this is inconsistent with the 3277 it emulates.

As part of the DPTX/DSC product, PT45DSC will not run correctly at 9600 bps. Because the PT45 is a slow device compared to the CPU, the PT45's buffers can be overrun, resulting in "broken" screen formats and scrambled messages (POLER #29480). The temporary solution is to run the PT45 at 4800 bps or to ensure that the DMQ size for that line is set at its default value.

DPTX/TSF: It is not possible to run DPTX/TSF with second generation BSC control units (3274 and 3276).

Alphabetic Data into NUMERIC ONLY Fields: In the DPTX/TCF product, if a host writes alphabetic data into a field defined as NUMERIC ONLY (which is legal in 3270 protocol), and this data is transferred to an IBM control unit attached to the Prime CPU via DPTX/TSF, subsequent updates of the virtual buffer by TCF will be rejected. TCF will report this problem as HOST DOWN. As a temporary solution, the application can be changed so that the field being written into is not NUMERIC ONLY. This can be done by changing the attribute sent out to define the field (POLER #32428).

ERASE INPUT Key: The ERASE INPUT key used on a terminal connected to an IBM control unit attached to the Prime CPU via DPTX/TSF will not cause the virtual buffer to be updated. This problem only affects those using TCF (POLER #35025).

DISLOG YES: Users of DPTX/TSF using DISLOG YES as a configuration option should be aware that specifying TTYNOP does not prevent the AMLDIM from performing logout abort checks when DISLOG YES is specified. The temporary solution is to not specify DISLOG YES as a configuration option (POLER #42909).

Hard Lock: In a WARM START condition, if a RING 3 user is attached to DPTX and is currently SELECTed, the "hard lock" will not be reset. (An 'unlock' message is not sent to the SELECTed user.) The temporary solution is to power the terminal off and back on again. The virtual buffer remains unchanged from the last successfully completed host transmission.

DOCUMENTATION CORRECTIONS

On page B-6 of the Distributed Processing Terminal Executive Guide (IDR4035) for Rev. 17, the last two lines of Table B-4 should read as follows (POLER # 20710):

```
X :00, :00, :00, :00, :00, :00, :00, :00, /* :160-:167
X :00, :00, :00, :00, :17, :00, :00, :52, / /* :170-:177
```

On page 8-24 of the Rev. 18.2 Software Release Document (MRU4304-005), the information describing PRINTER device specification should be changed to read as follows:

```
PRINTER { VFC
          }
          { PLATEN }
```

Specifies that the device is a printer. Either the VFC or the PLATEN option must be specified (but not both).

CHAPTER 8

THE FILE TRANSFER SERVICE (FTS)

INTRODUCTION

The File Transfer Service (FTS) is a separately-priced software product for transferring files between Prime computers over a communications network. It functions over any PRIMENET link, including RINGNET, full duplex, and half duplex, and also over a public data network (PDN) link, such as TELENET or DATAPAC.

This chapter provides an overview of FTS and tells how to use the FTR command to transfer files. Chapters 9 through 11 provide additional information on FTS.

FTS COMPARED WITH THE FAM

Both the File Access Manager (FAM) and FTS provide network file transfer. One or the other will be more appropriate for any specific application because they differ in several ways:

- The FAM provides immediate, direct access to a remote file system for all permitted file system actions. FTS provides queued file transfer only, which is not necessarily immediate.
- The FAM ties up your terminal throughout an entire file transfer. FTS leaves it free.
- The FAM requires the remote site to be up and the communications link to be working at the time you request a remote file transfer. FTS accepts requests for file transfers at any time, regardless of the state of the remote site or the communications link. FTS queues the requests, and transfers files at a time when the remote site and the communications are functioning.
- If the communications link (or either computer site) fails during a FAM transfer, you must try again later, manually. FTS retries automatically and performs the transfer when communications have been restored.
- Normally, the FAM must be constantly connected to the remote site. It therefore uses communications connect time even when there is no data traffic. If the communications link is expensive or scarce, FTS need only be connected from time to time, long enough to handle queued file transfer requests.

THE FTR COMMAND

Introduction

The FTR (File Transfer Request) command provides a method of transferring files between Prime computers that are connected via PRIMENET links. The FTR command is part of the File Transfer Service (FIS).

To transfer a file, you submit a request that details all the necessary information for the transfer to occur. You can make a request even when the communications link between the two relevant computers is not operational or the remote computer is down, because requests are queued on the local (requesting) computer.

Once you have submitted a request for a file transfer, you may display or cancel the request. The following sections explain these operations briefly. See Chapter 9 for a complete list of all FTR options and their uses.

FTR does not grant you any additional file access privileges such as passwords or protection attributes.

You can use FTR from within a command input file or from within a CPL file.

Note

At present only SAM and DAM type files can be transferred. Specifying any other type of file system object (such as a directory or segment directory) will cause the transfer to fail.

Source and Destination Sites: File transfers take place between sites. A site is a single computer, identified by a unique site name; Prime sites normally use their PRIMENET system names as site names. Files are transferred from a source site to a destination site. One of these must be your local site; the other is usually a remote site. (FTR cannot be used to transfer files between two remote sites.)

The following discussion assumes that you are using FTR between Prime machines that have been configured using the FTGEN command, explained in Chapter 11. To transfer files to or from sites that are not configured, see the descriptions of the -DSTN_SITE and -SRC_SITE options, under OPTIONS FOR SUBMITTING REQUESTS, in Chapter 9.

Request Names and Request Numbers: Each request has two means of identification associated with it: a request name and a unique request number. FTR assigns the number when you submit the request. You can use either the name or the number to identify the request. The name is either the name of the file to be transferred, or a specific name that

you assign to the request when you submit it. You usually refer to the request by its name. You may use the unique request number to distinguish between two requests that have the same name.

Submitting a File Transfer Request

Sending a File: To send a file to another Prime computer, use the FTR command with the following format:

```
FTR source_file destination_file -DSTN_SITE sitename
```

(Abbreviation for -DSTN_SITE: -DS)

source_file is the pathname of the file to be sent. You may give a filename if the file is in your current directory.

destination_file, also a pathname, specifies the name the file will be given at the destination site after it has been transferred.

Note

If the pathname for a source file or destination file contains directory passwords, the passwords must be included in the pathname, and the whole pathname must be included in quotation marks. For example:

```
'MARPLE CLUE>EVIDENCE'
```

-DSTN_SITE sitename specifies the name of the destination site.

FTR will give the following response to your request:

```
Request request-name (request-number) submitted.
```

request-number is the unique identification number assigned by FTR.

For example, assume you are on SYS2. The following dialogue illustrates how to submit a request:

```
OK, FTR 'CENTER KEY>REPORT' EXPOST>GROUP2>TEXT -DS SYS4
[FTR rev 1.0]
Request REPORT (65211149) submitted.
OK,
```

In this example, FTR queues a copy of the file REPORT in the UFD CENTER (with the password KEY) to send to system SYS4 for deposit in the directory EXPOST>GROUP2 under the name TEXT. The request name is the source filename, REPORT. The unique request number is 65211149.

Fetching a File: To fetch a file from another Prime computer, give the command:

```
FTR source_file destination_file -SRC_SITE sitename
```

(Abbreviation for -SRC_SITE: -SS)

source_file and destination_file are used as defined above for sending a file.

-SRC_SITE sitename specifies the name of the site where the file to be fetched is stored.

For example, assume you are on SYS2. The command:

```
FTR PEOPLE>LIST MYUFD>MYLIST -SS SYS6
```

will copy the file LIST in the UFD PEOPLE on SYS6 into the UFD MYUFD on SYS2 under the name MYLIST. The request name is the source filename, LIST.

Printing a File at a Remote Site: To print a hard copy of a file on a printer at another site, use the following format:

```
FTR source_file -DSTN_SITE sitename -DEVICE LP -DSTN_USER name
```

-DEVICE LP is the option that instructs FTR to print the file on a line printer at the remote site specified by -DSTN_SITE sitename.

-DSTN_USER name specifies the name of the person who should receive the printout at the remote site. The file will be printed with the name of the FTS server (set by your System Administrator) on the first line of the banner instead of a user-id. The name specified after -DSTN_USER will appear on the second line of the banner of the printed file. name need not be a user-id.

For example, assume you are on SYSA. The command:

```
FTR STUART>LETTER -DSTN_SITE SYSF -DEVICE LP -DSTN_USER JUDY_JONES
```

will cause the file LETTER to be printed at a line printer on SYSF. The first line of the banner will be the user-id of the FTS server (for example, "FTP"); the second line will be JUDY_JONES. The request name is LETTER.

Access

In order for FTS to work correctly, you, as a user, need the following access rights:

- Read access to source directories.
- Write and Delete access to any directory where you will create a log file (log files are discussed below).
- Read, Write, and Delete access to destination directories.
- Owner status in any directory where you will be creating a new file. For example, you need Owner status in the destination directory, and in any directory where a log file will be created.

Access rights are obtained according to the normal rules for using passwords. If the pathname of a source, destination, or log file contains passwords, you must include the passwords on the FTR command line. It is recommended that you use owner passwords as a matter of course, particularly in view of the last point in the list above. Source and destination directories that are password-protected should be protected in such a way that the passwords confer the rights indicated above.

If you wish to use FTS but do not want to communicate your passwords to people who will be sending you files, it is recommended that you create an unpassworded directory exclusively for FTS use, and tell people to deposit files there.

Checking the Status of Requests

Brief Reports: Once you have submitted a request you can check its status with the command:

```
FTR -STATUS request-name
           request-number
```

The -STATUS option returns a brief, one-line report for each of your file transfer requests identified by request-name or request-number. If you omit the request name or number, you will receive a report on all of your current requests. The report has the following form:

```
date.time user-id request-name (request-number) Status -category
```

date.time is in the form YY-MO-DD.HH:MI:SS. category will be one of the following:

```
waiting
transferring
put on hold by user
put on hold by operator
```

"waiting" indicates that the request is in the transfer queue but has not yet been transferred. "transferring" indicates that the transfer is in progress. The "hold" statuses indicate that a request is being indefinitely held in the transfer queue, either by the FTS system or by command of the user or the operator. (How to hold files is explained in Chapter 9.)

An example of checking request status is:

```
OK, FTR -STATUS
[FTR rev 1.0]
82-03-30.10:52:12 ELLEN CHAPTER (36949288) Status -waiting
OK,
```

In this example, ELLEN has one request, named CHAPTER, waiting to be transferred.

Full Reports: You can ask for a full report on the status of your requests with the command:

```
FTR -DISPLAY request-name
                request-number
```

Specifying request-name will print full information on all current requests with this name. Specifying request-number will print full information on the request with this number. If you omit request-name and request-number, FTR will print detailed information on all of your current requests.

The display takes the following form:

<u>Category</u>	<u>Information on the Request</u>
Request	Request-name (request-number)
User	User-id of submitter
Queue	Queuename where the request is queued
Queued	Date.time queued; Status - waiting, transferring, put on hold by user, or put on hold by operator
Last attempt	Date.time of transfer attempt; Attempts - number
Current time	Current date.time
Source file	Source pathname

Source file size	Number of bytes; displayed only if the source file is on the local site
Destination file	Destination pathname
Source site	Source sitename
Destination site	Destination sitename
Request log file	Pathname of log file; not always displayed
Log message level	Status of messages; not always displayed
Source user	A user-id (or another name) at the source site to be associated with the transferred file; not always displayed; useful when notifying a user at a remote site about a transfer
Destination user	A user-id (or another name) at the destination site to be associated with the transferred file; not always displayed; useful when printing files at remote sites or when notifying a user at a remote site about a transfer
Options	List of active options

If your user-id is SALLY on SYSB, you might use the -DISPLAY option with the following results:

```

OK, FTR PROSPECTUS JOHN>PROJECT -DSTN_SITE SYSC
[FTR rev 1.0]
Request PROSPECTUS (9635953) submitted.
OK, FTR -DISPLAY
[FTR rev 1.0]
Request      - PROSPECTUS (9635953)
User        - SALLY
Queue       - QUEUEA
Queued      - 82-03-30.16:03:59  Status - waiting
Last attempt - 00-00-00.00:00:00  Attempts - 0
Current time - 82-03-30.16:04:08
Source file  - <DISKA>SALLY>PROSPECTUS
Destination file - JOHN>PROJECT
Source site  - SYSB
Destination site - SYSC
Source user  - SALLY
Options :-
Binary, Copy, No Delete, No Source notify, No Destination notify.
OK,

```

If a Transfer Fails

In the event that a file transfer fails to complete successfully, one of two things will happen.

- If the error causing the failure does not preclude retrying, the request will be retried at a later time. (For example, the remote computer may be down, but will likely be up later.) The request is retried every 30 minutes, up to a maximum of 144 attempts (3 days), after which the request is put on hold.
- If the error precludes retrying (for example, if a user name or password is quoted incorrectly), then the request is suspended (put on HOLD). If -SN or -DN was specified on the FTR command line, FTS notifies the user for whom notification was requested. Either the submitting user or the system operator may subsequently delete the request on HOLD (using FTR -CANCEL), modify it (using FTR -MODIFY), or release it for another transfer attempt (using FTR -RELEASE), depending on the cause of the original transfer failure.

Logging Request Events

You can create an automatic log of file transfer request events by specifying the -LOG option, in the form -LOG pathname, when you submit a request. FTR will deposit logging information in pathname on the system originating the request. If pathname already exists, the logging information will be appended to the end of the file.

An example of use of the -LOG option is:

```
FTR SALLY>INFO WILLIAM>INFORM -DS SYSG -LOG SALLY>FTR.LOG
```

If the transfer of INFO was successful, the entries in the log file FTR.LOG would look like this:

```
15.55.13: [1.1] Request INFO (95423276) started Friday, April 23, 1982
15.55.14: [1.1] Submitting user is ANNE
15.55.14: [1.1] Local file is <DISKZ>ANNE>INFO
15.55.30: [1.1] RESULT: Transfer Terminated: Satisfactory and Complete.
15.55.30: [1.1] Request INFO (95423276) finished.
```

You can increase the detail entered in your log file by specifying on the FTR command line the -MESSAGE_LEVEL option with one of the following arguments: DETAILED, STATISTICS, TRACE. You must also specify the -LOG option. More information on the -MESSAGE_LEVEL option appears in Chapter 9.

Requesting Notification about Transfers

Receiving Notification Yourself: You may ask for notification about the progress of a submitted request by specifying on the FTR command line one of the following options:

<u>Option</u>	<u>Description</u>
-SRC_NTFY -SN	Source notify. Use -SRC_NTFY when you are sending a file.
-DSTIN_NTFY -DN	Destination notify. Use -DSTIN_NTFY when you are fetching a file.

You will receive messages at your terminal about your request. For example:

```
OK, FTR PROSPECTUS IDEAS>PROJECT -DS SYSC -SRC_NTFY
[FTR rev 1.0]
Request PROSPECTUS (9635953) submitted.

***FTSERV (user 109 on SYSB) at 16:41
Request PROSPECTUS (9635953) transfer started.
OK,

***FTSERV (user 109 on SYSB) at 16:41
Request PROSPECTUS (9635953) transfer ok.
OK,
```

In this example, the first message from FTSERV indicates that the file transfer has begun. The second message indicates that the transfer has been completed successfully. FTSERV is the name for the FTS server. (This name is set by the System Administrator, and may be different on your system.)

Notifying Others as Well as Yourself: You may specify both the -SRC_NTFY and -DSTIN_NTFY options together on the same command line if you wish to inform a remote source user or remote recipient, as well as yourself, of the progress of the transfer. In this case, you must also specify on the command line at least one of the following:

<u>Option</u>	<u>Description</u>
-DSTIN_USER user-id -DU	Required with -DSTIN_NTFY when you are sending a file so that FTR will know whom to notify
-SOURCE_USER user-id -SU	Required with -SRC_NTFY when you are fetching a file so that FTR will know whom to notify

For example, if your command line in the previous example had been:

```
OK, FTR PROSPECTUS JOHN>PROJECT -DS SYSC -SN -DN -DSTN USER JOHN
```

the messages listed above would have been sent both to you and to JOHN on SYSC.

Note

Because of the way `-SRC_NOTIFY` and `-DSTN_NOTIFY` work with the PRIMOS MESSAGE facility, you may not always receive all of your FTS notifications. It is recommended that you use the `-LOG` option instead of or in addition to `-SRC_NOTIFY` and `-DSTN_NOTIFY`.

Cancelling Requests

If you have submitted a request that is currently waiting in a queue to be transferred, you may cancel the request with the command:

```
FTR -CANCEL request-name
           request-number
```

For example, if the request you wished to cancel was named NEWS and the request number was 5684210, either of the following commands would cancel the request:

```
FTR -CANCEL NEWS
FTR -CANCEL 5684210
```

You would then receive the following message:

```
Request NEWS (5684210) cancelled.
```

You cannot cancel requests that are in the process of being transferred.

FTR's Help Facility

You can request a brief summary of FTR options at the terminal by simply specifying:

```
FTR
```

Requests on Hold

Under certain circumstances (such as an invalid directory password), FTR will automatically put the request "on hold". If you should receive a message to this effect, or see in an FTR -DISPLAY screen that your request is on hold, you may wish to do one of the following:

- Give the command:

```
FTR -CANCEL request-name
           request-number
```

This command will cancel the request, and you can resubmit.

- Give the command:

```
FTR -RELEASE request-name
           request-number
```

This command will instruct FTR to try the transfer again.

- Try to determine what may have caused the failure, and resubmit a corrected request.

Both you and the operator can hold a file intentionally with the -HOLD option. More information on this feature is in Chapter 9.

Other Options

The FTR command admits other options that allow you to modify, abort, and otherwise control your requests. Full information on these options (as well as on the -HOLD and -RELEASE options) appears in Chapter 9.

CHAPTER 9

FIS: FTR OPTIONS REFERENCE

INTRODUCTION

This chapter describes all of the options to the FTR command in two alphabetic lists. To learn which of these options you will need when transferring your files, see Chapter 8.

To see a summary of all FTR options, give the command

```
FTR
```

with no arguments.

There are two separate categories of FTR options: those used when submitting a file transfer request and those used to manage (display, modify, or cancel) an already-submitted request. Because the formats for using the two categories differ, and because one option name, -HOLD, occurs in both categories, two separate reference sections appear below: OPTIONS FOR SUBMITTING REQUESTS and OPTIONS FOR MANAGING REQUESTS.

OPTIONS FOR SUBMITTING REQUESTS

The options shown below are used when submitting requests for file transfer. They are used in combination with a source-file name in this format:

```
FTR source-file destination-file [options]
```

or, in the case of the -DEVICE option, in this format:

```
FTR source-file -DEVICE device-name [options]
```

Either source-file or destination-file may be a simple filename if the file is in the current directory on the local site, and if that directory is not passworded. Otherwise, the argument must be a pathname, which must be enclosed in single quotes if it contains a password. If a password is omitted or incorrectly specified for a local file, this error message will appear:

```
Passworded pathname must be fully qualified. (FTR)
```

Summary of Submittal Options

Here is a brief description of each submittal option:

<u>Option</u>	<u>Abbreviation</u>	<u>Purpose</u>
-COPY	-CPY	Transfer from copy of file, not from the original. (Default is -COPY.)
-DELETE	-DL	File deleted if transfer OK. (Default is -NO_DELETE.)
-DEVICE	-DEV	Transfer to a device. (Default is null. Only device is LP.)
-DSTN_NOTIFY	-DN	Send messages to destination user when file transfer starts and ends. (Default is no messages.)
-DSTN_SITE	-DS	Specify site to which file is to be transferred. (Default is local site.)
-DSTN_USER	-DU	Owner of file at destination site. (Default is null for remote destination, login user-id for local destination.)
-HOLD	none	Hold request on queue so it is not initiated. (Default is don't hold.)
-LOG	none	Pathname of request log file. (Default is no log.)
-MESSAGE_LEVEL	-MSGL	Level of detail of information entered in request log file. (Default is NORMAL, the minimum.)
-NAME	-NA	Specify the name of the request. (Default is name of source file.)
-NO_COPY	-NCPY	Transfer from the original, not a copy. (Default is -COPY.)
-NO_DELETE	-NDL	File not deleted if transfer OK. (Default is -NO_DELETE.)
-NO_DSTN_NOTIFY	-NDN	Don't send messages to destination user when file transfer starts and ends. (Default is no messages.)
-NO_SRC_NOTIFY	-NSN	Don't send messages to source user when file transfer starts and ends. (Default is no messages.)

-QUEUE	none	Name of queue onto which request is to be placed. (Default is the configured remote site queue, or OPEN_SYSTEM.)
-SRC_NOTIFY	-SN	Send messages to source user when file transfer starts and ends. (Default is no messages.)
-SRC_SITE	-SS	Specify site from which file is to be transferred. (Default is local site.)
-SRC_USER	-SU	Owner of file at source site. (Default is null for remote source, login user-id for local source.)

Details of Submittal Options

► -COPY

Abbreviation: -CPY

The -COPY option causes FTR to make a copy of the file to be sent to a remote destination. (-COPY has no effect on files being fetched from a remote source.) When FIS actually performs the queued transfer request, it will transfer the copy. -COPY is the default, and is the opposite of the -NO_COPY option.

-COPY allows the user to send a copy of the file in its current state, and then to modify, rename, or delete the original file without affecting the copy to be sent. (-NO_COPY causes the original file to be sent in the form to which it has been modified at the time of transmission.)

The copy to be sent will be deleted only upon completion of a successful transfer.

Specifying both -COPY and -NO_COPY options on the same command line is an error.

► -DELETE

Abbreviation: -DL

The -DELETE option to FTR specifies that the original file that was queued for transfer from the local site to a remote site is to be deleted after it has been successfully transferred. -DELETE has no effect on files being fetched from a remote site to the local site. -NO_DELETE is the default, and is the opposite of the -DELETE option.

Specifying both `-DELETE` and `-NO_DELETE` options on the same command line is an error.

► `-DEVICE device_name`

Abbreviation: `-DEV`

A file may be transferred to a device rather than to a file in the file system. The device name `device_name` is specified by the `-DEVICE` option. The only device name currently available is `LP`, for line printer. A file may only be sent, never fetched, to a device.

When the file is printed, the user banner will be "FIS" (the name of the FIS server), and the file banner will be "FIS_SPOOL_FILE", or the destination-user name (`-DU`), if specified.

► `-DSTIN_NOTIFY`

Abbreviation: `-DN`

The `-DSTIN_NOTIFY` option selects that FIS will send messages indicating the start of the file transfer and its termination to the logged-in destination user. FIS uses the MESSAGE facility of PRIMOS to send these messages, and cannot notify a logged-out user.

Note

The `-LOG` option is generally preferable to `-DSTIN_NOTIFY` for learning the progress and results of a file transfer.

This option is the opposite of the `-NO_DSTIN_NOTIFY` option, which is the default.

Specifying both `-DSTIN_NOTIFY` and `-NO_DSTIN_NOTIFY` options on the same command line is an error.

Note

To send the `-DSTIN_NOTIFY` message, FIS requires the name of the destination user. Use the `-DSTIN_USER` option to supply this name. Omission of the destination user name is an error, except when the destination site is the local site, in which case the submitting user's login name is used.

►-DSTIN_SITE destination-site-name

Abbreviation: -DS

The -DSTIN_SITE option specifies the site destination-site-name to which the file is to be transferred.

File transfers can take place between the local Prime site and a remote Prime site, but not between two remote Prime sites. Either the source site or destination site must be the local site. It is also permissible that both source and destination be the local site.

If no destination site is specified, the default is the local site.

The length of destination-site-name is limited to 128 characters.

Sites normally have been defined by the System Administrator, using the FTGEN command. The information held about each site includes:

- The site name
- The file transfer queue associated with the site
- The addressing information necessary to translate the site name into the physical address to establish the file transfer communication link

Site Configured

File transfer requests are usually made to destination sites that your System Administrator has already configured by using FTGEN. In this case, you need only specify the destination-site-name. FIS has the required additional information about the site in its stored configurations.

Site Not Configured (Open-Network Addressing)

On occasion, a file transfer may be required to a site that is not accessed frequently enough to warrant configuring it using FTGEN. In this case, the user must quote the open network address of the site as the destination-site-name.

This address should be prefixed by a : to distinguish it from a site name, and enclosed between single quotation marks to ensure that none of the characters of the address is interpreted by the command line processor (e.g., the + character).

Two examples of open-network addressing are:

```
:31109102345+FTP (PASSWORD)
```

```
SYS1+FTP (PASSWORD)
```

The first example references a network site unknown to the local system. The second references SYS1, which is known to PRIMENET (via NETCFG), but has not been configured for FTS (via FTGEN).

The address must include all the information necessary to establish the communication link to the destination site. In addition, the information usually obtained from the site's configuration must be supplied using other FTR control arguments:

- The queue into which this request is to be placed should be specified with the `-QUEUE` option.
- The queue specified must have been previously configured using FTGEN.
- If no queue is specified, then the default will be to place the request on the `OPEN_SYSTEM` queue, if it has been configured at the submitting user's site.

► `-DSTN_USER destination_user_name`

Abbreviation: `-DU`

The `-DSTN_USER` option specifies the destination user name. This name is the user name or owner of the file involved in the file transfer at the destination site. The destination_user_name must conform to Prime user naming conventions.

The length of destination_user_name is limited to 32 characters.

The default is the login user name if the destination site for the file is the local site, and null when the destination site is a remote site.

► `-HOLD`

Abbreviation: none

The `-HOLD` option causes the file transfer request being submitted to be marked as being held on the queue of requests waiting to be initiated. The request will not be initiated until it is released using the `-RELEASE` option of FTR. See `OPTIONS FOR MANAGING REQUESTS`, below.

Held requests can be released only by a user having the same user name as the request-submitting user, or by a user with operator privileges.

This `-HOLD` option on submission serves the same purpose as the `-HOLD` option described below under `OPTIONS FOR MANAGING REQUESTS`. The advantage of this `-HOLD` option over the other is that holding on submission ensures that the request will not be initiated immediately, as it would be if there were no requests already outstanding on the request queue.

► `-LOG` pathname

Abbreviation: none

This option controls the automatic logging of file transfer request events, such as submission, status/characteristic modification, initiation, and termination. The `-LOG` option is recommended as ordinarily preferable to the `-SRC_NOTIFY` and `-DSTIN_NOTIFY` options for learning the progress and results of a file transfer.

The file specified is a text file, similar to a command output file. An example of log file entries appears in Chapter 8, under Logging Request Events.

Whenever an event occurs to a file transfer request, such as its submission, initiation, or termination, the FIS subsystem records it.

If the specified log file already exists, new log entries are appended to it. If you always use the same log file name, entries will build up in it over a period of time.

If the log file specified does not exist, it will be created. A log file created by the FIS will have its read/write lock set to 2, to allow updating while other users are reading the file via `SLIST`, for example.

The length of this parameter is limited to 128 characters.

See the `MESSAGE_LEVEL` option for specifying the level of detail that is to be logged.

If no `-LOG` option is specified then no request log will be generated.

► `-MESSAGE_LEVEL` message_level

Abbreviation: `-MSGL`

The `-MESSAGE_LEVEL` option specifies the amount of information that is entered in the request log file specified by the `-LOG` option.

There are four levels of detail that can be specified using message_level, as follows:

<u>Message Level</u>	<u>Abbreviation</u>	<u>Function</u>
NORMAL	NRM	Only enter minimum details in the log. This is the default level.
DETAILED	DET	Log all events.
STATISTICS	STAT	DETAILED and STATISTICS information logged
TRACE	TRC	TRACE, DETAILED, and STATISTICS information logged.

If the `-MESSAGE_LEVEL` option is specified when submitting a request, a request log file must also be specified using the `-LOG` option; otherwise an error message will result.

The default level is NORMAL.

► `-NAME request-name`

Abbreviation: `-NA`

The name of the request is specified to be request-name. Further references to the request by its owner may use the name specified in request-name only if there are no other requests with that name belonging to the owner. Otherwise, the request number must be used.

The name specified by request_name should conform to the Prime file naming conventions; otherwise an error message will result.

If not specified, the request name defaults to the name of the file being submitted for transfer, stripped of pathname location information. For example, the default request name for "GARVIN>PUB>BACKROOM" would be "BACKROOM".

The length of request-name is limited to 32 characters.

► -NO_COPY

Abbreviation: -NCPY

The -NO_COPY option causes no copy to be made of the source file when it is queued for transmission to a remote site. (It has no effect when fetching from a remote site.) When FIS initiates the actual file transfer, the data transferred will be the actual contents of the source file at time of transfer.

Note

Any changes made to the source file between the time the file transfer request is made and the time transfer takes place will be included in the file transferred. If the file is deleted or renamed before transfer, the transfer will fail.

-NO_COPY is the opposite of the -COPY option. (The -COPY option results in a copy of the original file being taken and the copy being used in the file transfer.) Specifying both -NO_COPY and -COPY options on the same command line is an error.

The default if neither -NO_COPY nor -COPY is specified is to copy the original file.

► -NO_DELETE

Abbreviation: -NDL

The -NO_DELETE option to FTR specifies that the original file that was queued for transfer from the local site to a remote site is not to be deleted after it has been successfully transferred. This option does not pertain to files being fetched from a remote site to the local site. -NO_DELETE is the default, and is the opposite of the -DELETE option.

Specifying both -DELETE and -NO_DELETE options on the same command line is an error.

▶ `-NO_DSTIN_NOTIFY`

Abbreviation: `-NDN`

The `-NO_DSTIN_NOTIFY` option selects that no messages be sent by the file transfer server to the destination user indicating the start and termination of a file transfer.

If neither `-DSTIN_NOTIFY` nor `-NO_DSTIN_NOTIFY` has been specified, `-NO_DSTIN_NOTIFY` is the default.

Specifying both `-DSTIN_NOTIFY` and `-NO_DSTIN_NOTIFY` on the same command line is an error.

▶ `-NO_SRC_NOTIFY`

Abbreviation: `-NSN`

The `-NO_SRC_NOTIFY` option selects that no messages be sent by the file transfer server to the source user indicating the start and termination of a file transfer.

If neither `-SRC_NOTIFY` nor `-NO_SRC_NOTIFY` has been specified, `-NO_SRC_NOTIFY` is the default.

Specifying both `-SRC_NOTIFY` and `-NO_SRC_NOTIFY` on the same command line is an error.

▶ `-QUEUE queue-name`

Abbreviation: none

The `-QUEUE` option specifies the name of the queue queue-name on which the file transfer request is to be placed. Normally this option is not required, because a file transfer request will be placed on the default request queue for the remote site, as configured by the System Administrator using FTGEN.

If the remote site has not been configured using FTGEN, then this option should be used to specify the desired queue, which must have been previously configured using FTGEN. If no queue is specified for a non-configured site, the request will be by default placed on the `OPEN_SYSTEM` queue, providing it has been configured.

Specifying a queue using the `-QUEUE` option overrides any queue already defined.

The length of queue-name is limited to 32 characters.

► -SRC_NOTIFY

Abbreviation: -SN

The -SRC_NOTIFY option selects that FIS will send messages indicating the start of the file transfer and its termination to the logged-in source user. FIS uses the MESSAGE facility of PRIMOS to send these messages, and cannot notify a logged-out user.

Note

The -LOG option is generally preferable to -SRC_NOTIFY for learning the progress and results of a file transfer.

This option is the opposite of the -NO_SRC_NOTIFY option, which is the default.

Specifying both -SRC_NOTIFY and -NO_SRC_NOTIFY options on the same command line is an error.

Note

To send the -SRC_NOTIFY message, FIS requires the name of the source user. Use the -SRC_USER option to supply this name. Omission of the source user name is an error, except when the source site is the local site, in which case the submitting user's login name is used.

► -SRC_SITE source_site_name

Abbreviation: -SS

The -SRC_SITE option specifies the site source_site_name from which the file is to be transferred.

File transfers can take place between the local Prime site and a remote Prime site, but not between two remote Prime sites. Either the source site or destination site must be the local site. It is also permissible that both source and destination be the local site.

If no source site is specified, the default is the local site.

The length of source-site-name is limited to 128 characters.

Sites normally have been defined by the system administrator, using the FTGEN command. The information held about each site includes:

- The site name
- The file transfer queue associated with the site
- The addressing information necessary to translate the site name into the physical address to establish the file transfer communication link

Site Configured

File transfer requests are usually made from source sites that your System Administrator has already configured by using FTGEN. In this case you need only specify the source-site-name. FTS has the required additional information about the site in its stored configurations.

Site Not Configured (Open-Network Addressing)

On occasion, a file transfer may be required from a site that is not accessed frequently enough to warrant configuring it using FTGEN. In this case, the user must quote the open network address of the site as the source_site_name.

This address should be prefixed by a : to distinguish it from a site name, and enclosed between single quotation marks to ensure that none of the characters of the address are interpreted by the command line processor (e.g., the + character).

Two examples of open-network addressing are:

:31109102345+FTP (PASSWORD)

SYS1+FTP (PASSWORD)

The first example references a network site unknown to the local system. The second references SYS1, which is known to PRIMENET (via NETCFG), but has not been configured for FTS (via FTGEN).

The address must include all the information necessary to establish the communication link to the source site. In addition, the information usually obtained from the site's configuration must be supplied using other FTR control arguments:

- The queue into which this request is to be placed should be specified with the `-QUEUE` option.
- The queue specified must have been previously configured using `FTGEN`.
- If no queue is specified, then the default will be to place the request on the `OPEN_SYSTEM` queue, if it has been configured at the submitting user's site.

► `-SRC_USER source_user_name`

Abbreviation: `-SU`

The `-SRC_USER` option specifies the source user name. This name is the user name or owner of the file involved in the file transfer at the source site. The source_user_name must conform to Prime user naming conventions.

The length of source_user_name is limited to 32 characters.

The default is the login user name if the source site for the file is the local site, and null when the source site is a remote site.

OPTIONS FOR MANAGING REQUESTS

The options shown below are used to display information about submitted requests, or to control or modify requests in some way. They are used in the following format:

FTR option [request-name]

Summary of Management Options

The management options have no abbreviations. Here is a brief description of each management option:

<u>Option</u>	<u>Meaning</u>
<code>-ABORT</code>	Aborts a file transfer request.
<code>-CANCEL</code>	Cancels a file transfer request.
<code>-DISPLAY</code>	Prints the contents of one or more requests.
<code>-HOLD</code>	Delays a file transfer until a user or operator releases the request using the <code>-RELEASE</code> option.

-MODIFY	Modifies the characteristics of a request.
-RELEASE	Releases a held request.
-STATUS	Displays the status of one or more requests.

The Request Identifier

Each file transfer request has associated with it both a request name and a request number. Either the name or the number may be used as the request-identifier argument for the options shown below. The request identifier selects the particular request of interest.

The request name is the name of the file to be transferred or a name specifically assigned by the submitting user (using the -NAME control arguments on request submission).

The request number is a random-length number that is assigned by FTS to identify a request uniquely. The request number need be used only by a user who has two requests with the same request name, or by an operator in distinguishing between two requests that have the same request name.

The length of request-identifier is limited to 32 characters.

► -ABORT request-identifier

The -ABORT option is used to abort a request even if it is currently transferring. The request is suspended (put on hold) and can be released (by -RELEASE) for subsequent transfer, or cancelled (by -CANCEL).

If the request is already aborting, an error message will result.

A user may only abort a request submitted under his user name.

A user with operator privileges may abort any request.

► -CANCEL request-identifier

The -CANCEL option is used to delete a request from a file transfer request queue.

If the transfer is currently in progress, the request will not be cancelled.

A user may only cancel a request submitted under his user name.

A user with operator privileges may cancel any request.

►-DISPLAY [request-identifier]

The **-DISPLAY** option is used to obtain detailed information about the request request-identifier. This includes all the information given by the **-STATUS** option, and in addition, all the information that is included in the request itself. This information includes:

- Source and destination sites
- Whether the file is to be deleted after the transfer
- Request log file name
- Source and destination pathnames
- The queue on which the request is residing

If request-identifier is not specified, all the user's requests are displayed.

If request-identifier is a request name, then all requests belonging to that user with the specified request name are displayed.

If more than one request is to be displayed, the user will be prompted between requests for a decision to continue or not.

A user may only display contents of requests submitted under his user name.

A user with operator privileges may invoke the command for any request-identifier. If request-identifier is not specified, all requests of all users are displayed.

►-HOLD request-identifier

The specified request is marked as being held on the queue of requests waiting to be initiated. The request will not be initiated until it is released (see the **-RELEASE** option). If the transfer is already in progress, the command has no effect. (See the other **-HOLD** option, above under **OPTIONS FOR SUBMITTING REQUESTS**, which can hold a request even if none other is pending.)

A user may only hold requests submitted under his user name.

A user with operator privileges may hold any request.

►-MODIFY request-identifier control-argument...

Once a request has been submitted and before it has been initiated, most of its characteristics can be modified by any user whose user name matches that of the owner of the request.

If control-argument is omitted, -MODIFY resets the date and time of the last transfer attempt, zeros the number of retries, and causes the request to be retried as soon as possible, rather than waiting until the 30 minute retry interval has expired. This can be useful, for example, when a previously inoperative site becomes operational and the user would like his pending request to be retried as soon as possible.

The control-argument can be any of the following request submittal options:

- DELETE
- DSTN_NOTIFY
- DSTN_USER
- DEVICE
- LOG
- MESSAGE_LEVEL
- NAME
- NO_DELETE
- NO_DSTN_NOTIFY
- NO_SRC_NOTIFY
- SRC_NOTIFY
- SRC_USER

For details of these options, refer to OPTIONS FOR SUBMITTING REQUESTS, above. (The options -COPY, -DSTN_SITE, -HOLD, -NO_COPY, -QUEUE, and -SRC_SITE may not be used as control arguments.)

Modifying the characteristics of a request is similar to cancelling a request and resubmitting it. However, the request will remain in the same position in the queue. Therefore, modifying a request will not generally affect the time of request initiation, whereas cancelling and resubmitting a request would probably delay initiation time.

A user may only modify requests submitted under his user name.

A user with operator privileges may modify any request.

►-RELEASE request-identifier

A file transfer request request-identifier previously held by either the -HOLD Request Management option, the -HOLD Request Submittal option, or the FIS system is released, thus allowing the request to be eligible for initiation. If the request was not held, then an error message results.

A user may only release a request held under his user name. The user may not release a request that was held by a user with operator privileges, even though he submitted the request.

A user with operator privileges may release any request.

►-STATUS [request-identifier]

The -STATUS option displays information about the current status of the request request-identifier. The information returned by -STATUS for each request is:

- Date and time the request was queued
- User name of the submitting user
- Name and number of the request
- Current status of the request

A user may display only the status of requests submitted under his user name.

If request-identifier is not supplied, all requests owned by the user are displayed.

If request-identifier is a request name, then all user's requests having the specified request name are displayed.

If more than a page (22 lines) is to be displayed, the user will be prompted at the end of each page for a decision to continue or not.

A user with operator privileges may invoke the command for any request-identifier. If request-identifier is not specified, all requests of all users are displayed.

CHAPTER 1

INTRODUCTION

REV. 18.4 SOFTWARE RELEASE DOCUMENT

This document contains information on technical changes and enhancements to Prime user software for Rev. 18.4.

This guide contains information on the following topics:

- Installation of Rev. 18.4
- New software products released at Rev. 18.4
- Enhancements made to existing software
- Software problems fixed
- Documentation corrections and additions
- Software problems outstanding

Within each chapter, the information on each product begins on a new right-hand page. Pages can thus be extracted from this book and placed in other manuals as necessary.

Note

The Rev. 18.4 Software Release Document is designed to supplement other manuals. Its pages are not replacement pages, and its pagination does not correspond to the pagination of other books.

The remainder of Chapter 1 provides information on the following topics:

- Overview of Rev. 18.4
- New Prime technical publications
- Installation of Rev. 18.4

CHAPTER 10

FIS FOR SYSTEM OPERATORS

INTRODUCTION

This chapter explains the commands that a computer operator will use in day-to-day maintenance of the File Transfer Service. The operator has the task of initiating file transfer servers and monitoring their running.

The operator has two tools with which to control the file transfer system:

- The FTR command, to monitor and control individual file transfer requests.
- The FTOP command, to control the file transfer server processes.

OPERATOR USE OF THE FTR COMMAND

The FTR command, by which users submit and monitor their file transfer requests, is described in detail in Chapters 8 and 9. The operator, who has gained special privileges by logging in as SYSTEM, uses this command to manage all users' file transfer requests. The operator may use any of the following FTR options on any file transfer request:

- ABORT Abort a file transfer request.
- CANCEL Cancel a request from the request queue.
- DISPLAY Display detailed information about a request.
- HOLD Delay file transfer initiation.
- MODIFY Modify a request's characteristics.
- RELEASE Release a held request.
- STATUS Display the status of a request.

Chapter 9 describes the use of these options in detail.

Note

Although most users will be able to ignore the request numbers by which FTS identifies transfer requests, the operator will need to become familiar with them.

THE FTOP COMMAND

FTOP allows the operator to start, stop, and monitor the operation of the file transfer servers. A server is a phantom process that handles file transfer requests of a single queue. Server processes must be started from the supervisor terminal. Each server takes requests from a file request queue, of which there may be up to eight. An additional special server, YTSMAN, receives file transfer requests from remote sites and passes them to appropriate local servers.

The FTOP command is available only to a user with operator privileges, which he gains by being logged in as SYSTEM.

The general format of the FTOP command is:

FTOP option

One option, -START_MNGR, applies to the FTS manager process, YTSMAN. All the other options apply to ordinary server processes.

Summary of FTOP Options

<u>Option</u>	<u>Abbreviation</u>	<u>Purpose</u>
-ABND_SRVR	-ASV	Abandon an FTS server process
-ABRT_SRVR_LINK	-ASVL	Abort an FTS server link
-LIST_SRVR_STS	-LSVS	List server status
-START_MNGR	-STRMG	Phantom the FTS manager process (YTSMAN)
-START_SRVR	-STRSV	Phantom an FTS server process
-STOP_SRVR	-STPSV	Stop an FTS server process

To obtain a display of FTOP usage, enter the command

FTOP

with no options.

FTOP Options

All the options to the FTOP command are presented below in alphabetical order.

▶ -ABND_SRVR server_name

Abbreviation: -ASV

This option causes the file transfer server server_name to immediately abort all current file transfers, placing the requests on hold, and to log out.

If server_name is not running, an error message will result.

Note

The recommended way to stop a server is the -STOP_SRVR option. Forced LOGOUT of a server is specifically not recommended.

▶ -ABRT_SRVR_LINK server_name link_number

Abbreviation: -ASVL

This option causes the file transfer server server_name to abort the current file transfer on link link_number, placing the request on hold. The server continues running; it does not logout.

Each file transfer server may handle up to 8 concurrent file transfers. To find the link number of an ongoing transfer, use the command:

FTOP -LIST_SRVR_STS server_name

which lists transfers, identifying each by its link number, in the range 1 to 8.

If server_name is not running or link_number is not active, an error message will result.

▶ `-LIST_SRVR_STS [server_name]`

Abbreviation: `-LSVS`

This option lists the status of the server server_name. It indicates the state of the actual server, that is, whether it is running or not, and lists the state of each of the 8 possible file transfers that the server may be running. Each transfer is identified by a link number in the range of 1 to 8.

If no server_name is specified, the status of all the configured servers will be listed.

▶ `-START_MNGR [manager_name]`

Abbreviation: `-STRMG`

This option starts (phantoms) the FTS Manager process. The default manager_name is YTSMAN.

Note

The command `FTOP -START_MNGR` should only be invoked from the supervisor terminal. From any other terminal, an error message will result.

The command to start up the manager may be added to the PRIMOS cold start `C_PRMO` file.

▶ `-START_SRVR server_name`

Abbreviation: `-STRSV`

This option starts (phantoms) the file transfer server server_name. If that server is already running, an error message is displayed.

Note

The command `FTOP -START_SRVR` should only be given from the supervisor terminal. Doing so ensures that the server will be phantomized with a user name of server_name and the process priority and timeslice will be automatically set in accordance with the configuration of the server.

Invoking this option from a terminal other than the supervisor terminal would result in server_name not running under its proper user name and the server's configured priority and timeslice not being set. The system defaults for these values would be assigned instead.

The commands to start up the required file transfer servers may be added to the PRIMOS cold start C_PRMO file.

► -STOP_SRVR server_name

Abbreviation: -STPSV

This option causes the file transfer server server_name to complete the file transfers that are currently in progress, and then logout. If the server is not running, an error message is displayed. Use of this option is the recommended method for closing down an FTS server.

CHAPTER 11

FIS FOR SYSTEM ADMINISTRATORS

INTRODUCTION

This chapter explains the commands that the System Administrator will use in setting up, or configuring, the file transfer system (FIS) at a particular computer site.

The System Administrator uses FTGEN, an interactive program for defining, modifying, and displaying the characteristics of file transfer queues, servers, and sites.

THE FTGEN COMMAND

To use the FTGEN subsystem the System Administrator, who must be logged in as SYSTEM, enters:

FTGEN

FTGEN responds with the prompt:

FTGEN>

and awaits an FTGEN command. For example (user input shown underlined):

FTGEN> INITIALIZE_FIS

There are four categories of FTGEN commands:

- General commands
- QUEUE configuration commands and subcommands
- SERVER configuration commands and subcommands
- SITE configuration commands and subcommands

Each category is described separately below. Commands are alphabetical only within each category.

Subcommands

Each of the FTGEN commands ADD_QUEUE, MODIFY_QUEUE, ADD_SERVER, MODIFY_SERVER, ADD_SITE, and MODIFY_SITE displays a special prompt and awaits subcommands. For ease of FTS configuration, many of the subcommands required for configuring queues, servers, and sites have default values. Thus, certain subcommands are not required unless a parameter value other than the default is desired. FTS holds templates of default configurations so that future queues, servers, or sites with the same attributes may be configured by using only the appropriate ADD command (with the name of the new queue, server, or site) and the FILE subcommand. If the new queue, server, or site is to differ in some way from the default template, only those subcommands appropriate to define the difference need be used.

GENERAL COMMANDS

The following commands apply to the file transfer configuration as a whole:

STATUS	Display the current status of the FTS configuration.
INITIALIZE_FTS	Initialize the FTS Subsystem data base.

► STATUS

Abbreviation: ST

The STATUS command displays the current state of the FTS configuration, including the number of queues, sites, and servers that have been configured.

The STATUS command also checks whether the contents of the FTS Subsystem data base is valid, displaying a message if it is not.

► INITIALIZE_FTS

Abbreviation: IFTS

The IFTS command initializes the FTS Subsystem database to a consistent state, only creating and initializing files if they do not already exist. It then displays a listing of the current state of the FTS configuration.

This command should be used in the following circumstances:

- After installation of the FIS subsystem data base directory (FISQ*) and before any FTGEN commands to configure sites, queues and servers.
- To attempt to recover from the rare situation where the data base has been invalidated.

Executing IFIS will not adversely affect a correctly configured and running system.

QUEUE COMMANDS

The queue commands configure the queues that hold local file transfer requests:

ADD_QUEUE	Add a new queue.
BLOCK_QUEUE	Block submissions to a queue.
DELETE_QUEUE	Delete a queue.
LIST_QUEUE	List a queue configuration.
MODIFY_QUEUE	Modify an existing queue.
PURGE_QUEUE	Delete all requests from a queue.
UNBLOCK_QUEUE	Unblock a queue.

► ADD_QUEUE queue_name

Abbreviation: AQ

The ADD_QUEUE command adds the new queue queue_name to the FIS configuration. This command prompts the user for subcommands to define the attributes of the queue. See Queue Subcommands, below.

The queue_name must conform to FIS queue name conventions. If the queue already exists an error message is displayed.

▶BLOCK_QUEUE queue_name

Abbreviation: BQ

The BLOCK_QUEUE command blocks the queue queue_name. This stops any user from adding requests to that queue.

Use the UNBLOCK_QUEUE command to allow users to submit requests to the queue once again. The default state of a queue is unblocked.

▶DELETE_QUEUE queue_name

Abbreviation: DQ

The DELETE_QUEUE command deletes the existing queue queue_name. Only when a queue is empty and not currently serviced by a server can it be deleted.

▶LIST_QUEUE { queue_name
-ALL }

Abbreviation: LQ

The LIST_QUEUE command displays the characteristics of the queue queue_name.

If -ALL is specified instead of the queue_name then all queues are displayed.

▶MODIFY_QUEUE queue_name

Abbreviation: MQ

The MODIFY_QUEUE command modifies an already existing queue queue_name. This command prompts the user for subcommands to modify the attributes of the queue. See Queue Subcommands, below. If the queue does not already exist, an error message is displayed. A queue may not be modified if it is currently being serviced by a server process.

▶PURGE_QUEUE queue_name

Abbreviation: PQ

The PURGE_QUEUE command deletes all requests from the queue queue_name. Requests that are being processed at the time are not deleted.

► UNBLOCK_QUEUE queue_name

Abbreviation: UBQ

The UNBLOCK_QUEUE command unblocks the queue queue_name after that queue has been blocked by the BLOCK_QUEUE command. Unblocking a queue re-opens it and allows a user to submit requests to it.

Queue Subcommands

The ADD_QUEUE and MODIFY_QUEUE commands prompt the user for subcommands to configure the queue.

The general queue subcommand format is:

queue: subcommand

where queue: is a system prompt.

The queue subcommands are:

BLOCK_QUEUE	Block the queue.
FILE	File the queue configuration.
LIST_QUEUE	List the queue configuration
LOG	Set the queue log file.
MAXIMUM_REQUESTS	Set the maximum size of the queue.
MESSAGE_LEVEL	Set the queue log message level.
QUIT	Quit the queue configuration.
UNBLOCK_QUEUE	Unblock the queue.

When a new queue is added, the default configuration values can be used, in which case the only subcommand that is needed is FILE.

► BLOCK_QUEUE

Abbreviation: BQ

The BLOCK_QUEUE subcommand blocks the current queue. This prevents any user adding requests to this queue.

The UNBLOCK_QUEUE subcommand allows users to submit requests to the queue. The default state of a queue is unblocked.

► FILE

(No abbreviation)

The FILE subcommand files the queue configuration and returns the user to FTGEN command level. If the user does not wish to file the configuration, she may issue the QUIT subcommand instead of FILE.

► LIST_QUEUE

Abbreviations: LQ, L

The LIST_QUEUE subcommand displays the current queue configuration. When adding a new queue, it is a simple way of displaying the configuration parameters that need to be set and their initial defaults. At any other time it allows the user to check that the configuration is correct before filing.

► LOG { filename }
 { -OFF }

(No abbreviation)

The LOG subcommand specifies the filename of the log file for the current queue. All significant queue events are logged in this file under control of the MESSAGE_LEVEL subcommand. The log file resides in the FTSQ* directory.

A single subsystem-wide log may be specified, or a separate log file may be defined for each queue.

If -OFF is specified instead of the filename, no logging is performed for this queue.

The default is -OFF.

► MAXIMUM_REQUESTS number_of_requests

Abbreviation: MAXR

The MAXIMUM_REQUESTS subcommand specifies the maximum number of requests, number_of_requests, that may be held on the current queue.

This subcommand is only valid when the queue is being added to the configuration (using ADD_QUEUE). Once the queue has been configured, the maximum number of requests cannot be modified with the MODIFY_QUEUE command.

The default is 16384 requests.

► MESSAGE_LEVEL message_level

Abbreviation: MSGL

The MESSAGE_LEVEL subcommand specifies the level of information that is entered in the queue log file as defined by the LOG subcommand.

There are four levels of detail that can be specified using message_level, as follows:

<u>Message Level</u>	<u>Abbreviation</u>	<u>Function</u>
NORMAL	NRM	Only enter minimum details in the log. This is the default level.
DETAILED	DET	Log all events.
STATISTICS	STAT	DETAILED and STATISTICS information logged.
TRACE	TRC	TRACE, DETAILED, and STATISTICS information logged.

The default level is NORMAL.

▶ QUIT

Abbreviation: Q

The QUIT subcommand quits the current queue configuration and returns the user to the FTGEN command level. If the queue is being added, no record of it will be filed. If it is being modified, the old configuration will remain unchanged.

▶ UNBLOCK_QUEUE

Abbreviation: UBQ

The UNBLOCK_QUEUE subcommand unblocks the current queue, allowing users to submit requests to that queue. The default state of a queue is unblocked.

Note

If a file transfer request uses the "open networking" form of addressing for the source or destination site, and if no -QUEUE option is specified in the command line, then the request is placed on the OPEN_SYSTEM queue, if such a queue is configured. The system administrator should therefore consider configuring the OPEN_SYSTEM queue as an aid to users. If a log file is created for OPEN_SYSTEM, then the administrator can track the local use of "open network" requests, and decide whether frequently addressed sites should be configured in the FTS database.

SERVER COMMANDS

The Server commands configure the FTS phantom server processes, which service both local request queues and incoming remote requests.

ADD_SERVER	Add a new server.
DELETE_SERVER	Delete a server.
LIST_SERVER	List a server configuration.
MODIFY_SERVER	Modify an existing server.

▶ ADD_SERVER server_name

Abbreviation: AS

The ADD_SERVER command adds a new server server_name to the FTS configuration. This command prompts the user for subcommands to define the attributes of the server. See Server Subcommands, below.

The server_name must conform to FTS server_name conventions. If the server already exists, an error message is displayed.

▶ DELETE_SERVER server_name

Abbreviation: DS

The DELETE_SERVER command deletes the existing server server_name. A server may only be deleted if it is not running at the time.

▶ LIST_SERVER { server_name
-ALL }

Abbreviation: LS

The LIST_SERVER command displays the characteristics of the server server_name at the user terminal.

If -ALL is specified instead of the server_name, the characteristics of all servers are displayed.

▶ MODIFY_SERVER server_name

Abbreviation: MS

The MODIFY_SERVER command modifies an already existing server server_name. This command prompts the user for subcommands to modify the attributes of the server. See Server Subcommands, below. If the server does not already exist, an error message is displayed.

A server may not be modified while it is running.

Server Subcommands

The ADD_QUEUE and MODIFY_QUEUE commands prompt the user for subcommands to configure the server.

The general server subcommand format is:

```
server: subcommand
```

where server: is a system prompt.

The server subcommands are:

FILE	File the server configuration.
LIST_SERVER	List the server configuration.
LOG	Set the server log file.
MESSAGE_LEVEL	Set the server log message level.
PASSWORD	Define the server password.
PRIORITY	Set the server process priority level.
PORT	Set the server port number.
PROGRAM	Define the server program name.
QUEUE	Define the queue to be serviced.
QUIT	Quit the server configuration.
TIMESLICE	Set the server process time slice.

When a server is added, the PORT and QUEUE subcommands must be used, since they have no defaults; use of the LOG subcommand is recommended. The other subcommands can be omitted if the defaults are desired.

► FILE

(No abbreviation)

The FILE command files the server configuration and returns the user to FTGEN command level. If the user does not wish to file the configuration she may issue the QUIT command instead of FILE.

► LIST_SERVER

Abbreviations: LS, L

The LIST command lists the current server configuration. When adding a new server, it is a simple way of displaying the configuration parameters that need to be set and their initial defaults. At any other time it allows the user to check that the configuration is correct before filing.

► LOG { filename }
 { -OFF }

(No abbreviation)

The LOG command specifies the name filename of the log file for the server process being configured. All significant server events, including all transfers (local and remote) which the server has processed, are logged in this file under control of the MESSAGE_LEVEL command. The log file resides in the FTSQ* directory. It is recommended that all servers have log files assigned. These files are useful for analyzing performance, identifying problems with transfers to particular sites, and identifying users who are having difficulties.

A single subsystem-wide log may be specified, or a separate log file may be defined for each server.

If -OFF is specified instead of the filename, no logging is performed for this server.

The default is -OFF.

► MESSAGE_LEVEL message_level

Abbreviation: MSGL

The MESSAGE_LEVEL command specifies the level of information that is entered in the server log file as defined by the LOG command.

There are four levels of detail that can be specified using message_level, as follows:

<u>Message Level</u>	<u>Abbreviation</u>	<u>Function</u>
NORMAL	NRM	Only enter minimum details in the log. This is the default level.
DETAILED	DET	Log all events.
STATISTICS	STAT	DETAILED and STATISTICS information logged.
TRACE	TRC	TRACE, DETAILED, and STATISTICS information logged.

The default level is NORMAL.

► PASSWORD password

(No abbreviation)

The PASSWORD command specifies the password (password) to be associated with the file transfer server that is being configured. A password increases the security of the file system within which the server is operating by keeping out unauthorised file transfer requests. Password protection is recommended for servers accessible via public data networks.

Remote file transfer requests that are addressed to the server must quote this password correctly for the request to be honored.

Password checking is case-sensitive. Uppercase and lowercase letters are distinguished. For example, 'ABcd' and 'abcd' do not match.

The password may contain only characters from this set:

a-z A-Z 0-9 # \$ _ * &

It must not begin with a numeral, and can have a maximum of 32 characters. The password must be used by every site communicating with the server. See the ADDRESS subcommand of the ADD_SITE and MODIFY_SITE commands, below.

The default is to have no password.

► PRIORITY priority_level

Abbreviation: PR

The PRIORITY command selects a PRIMOS process priority level priority_level in the range of 0 to 3 for the server being configured. When the server is subsequently started up, that priority will be automatically assigned to the server.

The default value is 1, which is the value for a normal user process.

► PORT port_number

(No abbreviation)

The PORT command specifies the PRIMENET port port_number on which the current server process should "listen" for incoming calls passed off by the Transport Service Manager. Port_number must be in the range 1 to 99. If there are several servers, each must have a unique port number.

Note

The port_number must not be used by any other PRIMENET application running on the computer. The System Administrator must ensure the uniqueness of port numbers, since FTGEN cannot check for it.

►PROGRAM filename

(No abbreviation)

The PROGRAM command specifies the name filename of the run file for the server being configured. The file must reside in the directory FTSQ*. When the server is started up, this program is phantomd.

The full name of the program must be given, including any suffix. The default program is the Prime-supplied server FTP.SEG.

►QUEUE queue_name

(No abbreviation)

The QUEUE command defines the single queue queue_name that is to be serviced by the server being configured. The queue should be configured by the FTGEN ADD_QUEUE command and appropriate queue subcommands. Each queue is serviced by only one server.

There is no default queue; a queue must be defined.

►QUIT

Abbreviation: Q

The QUIT command quits the current server configuration and returns the user to the FTGEN command level. If the server is being added, no record of it will be filed. If the server is being modified, the old configuration will remain unchanged.

►TIMESLICE time_slice

Abbreviation: TS

The TIMESLICE command sets the timeslice time_slice for the server process being configured.

The value of time_slice is in deciseconds (tenths of a second).

When the server is subsequently started up, the timeslice that has been set will be automatically assigned to the server process.

The default value is 20 (that is, 2 seconds).

SITE COMMANDS

The site commands are for configuring the well-known sites between which file transfers are made.

ADD_SITE Add a new site.
 DELETE_SITE Delete a site.
 LIST_SITE List a site configuration.
 MODIFY_SITE Modify a site.

▶ ADD_SITE site_name

Abbreviation: ASITE

The ADD_SITE command adds a new site site_name to the FTS configuration. This command prompts the user for subcommands to define the attributes of the site. See Site Subcommands, below.

The site_name must conform to FTS site name conventions. If the site already exists an error message is displayed.

▶ DELETE_SITE site_name

Abbreviation: DSITE

The DELETE_SITE command deletes the existing site site_name.

▶ LIST_SITE { site_name }
 -ALL }

Abbreviation: LSITE

The LIST_SITE command displays the characteristics of the site site_name.

If -ALL is specified instead of the site_name, the characteristics of all sites are displayed.

►MODIFY_SITE site_name

Abbreviation: MSITE

The MODIFY_SITE command modifies an already existing site site_name. This command prompts the user for subcommands to modify the attributes of the site. See Site Subcommands, below. If the site does not already exist, an error message is displayed.

Site Subcommands

The ADD_SITE and MODIFY_SITE commands prompt the user for subcommands to configure the site.

The general site subcommand format is:

site: subcommand

where site: is a system prompt.

The site subcommands are:

ADDRESS	Define the address of the current site.
FILE	File the site configuration.
LIST_SITE	List the site configuration.
LOG	Set the site log file.
MESSAGE_LEVEL	Set the site log message level.
QUEUE	Set the site default request queue.
QUIT	Quit the site configuration.

When a site is added, the ADDRESS and QUEUE subcommands must be used.

► ADDRESS address

Abbreviation: ADDR

The ADDRESS command specifies the address of the current site. The address parameter must be in one of the following formats:

sitename+servername

sitename+servername(password)

number+servername

number+servername(password)

The sitename is the name of a configured (by NETCFG) site. The number is the "open network address" of a non-configured site. The servername is the name of the remote FTS server. The password is the password (if any) of the remote server.

Two examples of addresses are:

aarhus+ftp(ikke)

311081800098+ftp5(morn)

There is no default address; a site address must be defined.

► FILE

(No abbreviation)

The FILE command files the site configuration and returns the user to FTGEN command level. If the user does not wish to file the configuration he may issue the QUIT command instead of FILE.

► LIST_SITE

Abbreviations: LS, L

The LIST command lists the current site configuration. When adding a new site, it is a simple way of displaying the configuration parameters that need to be set and their initial defaults. At any other time it allows the user to check that the configuration is correct before filing.

```

▶ LOG { filename
      -OFF }

```

(No abbreviation)

The LOG command specifies the name filename of the log file for the current site. All significant site events are logged in this file under control of the MESSAGE_LEVEL command. The log file resides in the FTSQ* directory.

A single subsystem-wide log may be specified, or a separate log file may be defined for each site.

If -OFF is specified instead of the filename, no logging is performed on a site basis.

The default is -OFF.

```

▶ MESSAGE_LEVEL message_level

```

Abbreviation: MSGL

The MESSAGE_LEVEL command specifies the level of information that is entered in the site log file as defined by the LOG command.

There are four levels of detail that can be specified using message_level, as follows:

<u>Message Level</u>	<u>Abbreviation</u>	<u>Function</u>
NORMAL	NRM	Only enter minimum details in the log. This is the default level.
DETAILED	DET	Log all events.
STATISTICS	STAT	DETAILED and STATISTICS information logged.
TRACE	TRC	TRACE, DETAILED, and STATISTICS information logged.

The default level is NORMAL.

▶ **QUEUE** queue_name

(No abbreviation)

The QUEUE command defines the site default request queue queue_name in which requests are held if a user does not specify a queue name in the FTR command line when submitting a request.

There is no default for this command; a queue must be defined.

▶ **QUIT**

Abbreviation: Q

The QUIT command quits the current site configuration and returns the user to the FTGEN command level. If the site is being added, no record of it will be filed. If the site is being modified, the old configuration will remain unchanged.