CVSI, Inc.
4G Crosby Drive
Bedford, MA 01730

RAS Guide for 50 Series™ System Administrators

Revision 23.3

DOC13156-1LA

RAS Guide for 50 Series System Administrators

First Edition

William T. Carbonneau

This manual documents the software operation of the PRIMOS operating system on 50 Series computers and their supporting systems and utilities as implemented at Master Disk Revision Level 23.3 (Rev. 23.3).

Prime Computer, Inc., Prime Park, Framingham, Massachusetts 01701

Printing History

First Edition (DOC13156-1LA) May 1992 for Revision 23.3

Credits

Project Support: Andy Griffith, George Gove, Ron McKenzie, Charles Hulme, Robert Love, Steve Almquist
Editorial: Irene Rubin, Mary Skousgaard
Illustration: Elizabeth Wahle
Production: Judy Gordon

Contents

About This Book

About This Book . . . v

Recommended Reading . . . v

Book Organization . . . v

Prime Documentation Conventions . . . vi

1 The System Recovery Strategy

What Is RAS? . . . 1-1

RAS Software Components . . . 1-1

Why Should I Use System Recovery Features? . . . 1-4
- Minimal File System Recovery . . . 1-4
- File System Recovery Using RAS Features . . . 1-5

Recommendations . . . 1-6

How Do I Use These Features? . . . 1-7

2 Automated System Recovery

Introduction . . . 2-1

How ASR Works . . . 2-1

Maintenance Processor Microcode . . . 2-2

Automated System Recovery . . . 2-2
- Using SYSTEM_RECOVER in Default Mode . . . 2-3
- How Automated System Recovery Works . . . 2-4

Hangs Versus Halts in ASR Mode . . . 2-7

Hangs Versus Halts in Non–ASR Mode . . . 2-9

Using SYSTEM_RECOVER in Non–default Mode . . . 2-11
- Summary . . . 2-12

3 Handling Halts and Hangs

General Procedure for Handling Halts and Hangs . . . 3-1

Identifying Halts and Hangs . . . 3-2
- Hang Symptoms . . . 3-3

Halt Symptoms . . . 3-3
Types of Halts . . . 3-4
Sensor Checks . . . 3-5
Forced Shutdown Halts . . . 3-5
Trapped Halts . . . 3-6
Immediate Halts . . . 3-7
Recovering From Halts and Hangs While Booting . . . 3-8
Recovering From Hangs Under PRIMOS . . . 3-9
Recovering From Halts Under PRIMOS . . . 3-11
Recovering From Forced Shutdown Halts . . . 3-15
Recovering From Trapped Halts and Immediate Halts . . . 3-15
Warm Starts . . . 3-16
Cold Starts . . . 3-18

4 Crash Dump to Disk

Advantages of Crash Dump to Disk . . . 4-1
Creating a Crash Dump Disk . . . 4-2
CDD –INFO . . . 4-3
Activating a Crash Dump Disk . . . 4-6
Performing a Crash Dump to Disk . . . 4-7
Analyzing a Crash Dump to Disk . . . 4-7
Recommendations . . . 4-8

5 Crash Recovery Facilities

Resident Forced Shutdown (RFS) . . . 5-2
Running RFS . . . 5-2
Summary . . . 5-3
What Is FS_RECOVER? . . . 5-4
Effects of a System Crash on Your File System . . . 5-4
What Does FS_RECOVER Do? . . . 5-4
FS_RECOVER Using a Crash Dump . . . 5-5
FS_RECOVER Without a Crash Dump . . . 5-7
Considerations When Using FS_RECOVER . . . 5-7
Installing FS_RECOVER . . . 5-7
FS_RECOVER Installation Tape . . . 5-8
Using FS_RECOVER.INSTALL.CPL . . . 5-8
Changes to Search Rules . . . 5-8
ACL Requirements . . . 5-8
Segment Requirements . . . 5-9
Changes to PRIMOS.COMI . . . 5-9
Allocating Disk Records for Tape Dumps . . . 5-10
Using FS_RECOVER . . . 5-11
Recommended Strategy After a System Crash . . . 5-11

FS_RECOVER Main Menu . . . 5-12
Reading Crash Dump Tapes . . . 5-14
Performing the Recovery Analysis . . . 5-15
Recommendations for Running FIX_DISK . . . 5-17
FS_RECOVER Summary Display . . . 5-20
Automated FIX_DISK . . . 5-20
Administrative Setup for Automated FIX_DISK . . . 5-21
The INIT_RECOVER -AUTO_ANALYSIS Option . . . 5-22
FIX_DISK Manager Phantom . . . 5-22
Disk Manager Subsystem . . . 5-22
Displaying the State of Currently Mounted Disks . . . 5-23

6 Other RAS Features

Introduction . . . 6-1

Disk Mirroring . . . 6-1

Mirroring Requirements . . . 6-2
Performance . . . 6-3
SPIN_DOWN Command . . . 6-4

Robust Partitions . . . 6-4
Understanding The Robust Partition File System . . . 6-5
What Robust Partitions Can Provide . . . 6-5
Restrictions on the Use of Robust Partitions . . . 6-7
Understanding the Concept of Recoverability . . . 6-8
Understanding the -FAST Option of FIX_DISK . . . 6-9

VCP-V Maintenance Processor . . . 6-11
Quick Boot . . . 6-11
BOOTQ Command . . . 6-11
New Switch Settings . . . 6-12
Microdiagnostics . . . 6-13

Index

About This Book

About This Book

Since Rev. 21.0, Prime has had as a working goal to implement features that improve the operational availability of 50 Series™ systems. Prime® collectively calls this strategy *RAS*: Reliability, Availability, and Serviceability. This book presents information on several RAS implemenations, especially in the area of automated system recovery.

Recommended Reading

You are expected to have some familiarity with Prime systems before reading this book. If you are not familiar with the PRIMOS® operating system, you should read the *PRIMOS User's Guide* (DOC4130–5LA), which explains Prime's file management system and provides introductory and tutorial information about essential commands and utilities.

You should also be familiar with the administrative duties associated with Prime systems, outlined in the three volumes of the *System Administrator's Guide*. You should also be familiar with the *DSM User's Guide* and the *Prime Networks Release Notes*. Other recommended reading includes the *Operator's Guide to File System Maintenance* and the *Operator's Guide to System Commands*.

Book Organization

This book contains 6 chapters:

- Chapter 1, The System Recovery Philosophy, is an introduction to the subject and some of the software components that make up its structure.
- Chapter 2, Automated System Recovery, recommends how to set up system recovery so that minimal manual intervention is required.

- Chapter 3, Handling Halts and Hangs, details different types of interruption of system operations and the ways to recover from them.
- Chapter 4, Crash Dump to Disk, outlines the method and general workings of taking a crash dump to disk following a system halt.
- Chapter 5, Crash Recovery Facilities, presents more information about the crash recovery facilities Resident Forced Shutdown (RFS) and FS_RECOVER.
- Chapter 6, Other RAS Features, provides information about robust partitions, disk mirroring, disk spindown, and Quick Boot.

Prime Documentation Conventions

The following conventions are used throughout this document. The examples in the table illustrate the uses of these conventions.

Convention	*Explanation*	*Example*
Uppercase	In command formats, words in uppercase bold indicate the names of commands, options, statements, and keywords. Enter them in either uppercase or lowercase.	**SLIST**
Italic	Variables in command formats, text, or messages are indicated by lowercase italic.	**LOGIN** *user-id*
Abbreviations	If a command or option has an abbreviation, the abbreviation is placed immediately below the full form.	**SET_QUOTA** **SQ**
Brackets	Brackets enclose a list of one or more optional items. Choose none, one, or several of these items.	**LD** [**-BRIEF** / **-SIZE**]
Braces	Braces enclose a list of items. Choose one and only one of these items.	**CLOSE** { *filename* / **-ALL** }
Braces within brackets	Braces within brackets enclose a list of items. Choose either none or only one of these items; do not choose more than one.	**BIND** [{ *pathname* / *options* }]
Monospace	Identifies system output, prompts, messages, and examples.	`address connected`

Convention	*Explanation*	*Example*
Underscore	In examples, user input is underscored but system prompts and output are not.	`OK, RESUME MY_PROG`
Hyphen	Wherever a hyphen appears as the first character of an option, it is a required part of that option.	**SPOOL -LIST**
Ellipsis	An ellipsis indicates that you have the option of entering several items of the same kind on the command line.	*pdev-1* [. . .*pdev-n*]
Bullet	In a list of options, a bullet indicates the default choice, if one exists. If you do not select an option, the system chooses the default option.	{ **A** • / **O** / **D** }
Subscript	A subscript after a number indicates that the number is not in base 10. For example, the subscript 8 is used for octal numbers.	200_8
Vertical bars	Vertical bars enclose a list of items. Choose one or more of these items.	**OUTPUT** ‖ *filename* / *options* ‖
Parentheses	Parentheses in command or statement formats are a required part of that format. Enter them as shown.	**DIM** *array* (*row, col*)

The System Recovery Strategy

1

What Is RAS?

One of Prime's major goals over the past few years has been to provide inherently reliable computer systems that are also easy to service and maintain. Prime uses the term **RAS** to describe this goal: Reliability, Availability, and Servicability. This means not only providing systems with greater uptime, but also having those systems experience minimal downtime in the event of a halt or a hang condition. This concept of RAS covers both hardware and software.

Prime has been introducing various system recovery features since Rev. 23.0. This document covers these features, and brings together information from previous revisions covered in other documents into a single document.

The RAS strategy states that Mean Time To Recover (MTTR) should be reduced as much as possible, that the System Administrator should have as much flexibility as possible in determining when disks should be fixed, and that a site should be able to run with clean disks much more often because the time and effort involved in identifying and fixing problems is greatly reduced.

RAS Software Components

The software features that make up the components of the RAS strategy are

- SYSTEM_RECOVER command
- Crash Dump to Disk (CDD)
- Resident Forced Shutdown (RFS)
- The FS_RECOVER utility
- The INIT_RECOVER.CPL program
- Quick Boot

These components are briefly defined in the following sections, and are discussed in greater detail later in this document.

SYSTEM_RECOVER

The SYSTEM_RECOVER command specifies five startup parameters

- Auto Recovery
- Crash Dump
- RFS
- System Verify
- Cold Restart

that reside in a special location in memory. These parameters are automatically executed in the event of a system failure. You can employ these parameters to the degree that suits the needs of your particular computer environment, from having minimal operator intervention to having complete manual control over the reboot process.

Crash Dump to Disk

Crash Dump to Disk (CDD) allows you to direct a crash dump to go directly to disk rather than to tape. Before the introduction of CDD, the operator was required to manually intervene in the crash dump. With CDD, no manual intervention is required for the dump itself, and its execution time is usually much faster than tape because the data transfer rate for disk is faster than tape. Also, a CDD image can be analyzed automatically by FS_RECOVER as part of your recovery setup and, if need be, a CDD image can be analyzed by DOC, a diagnostic tool used by PrimeService.

Resident Forced Shutdown (RFS)

Resident Forced Shutdown (RFS) minimizes the number of partitions that *really* require the use of FIX_DISK. RFS attempts to shut down local disk partitions after a halt. RFS shuts down the partitions properly, and identifies the specific disk or disks that really do require the use of FIX_DISK.

PRIMOS buffers up to 8192 disk records in memory to avoid access delays each time a disk record is handled. Records are written back to the disks on a *timed* basis, rather than as each operation is completed. This manner of I/O handling greatly increases performance, but if the system were to halt or hang in a manner that prevented these buffers from being written back to the disks, the file system structure could become corrupted.

Focusing on the file system, systems halt in one of two ways:

- A **fast shutdown**, in which all of the locate buffers are successfully flushed to disk and file system integrity is maintained. You need not run FIX_DISK in this case.

- A halt that prevents a flush of the locate buffers. In this case, PRIMOS marks all partitions as requiring FIX_DISK.

RFS addresses this second halt instance. RFS is a special routine that is guaranteed to be in memory after the system halts, and performs certain file system services while PRIMOS is not running. For example, RFS checks partitions for transactions that modify the file system structures, such as file extend, file create, and file delete. Partitions that do not have such a transaction in progress will be marked as clean, and the file system cache (locate buffers) will be flushed. RFS maintains file system integrity following a halt or hang in approximately 95% of such incidents.

FS_RECOVER

The FS_RECOVER utility is an Independent Product Release (IPR) that allows you to reduce recovery time after a crash, and to get a detailed analysis of the state of the disk partitions. FS_RECOVER performs the following tasks:

- Assesses the state of the file system. It determines which disks are not clean, which disks are clean, and which disks are not clean but can have a deferred FIX_DISK. (The term **clean partition** refers to a partition that does not generate a warning message at the time it is mounted.)
- Attempts to identify the file system objects damaged by the crash.
- Performs a crash dump analysis following reboot that identifies the type of crash, the file system activity at the time of the crash, and any file system corruption that existed prior to the crash.
- Invokes automated FIX_DISK facilities and keeps a COMO record of each one.

FS_RECOVER usually completes its dump analysis within ten minutes. It is also possible to use FS_RECOVER without a crash dump in order to get a general assessment of the file system. You can invoke FS_RECOVER manually, or have it issued automatically by invoking INIT_RECOVER.CPL inside of your PRIMOS.COMI file. FS_RECOVER is available to all customers with a service contract.

INIT_RECOVER.CPL

The INIT_RECOVER.CPL program, part of the FS_RECOVER utility, is invoked from PRIMOS.COMI and allows you to further automate the recovery process by invoking the FS_RECOVER utility. INIT_RECOVER.CPL encaches the PRIMOS maps, enables Automated System Recovery, activates CDD, and reports on the current System Recovery configuration. Also, INIT_RECOVER moves a crash dump from the crash dump partition to a file system partition so that it is available for analysis.

Quick Boot

The Quick Boot processor option allows you to significantly reduce system power-up time by bypassing normal diagnostic checking during system boot.

Why Should I Use System Recovery Features?

Before the introduction of these features, recovering from a system halt could be costly in terms of time spent analyzing the cause of the halt and bringing the sytem back up.

The following short example illustrates the rationale of using these features.

Minimal File System Recovery

Suppose your machine experiences a hang condition. You or the operator would then attempt to halt the machine in order to begin recovery. (Halts and hangs are discussed in greater detail in Chapter 3 of this manual.) At this point, you do not know what state the file system is in. You must assume that there has been some compromise in file system integrity. Although the percentage of file system activity occuring at any one time is relatively small, you cannot be sure that the file system is intact. Suppose you were adding a new record to a file, or a new file was added to a directory; in either case, changes must be made to more than one record in the file system. For example, to add a file, the directory record must be changed to include the new file. The two records are not written out immediately but are put into a temporary holding area called the file system cache, or **locate buffers**. Also, it is not physically possible to write these records out to disk exactly at the same time. If the system halts when only one record has been written out, the file system on the disk has become inconsistent.

At this point, the administrator of a system that contained data whose integrity was paramount would probably take a crash dump on tape, then run FIX_DISK on every partition (except perhaps the COMDEV) without the –FIX option, examine the results, and then run FIX_DISK –FIX on the affected partitions. The time to complete this process is lengthy.

On the other hand, the administrator of a system whose availability is paramount would simply reboot after the halt and run FIX_DISK only if users complained. Or, at the most, the administrator would simply run RFS before booting in order to flush the locate buffers. The administrator in this example is resigned to running with a corrupted file system.

In either of the above cases, the remedy is less than optimum.

File System Recovery Using RAS Features

System recovery features allow you to recover from a system interruption quickly, and also run more cleanly after the halt. If the System Administrator has employed the automated capability of System Recovery to its fullest extent, the following steps are performed without operator intervention:

1. The machine detects a problem and halts. This causes control to be transferred to the Maintenance Processor. The MP looks at a reserved location in memory to find what pre-set actions have been specified by SYSTEM_RECOVER, and executes these actions in the correect sequence.
2. CDD is automatically run. The CDD software takes the crash dump and puts it on disk. CDD is not only usually faster and easier than a crash dump to tape, but a dump generated by CDD can be analyzed by FS_RECOVER, and also by PrimeService (if need be) using the Diagnostic ToolBox (DTB).
3. RFS is automatically run. Before the introduction of RFS, all partitions were marked as not having been properly shut down after a system halt. This was due to the fact that the system could not determine which disks had been in the process of being written to; therefore, file system integrity could not be verified.

   RFS achieves an orderly system shutdown by flushing the locate buffers in order to write the disk records maintained in memory back to the disk (this action is equivalent to that of the SHUTDN ALL command). RFS also determines which disks had actually experienced interrupted file operations, and which ones had been flushed successfully. This greatly minimizes the number of partitions needing a FIX_DISK operation. Also, remember that RFS runs relatively quickly, so you earn tremendous gains in the time saved by not having to run FIX_DISK.
4. At this point, the Maintenance Processor cold starts the system.
5. If you have configured PRIMOS.COMI correctly, it shares most products as phantom processes so that shares can be done *in parallel with* the rest of the PRIMOS.COMI operation. The disks are automatically added.
6. Now the FS_RECOVER utility is initiated by the invocation of INIT_RECOVER.CPL in PRIMOS.COMI. FS_RECOVER moves the crash dump to the file system so the crash dump partition can be reused in the event of another system crash. FS_RECOVER then determines which disks have to be fixed, and provides an automated interface to run FIX_DISK. Fix the disk or disks that need immediate fixing and, if you wish, defer fixing the other disks that are not damaged as badly until a more convenient time. Control returns to INIT_RECOVER.CPL.
7. INIT_RECOVER.CPL invokes the SYSTEM_RECOVER command in order to reset the ASR values in memory that were cleared at boot time. (SYSTEM_RECOVER is discussed in Chapter 2.)

8. CDD moves the crash dump to the file system and activates the CDD partition.
9. PRIMOS.COMI initializes DSM, issues MAXUSR, and finishes the boot.

Recommendations

As you can see, the recovery process has been largely automated and takes much less time performing this process manually. The crash dump is simpler and faster, fewer disk partitions have to be repaired before startup, and coldstart time is quicker. Therefore,

Use the tools.

If it is at all possible, set up the full implementation of system recovery, including the INIT_RECOVER.CPL tool. Prime has designed its recovery tools to work together and, although you can use them individually, their operations are much more efficient when used together.

Always take a crash dump.

If you do not take a crash dump following a system halt, you cannot use FS_RECOVER and therefore ensure that the condition that caused the halt will not recur.

Use CDD.

Try to use CDD rather than crash dump to tape unless you have non-intelligent controllers that cannot use CDD. CDD is usually much faster than CDT, and it does not require operators to mount and change tapes. The space used for CDD is relatively small.

Always run RFS.

If there is one RAS tool that you should always employ, it is this one. It reduces the number of disk partitions that require FIX_DISK. It costs almost no elapsed time, and provides invaluable benefits in terms of maintaining and restoring file system integrity.

Use FS_RECOVER.

FS_RECOVER makes recommendations for fixing the disks, and usually takes less time to fix per partition.

Fix your corrupted file system.

A major part of the RAS philosophy is to make it as easy as possible to run with clean disks. As soon as you can, run FIX_DISK –FIX on disks which you deferred fixing at the time of the crash. If a halt condition occurs before you run FIX_DISK, RFS and FS_RECOVER are much less effective.

How Do I Use These Features?

You must set up your PRIMOS.COMI file properly in order to employ ASR. The PRIMOS.COMI file can either invoke a CPL file that in turn calls the various recovery components, or it can call the INIT_RECOVER.CPL file, which is the most automated form of system recovery, and is part of FS_RECOVER.

All of these separate RAS tools are quite helpful in expediting system recovery, but how do you maximize their functionality? ASR is the process that brings together these RAS components into a single operational scheme: the idea is to have as much knowledge as possible about the cause of a system crash, and to get the system back up as fast as possible and in the best condition possible *based upon* that knowledge.

The next chapter, Automated System Recovery, documents the setup of these recovery features in order that you may automate the recovery/reboot process as much or as little as you wish.

2 Automated System Recovery

Introduction

This chapter presents backround information about Automated System Recovery and then presents general guidelines for setting up ASR on your system. This chapter is intended to be used as a quick-reference by operators or System Administrators who handle operations duties. If you are already familiar with ASR, you can use this chapter to help you decide the best way to configure it for your system. If you are not familiar with ASR, detailed information on specific components of ASR, including the SYSTEM_RECOVER command itself, is presented in this and subsequent chapters.

How ASR Works

Automated System Recovery uses the SYSTEM_RECOVER command to control the actions of the Maintenance Processor after the system has halted and PRIMOS is no longer running. When PRIMOS halts the machine, the Maintenance Processor executes a special piece of code in memory at location 660. This code inspects a checklist of system recovery actions.

Note The same system recovery actions can be manually initiated by issuing the MP commands SYSCLR and RUN 660 on the supervisor terminal of those machines whose Maintenance Processors do not support Automated System Recovery and cannot initiate recovery after a halt.

The checklist speeds and simplifies the steps recommended to recover a system following a system crash. These operations, used in the order specified below, can be automated using SYSTEM_RECOVER:

- Crash Dump to Disk (CDD)
- Resident Forced Shutdown (RFS)

- System hardware verification
- Cold start

Configure these operations prior to a system crash, and specify whether you want system recovery to be automated or to require operator intervention. These operations are discussed in greater detail below.

Maintenance Processor Microcode

All IX-mode CPUs that are supported at Rev. 23.3 can run Automated System Recovery. The CPUs listed below have enhancements that eliminate the need for operator intervention in the event of a system halt. These CPUs, operating with microcode floppy diskettes at or above the revisions listed below, can be enabled to automatically begin ASR following a halt. With firmware prior to these revisions (as with other CPUs not listed), minimum operator intervention is required. Prime recommends that customers employ the latest revision available for their systems.

CPU	DSK7084	Revision
2850	-950	D
2950	-953	D
4050	-935	E
4150	-928	J
5310	-958	J
5320	-960	J
5330	-962	K
5340	-956	K
5370	-964	C
6150	-940	J
6350	-924	S
6450	-941	E
6550	-927	L
6650	-943	E

Automated System Recovery

Automated System Recovery (ASR) is a feature that allows your system to automatically initiate and complete all the steps necessary to recover after a system crash without any manual intervention. You can also configure ASR to require a manual start, rather than starting automatically.

Using SYSTEM_RECOVER in Default Mode

Use the SYSTEM_RECOVER command to configure the Maintenance Processor to automatically perform the necessary steps to bring your system back online after a system crash. These steps are

1. Perform a crash dump to disk.
2. Run RFS.
3. Perform a cold start of the system without verifying system hardware.

Note If the cold start fails, the system performs the hardware verification.

Use the SYSTEM_RECOVER command with no options to configure ASR in the above manner. In order to configure your system for ASR at each cold start, you place appropriate commands in your PRIMOS.COMI startup file. A recommended approach is to

- Write a CPL file to set the recovery parameters.
- Place a command near the end of your PRIMOS.COMI file to run the CPL file.

For example, the end of your PRIMOS.COMI file may look like this:

```
/* Set system recovery parameters
/*
CPL CMDNC0>SYS_RECOVERY.CPL
CO -END
```

The SYS_RECOVERY.CPL file may look like this:

```
/* SYS_RECOVERY.CPL       Friday, 29 November 1991
/*
/* Set system recovery parameters
/*
&SEVERITY &ERROR &IGNORE
COMO BOOT*>SYS_RECOVERY.COMO      /* Start a COMO file
TYPE
DATE                              /* Get time/date
TYPE
&DEBUG &ECHO
STATUS SYSTEM                     /* Get system info
DISKS 111161                      /* Put crash disk in Assignable Disks
Table
CDD 111161 -RD SYSTEM_DUMPS -AD   /* Recover dump; reactivate crash  disk
CDD -QD                           /* Get the current status of crash disk
SYSTEM_RECOVER                    /* Set default recovery parameters
SYSTEM_RECOVER -RC                /* Get the recovery configuration
COMO -END
MAIL BOOT*>SYS_RECOVERY.COMO HAROLD@TPUB.2
&RETURN                           /* Send COMO to System Administrator
```

After these commands run at cold start, your system is ready for automated system recovery. If your system crashes, the Maintenance Processor automatically initiates recovery.

Restrictions: The MP does not automatically start ASR in these cases:

- If you do not configure ASR to be automatic.
- If you are using a CPU or microcode that does not have enhancements for ASR (see the section Maintenance Processor Microcode earlier in this chapter).
- If the halt is due to an environmental condition detected by the MP, such as a power failure, an over temperature, or insufficient airflow.
- If you manually halt the system such as after a hang by using the MP commands STOP or HALT, even if you configure it to be automatic.

If the MP does not initiate SYSTEM_RECOVER automatically, you can initiate recovery manually by entering the following commands at the supervisor terminal in Command Processor (CP) mode:

```
CP1> SYSCLR
CP1> RUN 660
```

You can also manually initiate any of the ASR functions.

How Automated System Recovery Works

Suppose the CPU executes a halt. If you have ASR enabled, the MP begins executing its automated restart code and prints the message

```
DPM402: Beginning auto restart operation.
```

After the DPM402 message is printed, the MP reads an Auto Recovery Restart Address from main memory and then replaces it with zero. If the recovery address read from memory is not zero, the MP will SYSCLR the CPU and start it executing at the recovery address. ASR remains enabled. The operations of this recovery code are defined by the SYSTEM_RECOVER options (listed at the end of this chapter), and may include performing a crash dump, performing a memory dump, and initializing RFS.

After this recovery code has been run, a halt is executed. At this point, ASR is still enabled, and the MP re-enters its auto restart code. The following example illustrates this process: it shows an unexpected halt, the automatic recovery actions (crash dump to disk, memory dump, and RFS) specified by SYSTEM_RECOVER, and the subsequent halt to reboot the system:

```
DPM400: Primary CPU halted at 000006/014263: 045420
         02 Apr 92 18:42:07 Thursday

DPM401: Secondary CPU halted at 000053/033711: 140610
         02 Apr 92 18:42:10 Thursday

DPM402: Beginning auto restart operation.
         02 Apr 92 18:42:17
DPM006: Central Processor System initialization completed.
         02 Apr 92 18:42:18 Thursday

Initializing dump disk 121060 .... OK
Beginning partial dump ......
CORE dump done      6271 records written, 18536 left on disk
MAPS dump done        42 records written, 18494 left on disk
PIOS dump done        65 records written, 18429 left on disk
Crash dump to disk 121060 completed.


*** From RFS: Forced shutdown started!
         Shutting down partition    2060 ... OK
         Shutting down partition    3062 ... OK
         Shutting down partition    3560 ... OK
         Shutting down partition    2266 ... OK
         Shutting down partition    6260 ... OK
         Shutting down partition    2264 ... run FIX_DISK
         Shutting down partition   41666 ... OK
                                 .
                                 .
                                 .
```

If the Auto Recovery Restart Address the MP reads from main memory is zero, a software cold start condition (specified by SYSTEM_RECOVER -COLD_RESTART) is tested. If -COLD_RESTART has not been set, auto restart is disabled and the MP will enter Control Panel mode and the following message is printed on the supervisor terminal:

```
DPM404: Unable to restart.  Entering Control Panel mode.
```

If -COLD_RESTART (the default) is enabled, a number of other operations are possible before the CPU is booted. A software condition may direct the MP to put a dual CPU system into degraded mode. If the system had been in dual mode, the following message is printed:

```
DPM403: Changing to degraded mode for auto restart.
```

If this is not possible because the system was already in degraded mode on the other CPU, an error message is printed:

```
ERR911: Error attempting to reconfigure for auto restart.
```

This message is followed by the DPM403 message and the MP will enter Control Panel mode.

Another SYSTEM_RECOVER option, -SYSV, can direct the MP to load and run sysverify microdiagnostics. After successful completion of these microdiagnostics the functional microcode and decode net are reloaded.

After these conditions have been tested, and after their operations have performed sucessfully, the MP completes the cold start operation by loading the default boot code into main memory and starting the CPU with the sense switch and data switch settings that were used in the previous boot. The example above is continued below to ilustrate.

```
                          .
                          .
                          .
          Shutting down partition   63022 ... OK
*** From RFS: Shutdown completed.

DPM400: Primary CPU halted at 000014/035651: 003403
          02 Apr 92 18:43:37 Thursday

DPM401: Secondary CPU halted at 040000: 160660
          02 Apr 92 18:43:40 Thursday

DPM402: Beginning auto restart operation.
          02 Apr 92 18:43:51
DPM006: Central Processor System initialization completed.
          02 Apr 92 18:43:53 Thursday
DPM007:  System booting, please wait.
[CPBOOT Rev. 19.0 Copyright (c) 1990, Prime Computer, Inc.
[BOOT Rev. 23.3 Copyright (c) 1991, Prime Computer, Inc.

BOOTING FROM 002060 PRIRUN>PRIMOS.SAVE

Verifying memory...
Coldstarting PRIMOS, Please wait...
```

At this point, the default option -AUTO of the SYSTEM_RECOVER command causes PRIMOS.COMI to be automatically inititated.

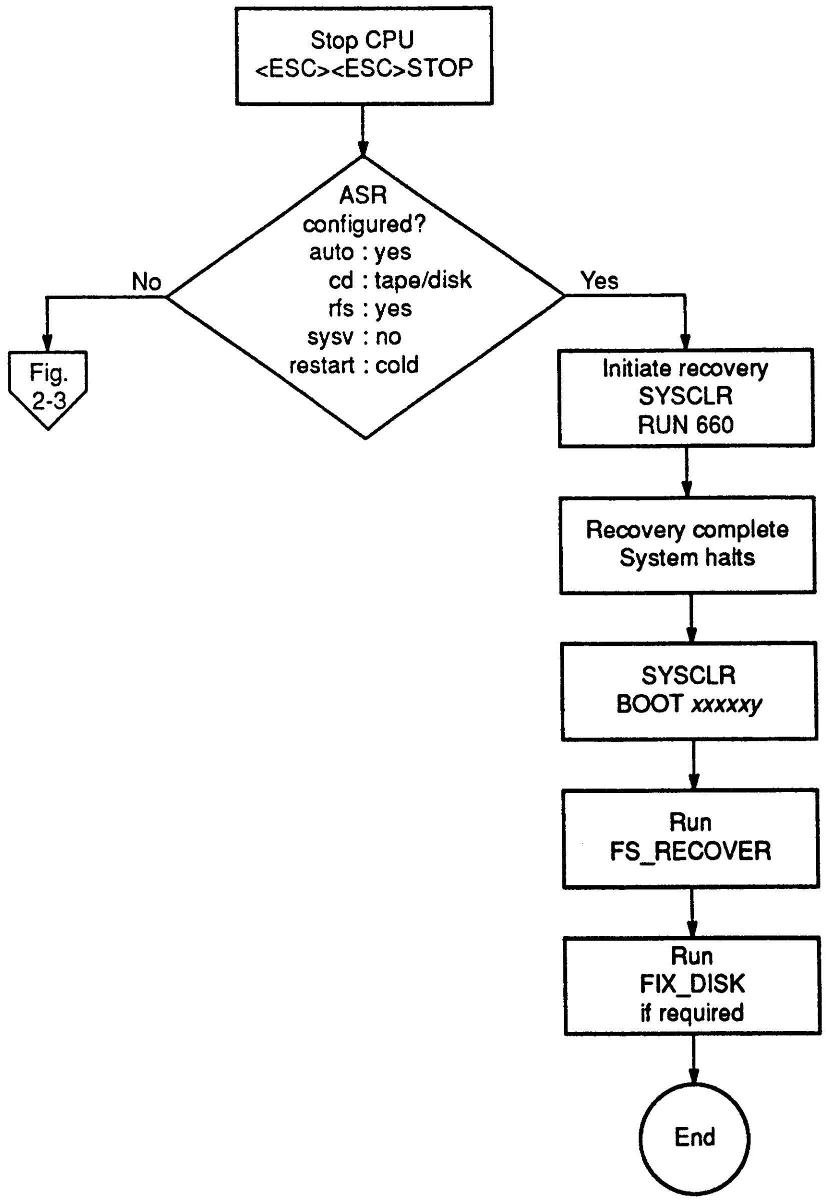
Be aware that ASR is *disabled* on a cold start. Halts during the cold start will put the MP in Control Panel mode unless and until ASR is enabled again by the operating system.

Note ASR is automatically disabled by the MP upon encountering environmental checks, power failures, the soft shutdown, or the STOP command. Issuing a RUN command following a STOP command will *not* re-enable ASR. In this case, ASR remains disabled until it is enabled again by software.

Hangs Versus Halts in ASR Mode

Figure 2-1 presents the steps you should follow when the system hangs and Figure 2-2 shows the steps when the system halts. If ASR is configured, follow the steps on the right of Figure 2-1and Figure 2-2. If automated system recovery is not configured, follow the steps in Figure 2-3. Hangs and halts are discussed in greater detail in Chapter 3. In the case of a hang (Figure 2-1), if ASR is configured to be automatic, follow these steps:

1. Enter the MP commands SYSCLR and RUN 660 to initiate recovery.
2. When the system comes up, run FS_RECOVER.
3. Follow the recommendations of FS_RECOVER to run FIX_DISK.



I02.01.D13156.1LA

Figure 2-1. Hangs and Automated System Recovery

In the case of a halt (Figure 2-2), if ASR is configured to be automatic, the MP initiates recovery. You only need to perform FS_RECOVER and run FIX_DISK (if necessary) to maintain the integrity of the file system.

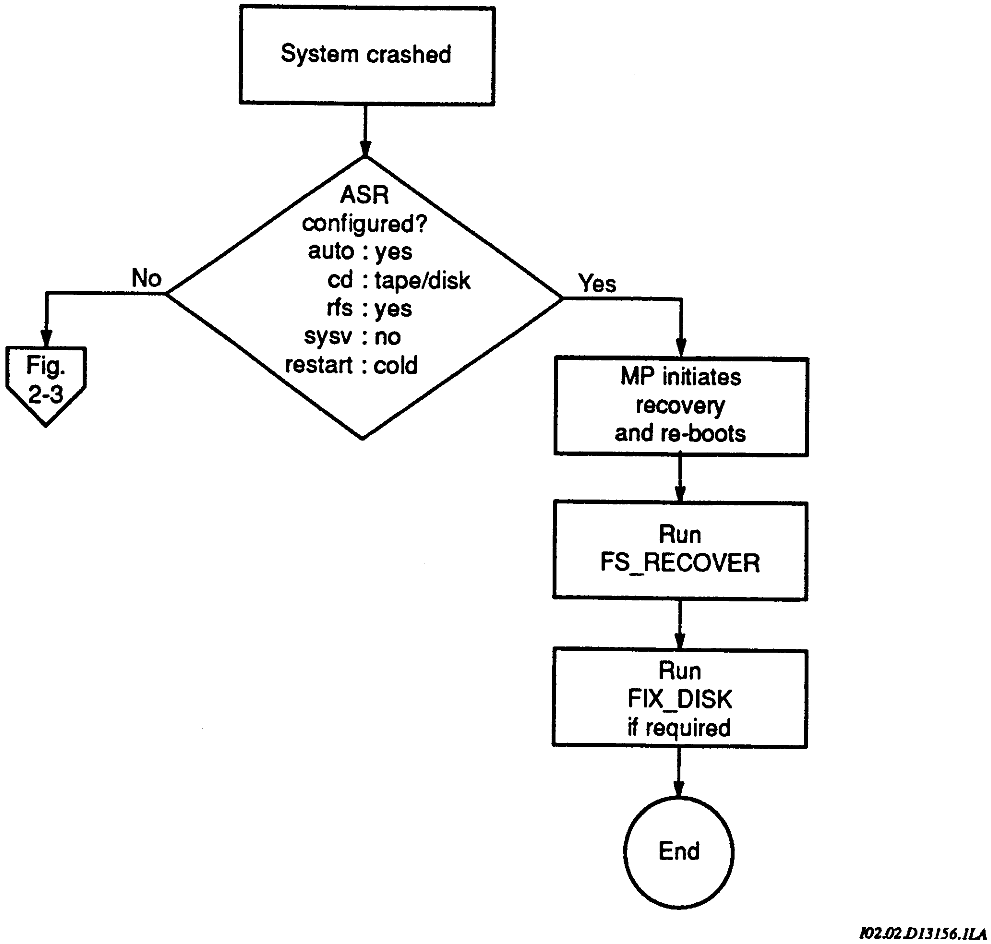


Figure 2-2. Halts and Automated System Recovery

Hangs Versus Halts in Non–ASR Mode

If you do not configure ASR for your system, follow these steps (Figure 2-3):

1. If you created and activated a crash dump disk, initiate a crash dump to disk by entering the MP commands SYSCLR and RUN 661.

   If you did not activate a crash dump disk, initiate a crash dump to tape by entering the MP commands SYSCLR and RUN 774.

2. Run RFS by entering the MP commands SYSCLR and RUN 662.
3. Boot the system by entering the MP commands SYSCLR and BOOT with the appropriate switches.
4. When the system comes up, use CDD to recover the crash dump. (FS_RECOVER can do this if you do not, even if the dump is to tape.)

5. Run FS_RECOVER.
6. Follow the recommendations of FS_RECOVER to run FIX_DISK.

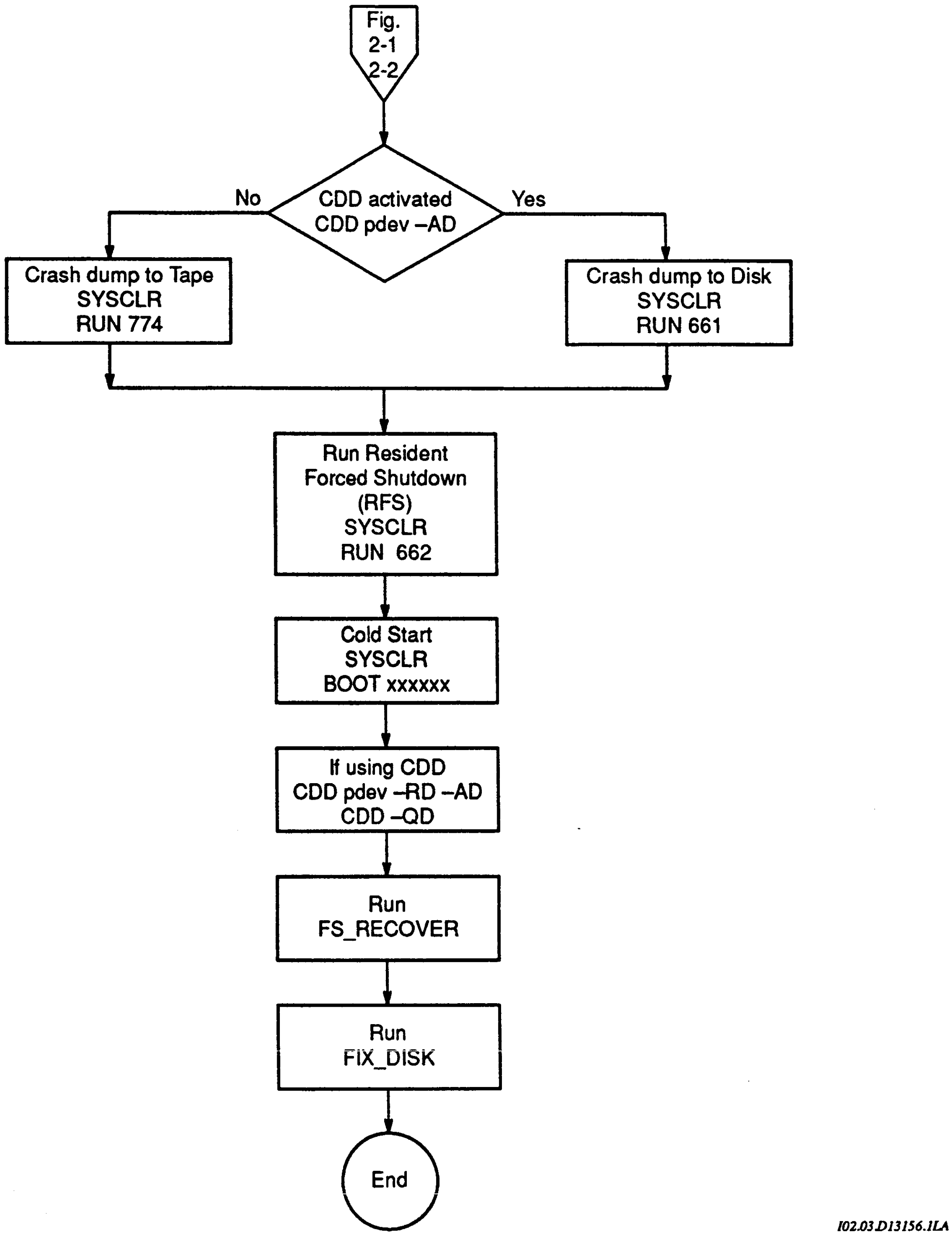


Figure 2-3. Halts and Automated System Recovery Not Activated

Using SYSTEM_RECOVER in Non–default Mode

If you want to configure your system for ASR in a different manner, you can use the SYSTEM_RECOVER options. The easiest way to change configuration is to use the SYSTEM_RECOVER command with no options, thus setting the default configuration. Follow that command with another SYSTEM_RECOVER command and the appropriate option to change the configuration. For example, if you want to configure crash dump to tape, use the –CDT option:

```
SYSTEM_RECOVER
SYSTEM_RECOVER -CDT
```

The options to the SYSTEM_RECOVER command and their meanings are listed below.

–AUTO [*delay*] Configure automated system recovery. *delay* causes a delay time in minutes between the time you issue the SYSTEM_RECOVER –AUTO command and the time when it takes effect. The default for delay is zero minutes. –AUTO is a default option.

–NO_AUTO ASR is not configured such that the MP automatically starts recovery. You initiate recovery manually by using the MP commands SYSCLR and RUN 660, and when recovery is completed, SYSCLR and BOOT *xxxxxy*.

–CDD Configure a crash dump to disk. This is the default.

–CDT Configure a crash dump to tape.

–NO_CD Do not perform a crash dump.

–RFS Configure resident forced shutdown (RFS). This is the default.

–NO_RFS Do not perform resident forced shutdown.

–SYSV Perform system hardware verification prior to coldstart.

–NO_SYSV Do not perform system hardware verification prior to cold start. This is the default.

–COLD_RESTART
Perform a cold start. –AUTO must also be used with this option. This is the default.

–NO_RESTART
Do not perform any restart of the system.

–NO Do not use automated system recovery and deconfigure all SYSTEM_RECOVER options. You cannot invoke ASR manually if you use this option.

Summary

You can manually use the SYSTEM_RECOVER command by itself, or you can include it in your PRIMOS.COMI file in order to initiate Automated System Recovery. Prime recommends that you automate your recovery process as much as you can in order to minimize errors due to manual intervention.

Handling Halts and Hangs

3

Certain hardware or software failures may cause PRIMOS (or a boot of PRIMOS) to stop unexpectedly. Depending on its nature, such a failure is called a halt or a hang.

This chapter describes the recovery procedures that you use to handle halts and hangs, including

- How to identify halts and hangs
- How to perform cold starts and warm starts
- How to prepare for partial and full crash dumps
- How to set up for automated system recovery (ASR)

General Procedure for Handling Halts and Hangs

The general procedure for handling halts or hangs is described below. The remaining sections of the chapter describe these steps in detail.

1. Determine whether a halt or a hang has occurred.
2. If a hang occurred, try to halt the CPU so that you can treat the problem as a halt. If a halt occurred, identify the type of halt so that you can choose the correct recovery procedure. The recovery procedure, which requires either a warm start or a cold start, also depends on whether your system is running ROAM-based products (such as DISCOVER™, PRISAM™, or DBMS).
3. Record any information displayed at the supervisor terminal. Use the MP command DSW to display the DSW registers and record that information.
4. Always perform a crash dump; use a partial dump unless otherwise instructed.
5. Run RFS if you plan to cold start. (See Chapter 5 of this manual for more details on RFS.)
6. Perform a cold start or a warm start to restart the system. If you use a warm start and it fails, you must perform a cold start.

7. Run FS_RECOVER and follow its FIX_DISK recommendations in order to ensure the integrity of your file system. (See Chapter 5 for details on FS_RECOVER.) Use the -FAST option of FIX_DISK on robust partitions.
8. Record in the system logbook all the information about the halt or hang (including the time of the event and, if displayed, halt addresses and the contents of CPU registers) and the actions that you took to correct it.

Cold Start or Warm Start?: When deciding whether to use a cold start or a warm start after a hang or a halt, keep in mind the following rules of thumb:

- In general, cold starts preceded by RFS starting at Rev. 23.1 offer the highest probability of not corrupting data or the file system. However, cold starts alone could cause the system to lose data or could damage the file system.
- Warm starts, if successful, preserve the data. However, some situations (for example, forced shutdown halts) do not allow a warm start.

In general, Prime recommends that you take a crash dump, then run RFS and cold start the system. Prime systems, for the most part, now head off problems that would have previously resulted in halts on which a warm start would have been appropriate. In addition, running RFS and cold starting the system protects the PRIMOS file system. However, Prime INFORMATION-based products may, as in the case of a trapped halt (discussed later in this chapter), still benefit from warm starts by preserving the state of the database at the time of the halt.

Note Avoid using the MASTER CLEAR button to stop a system unless all other means have been unsuccessful. A Control-P issued at the supervisor terminal may occasionally unhang a Maintenance Processor. Do not use the MASTER CLEAR button or the MP commands VIRY, SYSCLR, or RUN before all data relevant to the halt, such as the halt address and the contents of the registers, has been recorded.

Identifying Halts and Hangs

If your system suddenly stops, your first task is to determine whether the problem is a halt or a hang. Two easy ways to distinguish halts from hangs are as follows:

- A message preceded by the code DPM400 halt message from the Maintenance Processor is always displayed after halts, but never after hangs.
- The SYSTEM HALTED light on the Status Panel always comes on after halts, but never after hangs.

The next two sections, entitled Hang Symptoms and Halt Symptoms, list the identifying characteristics of hangs and halts.

After you have determined whether the problem is a halt or a hang, refer to the appropriate section of this chapter, as indicated below:

- If the halt or hang occurred while PRIMOS was being booted, go to the Recovering From Halts and Hangs While Booting section.
- If the *hang* occurred while PRIMOS was running, go to the Recovering From Hangs Under PRIMOS section.
- If the *halt* occurred while PRIMOS was running, first determine the type of halt (by referring to the Types of Halts section) and then go to Recovering From Halts Under PRIMOS.

Hang Symptoms

Hangs are identified by these symptoms:

- The SYSTEM HALTED light on the Status Panel is off, which normally indicates that the CPU is running. The system, however, does not respond to commands from user terminals or the supervisor terminal.
- The supervisor terminal may or may not function in CP mode.
- The DPM400 halt message is not displayed at the supervisor terminal, but some Maintenance Processor error messages (with the ERR prefix) may be displayed.

To recover from the hang, go to the section Recovering From Halts and Hangs While Booting or the section Recovering From Hangs Under PRIMOS, depending on when the hang occurred, as explained below.

Halt Symptoms

Halts are identified by one or more of these symptoms:

- The SYSTEM HALTED light on the Status Panel is on, which indicates that the CPU is not running.
- The halt places the supervisor terminal in CP mode, as indicated by the CP1> prompt.
- The DPM400 halt message from the Maintenance Processor is displayed at the supervisor terminal, as in the following example:

```
DPM400: CPU B halted at 000006/004577: 005262
        16 March 92 18:35:17 Monday
```

Depending on the type of halt, you may see additional messages.

- Immediate halts cause the message preceded by the code DPM701 to be displayed (in addition to the DPM400 message) if the Maintenance Processor determines that the halt was caused by a hardware failure.

  This type of halt and its accompanying message is explained under Types of Halts below.

- Forced shutdown halts and trapped halts cause PRIMOS to display appropriate messages (in addition to any Maintenance Processor messages).

  Both of these types of halts and their accompanying messages are explained under Types of Halts below.

After you identify the halt, your next action depends on when the halt occurred:

- If the halt occurred while PRIMOS was being booted, go to the section titled Recovering From Halts and Hangs While Booting.
- If the halt occurred while PRIMOS was running, first identify the type of halt (by reading the next section, Types of Halts) and then go to the section Recovering From Halts Under PRIMOS.

Types of Halts

The PRIMOS halt mechanism is designed so that halts affect the integrity of the file system as little as possible. For recovery purposes, halts can be grouped into four types:

- Sensor checks
- Forced shutdown halts
- Trapped or slow halts
- Immediate halts or machine checks

You can recognize the type of halt by the message displayed by PRIMOS or the Maintenance Processor. Table 3-1 summarizes the halt types and messages. The next four sections describe the halts in detail.

Table 3-1. Types of Halts

Halt Type	*Messages From PRIMOS or Maintenance Processor*

Sensor checks	`ERR076: MP detects high board temperature`
	`ERR401: MP detects insufficient air flow`
	`ERR950: MP detects insufficient air flow`
	`ERR402: MP detects high voltage`

Forced shutdown	`*** From PRIMOS: Forced Shutdown in progress.`
	`*** From PRIMOS: Forced Shutdown!`
	`*** From PRIMOS: Forced Shutdown completed successfully.`

Trapped	`PRIMOS HALTED AT` *xxxxxx/yyyyyy*
Immediate	No PRIMOS message; possible Maintenance Processor message: `DPM701: Machine check.`

Sensor Checks

Halts due to sensor checks are discussed in the section Emergency Shutdowns Caused by Sensor Checks in Chapter 5 of the *Operator's Guide to File System Maintenance*. In general, these types of halts require you to call PrimeService.

Forced Shutdown Halts

Forced shutdown halts usually occur when PRIMOS detects an internal inconsistency in the file system or other data structures. An orderly shutdown normally gives PRIMOS time to perform a graceful shutdown of all disks to ensure that the file system is not compromised any more. The fault may be a software one, but it might also be a hardware problem, in which case the system must shut itself down in order to avoid further damage.

During the forced shutdown, PRIMOS displays a series of three messages to keep you informed of the state of the shutdown procedure:

```
*** From PRIMOS: Forced Shutdown in progress.

*** From PRIMOS: Forced Shutdown!

*** From PRIMOS: Forced Shutdown completed successfully.
```

Three messages are displayed on the supervisor terminal. (The second message is also displayed on all connected user terminals.)

The third message (`*** From PRIMOS: Forced Shutdown completed successfully.`) is especially important because it tells you that PRIMOS successfully completed all the tasks of the shutdown procedure, thus assuring the integrity of the file system. Keep in mind that on a system with many logged-in users, it may take as long as 3 to 5 minutes between the second and third messages, and even as long as 10 minutes in some extreme cases.

Unsuccessful Forced Shutdown Halt: If the third message is not displayed within 10 minutes after the second message, then the forced shutdown halt was unsuccessful. The system will hang or continue to run in an unpredictable state. To recover from an unsuccessful forced shutdown halt, use one of the following two procedures, which are discussed in more detail later in the chapter:

- If the system hangs, treat it as a normal hang, as explained in the section below, Recovering From Hangs Under PRIMOS.
- If the system continues to run, use the SHUTDN ALL command to stop PRIMOS. If this does not work, use the MP command STOP.

WARNING Do not under any circumstances let the system continue to run after an unsuccessful forced shutdown halt.

After you stop the CPU, follow the procedure in the section Recovering From Forced Shutdown Halts, later in this chapter.

Trapped Halts

Trapped halts rarely occur. They are caused by unexpected hardware or software errors in situations where PRIMOS is not able to guarantee that a forced shutdown will succeed. The trapped halt mechanism is less sudden than an immediate halt, and allows time for the completion of any in-progress data transfers between the CPU and the peripheral devices before the CPU is actually stopped. A trapped halt thus avoids file damage due to partially-written records (but not partially-written file structures).

Note A trapped halt is so called because of the way it is implemented in PRIMOS: the CPU executes a special illegal instruction, which is trapped by a special fault handler, which in turn initiates the trapped halt shutdown.

You will know that a trapped halt has occurred because PRIMOS displays a message in the following format at the supervisor terminal:

```
PRIMOS HALTED AT xxxxxx/yyyyyy
```

xxxxxx/*yyyyyy* (where *xxxxxx* is the segment number and *yyyyyy* is the offset) specify the location in memory where PRIMOS actually encountered the halt instruction. Note that this message is displayed only after a trapped halt.

The following example illustrates the PRIMOS and Maintenance Processor messages that result from a trapped halt:

```
PRIMOS HALTED AT 000006/040660

DPM400: CPU B halted at 000006/004577: 005262
        17 March 91 18:35:17 Tuesday

CP1>
```

The DPM400 message indicates a preset location in memory at which the CPU stopped. This preset location is always the same, regardless of the reason for the halt. To find out exactly where PRIMOS halted, check the address given in the `PRIMOS HALTED AT` message.

Immediate Halts

Immediate halts cause PRIMOS to halt suddenly, without performing the full range of halt-handling procedures that help maintain the integrity of the file system. Immediate halts are caused by software errors or by certain kinds of hardware failures (including uncorrectable memory parity errors, known as ECCUs). Even if the system is using the MEMHLT NO configuration directive, an ECCU halt can still occur. Some of these hardware failures may result in machine checks.

Immediate halts do not produce a halt message from PRIMOS. If the immediate halt is caused by a machine check, the following Maintenance Processor message is displayed:

```
DPM701: Machine check.
```

As with every other type of halt, the DPM400 message is displayed. The DPM701 message also lists the contents of CPU registers containing diagnostic status words. These are some of the registers that may be displayed: DSWSTAT, DSWPAR, DSWPAR2, DSWRMA, DSWBCY, and DSWPB. The data in these registers indicate the type of halt. You can also use the MP command DSW to display these registers. You should log the contents of these registers.

Recovering From Halts and Hangs While Booting

If the halt or hang occurs while PRIMOS is being booted, the action you take depends on what stage of the boot process the system is in. You can determine the stage by the messages displayed at the supervisor terminal, as discussed below.

Use this procedure to recover from a hang or a halt while booting PRIMOS:

1. Make sure that the system disks are operational and that the disk drives containing the command and paging partitions are not write-protected.
2. Check the messages on the supervisor terminal:

   - If a Maintenance Processor error message is displayed, refer to the *Operator's Guide to File System Maintenance* for an appropriate response.
   - If no message is displayed, press the ESC key twice or press Control-P. If this fails to return the CP1> prompt, press the MASTER CLEAR button. In either case, enter the BOOTP or BOOTQ commands at the CP1> prompt. If this action does not work, turn the power off and on by pressing the ON/INITIATE SHUTDOWN button twice. PRIMOS should autoboot.
   - If the halt occurred after the DPM007 message displayed (not applicable on VCP-V in Quick Boot mode), first try an autoboot by pressing the ON/INITIATE SHUTDOWN button twice. If this action does not work, the disks or PRIMOS itself (such as the BOOT program) may be corrupt. On the VCP-V in Quick Boot mode, invalid default sense switch settings or data switch settings could cause a hang while booting. Appending the appropriate sense switch settings and data switch settings to the BOOTQ or BOOTP command updates the default settings and may resolve the problem:

     ```
     BOOTQ 14114 0

     BOOTP 14114 0
     ```

     Remember that booting from disk or tape in Quick Boot mode requires a data switch setting of zero.
   - If a message from PRIMOS is displayed, refer to Appendix B in the *Operator's Guide to File System Maintenance* for an appropriate response.

3. If you still cannot boot, make a note of the supervisor terminal messages and call your PrimeService representative.

You can assume that PRIMOS is running successfully when the first OK, prompt appears at the supervisor terminal.

Recovering From Hangs Under PRIMOS

When a hang occurs while PRIMOS is running, your first step is to try to force the CPU to halt so that you can treat the problem as a normal halt, as described in the next section, Recovering From Halts Under PRIMOS.

Note You should first determine if the system is really hung or if it is busy or the supervisor terminal is hung. Check the activity at user terminals or check the disk activity lights. Attempt to log in at or get response from a user terminal.

Use the procedure below to recover from hangs when PRIMOS is running. Figure 3-1 is a flow chart of Steps 1 and 2, and Figure 3-2 details Steps 3 and 4.

1. Enter in the system logbook the time and date of the hang. If the supervisor terminal is not in CP mode, check that the key switch on the Status Panel is unlocked and press the ESC key twice. (If the CP1> prompt does not appear, go to Step 3.)
2. Use the STOP command to halt the CPU:
   - If the STOP command does not work, go to Step 3. (See Figure 3–2.)
   - If the STOP command halts the CPU, go to the section titled Recovering From Halts Under PRIMOS and treat the problem as a halt. (See Figure 3–3.) You know that the CPU halted if the SYSTEM HALTED light is on and the DPM400 halt message is displayed at the supervisor terminal.

```
CP1> STOP
DPM400: CPU B halted at 000006/037515: 013404
        17 March 92 13:43:27 Tuesday
CP1>
```

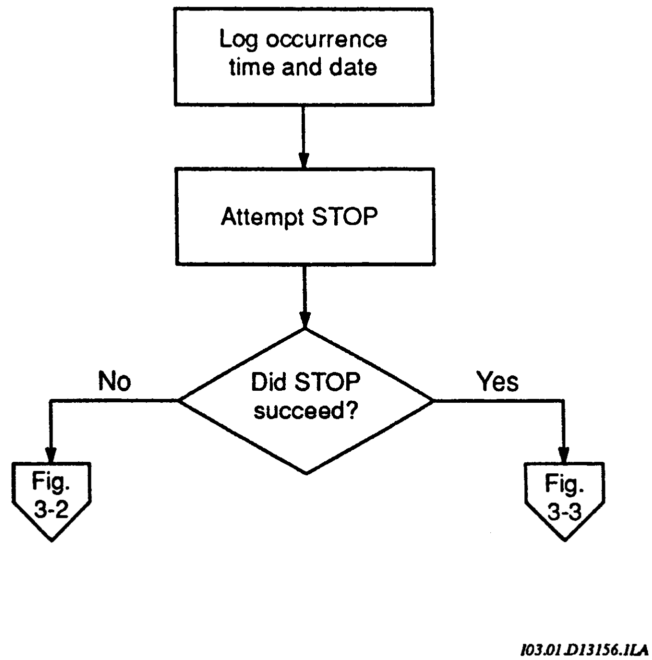


Figure 3-1. Recovering From Hangs (Steps 1 and 2)

3. If the CP1> prompt did not appear in Step 1 or if the STOP command did not work in Step 2, press the MASTER CLEAR button on the Status Panel to initialize the system. (See Figure 3-2.)

   - If the MASTER CLEAR button works, a series of DPM messages will indicate that the MASTER CLEAR was successful. Perform a crash dump and then run RFS. Then cold start the system.
   - If the MASTER CLEAR does not work, press the ON/INITIATE SHUTDOWN button twice to turn the system power off and on. The system should initialize and autoboot PRIMOS. If it does not, contact your PrimeService representative.

4. Record all hang-handling actions you take, and their results, in the system logbook. If PRIMOS booted successfully, run FS_RECOVER and follow the recommendations to run FIX_DISK to ensure the integrity of the file system.

Recovering From Halts Under PRIMOS

To recover from a halt, you must use a cold start or a warm start to get PRIMOS running again. The sections titled Warm Starts and Cold Starts, both later in this chapter, describe each type of restart.

Use the procedure below to recover from a halt incurred when PRIMOS was running. Figure 3-3 is a flow chart of these steps.

1. Examine the halt message to determine which type of halt occurred. (Refer to Types of Halts and Table 3-1, earlier in this chapter.) Record the message in your system logbook, together with the time and date of the halt, values from the DSW registers, and any other information displayed by the Maintenance Processor. To obtain the contents of the DSW registers, enter DSW at the CP1> prompt.
2. Perform a crash dump. Use the MP command SYSCLR, followed by RUN 661 for a crash dump to disk or RUN 774 for a crash dump to tape. A full dump is not necessary and should be done only if you are instructed to do so. The information in the crash dump is necessary to determine the cause of the halt and to be used by FS_RECOVER for analysis of the file system.

   Be sure to perform the crash dump before using any other MP command, because such commands may corrupt the state of the data in memory and make the information saved by a crash dump useless. (See Chapter 4 later in this manual for more information.)

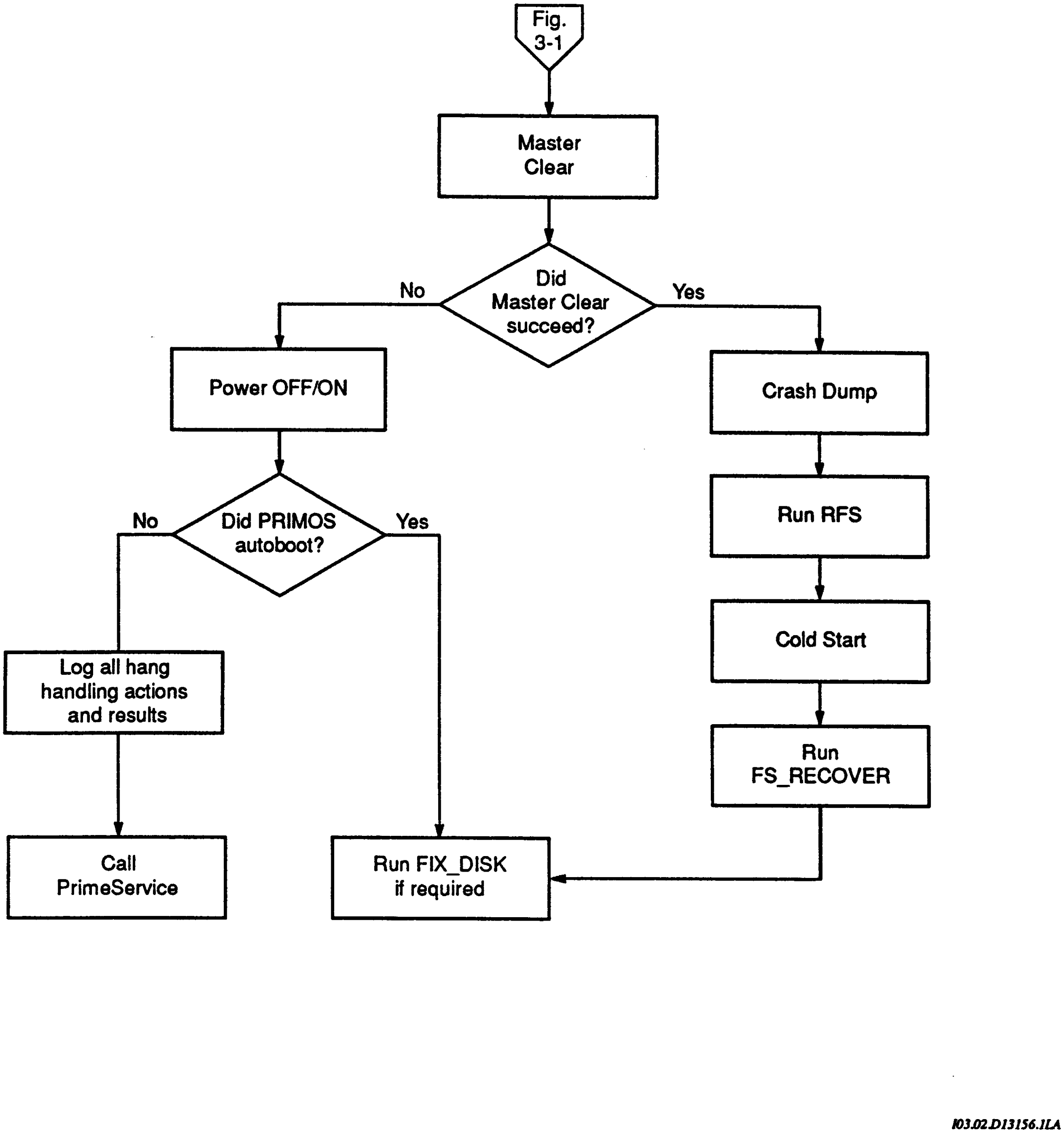


Figure 3-2. Recovering From Hangs (Steps 3 and 4)

3. Use a warm start or a cold start to get the system running again:

   - If your system *is not* running ROAM-based products, use Table 3-2.
   - If your system *is* running ROAM-based products or if the warm start failed, run RFS by issuing the SYSCLR and RUN 662 commands, and then issue the BOOTP or BOOTQ command, or the SYSCLR and BOOT commands.

4. Run FS_RECOVER and follow the recommendations to run FIX_DISK to ensure the integrity of the file system. (The only exception to running FIX_DISK is if a successful shutdown halt occurs and you receive no messages from subsequent ADDISK commands about running FIX_DISK.)
5. Record all your halt-handling actions and their results in the system logbook. This information is helpful to your system analyst or to your PrimeService representative in determining the cause of the halt.

If you cannot restart the system by following the above prescribed procedure, or if halts and hangs recur, call your PrimeService representative.

For systems that do *not* run ROAM-based products, Table 3-2 and Figure 3-3 summarize the recovery procedures for each type of halt. The following four sections contain more details.

Table 3-2.Halt Actions on Non-ROAM System

Message Displayed	*Type of Halt/Corrective Action*

`*** From PRIMOS: Forced Shutdown in progress.`
`*** From PRIMOS: Forced Shutdown!`
`*** From PRIMOS: Forced Shutdown completed successfully.`
	Forced shutdown halt
	1. Crash dump.
	2. Cold start.
	3. Run FS_RECOVER and follow recommendations.

`PRIMOS HALTED AT xxxxxx/yyyyyy`	Trapped halt 1. Crash dump 2. Warm start; if this fails, run RFS and cold start. 3. Run FS_RECOVER and follow recommendations.
No PRIMOS message. Possible Maintenance Processor message: `DPM701: Machine check.`	Immediate halt 1. Crash dump 2. Warm start; if this fails, run RFS and cold start. 3. Run FS_RECOVER and follow recommendations.

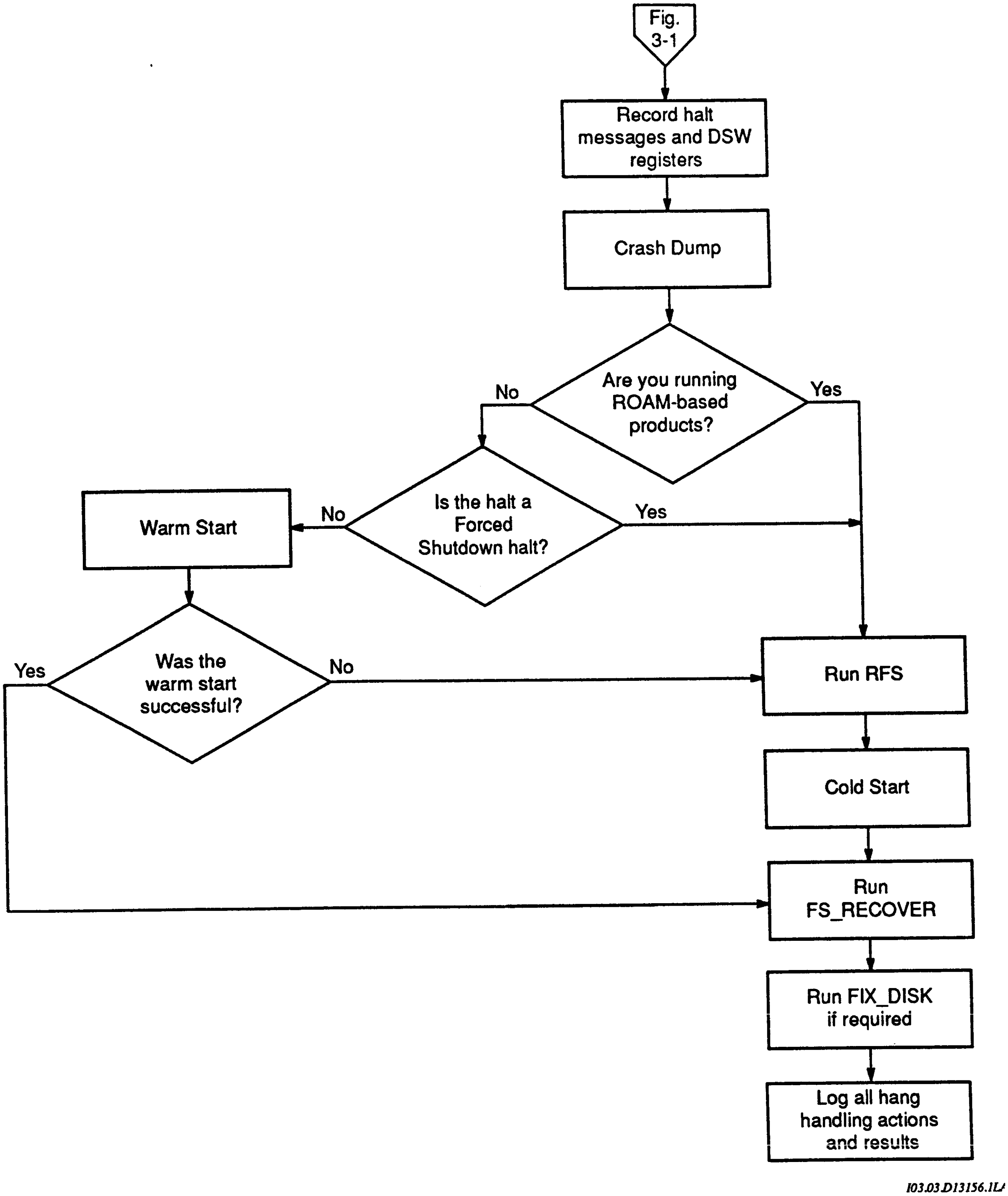


Figure 3-3. Recovering From Halts Under PRIMOS

Recovering From Forced Shutdown Halts

The procedure for recovering from a forced shutdown halt depends on whether PRIMOS successfully performed the forced shutdown. A successful forced shutdown halt is signaled by the third forced shutdown message from PRIMOS:

```
*** From PRIMOS: Forced Shutdown completed successfully.
```

Successful: Use this procedure to recover from a *successful* forced shutdown:

1. Perform a crash dump.
2. Cold start the system, regardless of whether you are running ROAM-based products.
3. After system startup, run FS_RECOVER if you receive the following message from an ADDISK command during the booting procedure:

```
*** Disk "disk" was not shutdown properly, Run FIX_DISK.***
```

   In this case, follow the recommendations of FS_RECOVER to run FIX_DISK.

Unsuccessful: Use this procedure to recover from an *unsuccessful* forced shutdown:

1. Perform a crash dump.
2. Run RFS.
3. Cold start the system.
4. After system startup, run FS_RECOVER and follow the recommendations to run FIX_DISK. Alternatively, run full FIX_DISK on all standard partitions and fast FIX_DISK on all robust partitions.

Recovering From Trapped Halts and Immediate Halts

For trapped halts (also called slow halts), use this recovery procedure if you *are* running ROAM-based products:

1. Perform a crash dump.
2. Run RFS.
3. Cold start the system.
4. After system startup, run FS_RECOVER and follow the recommendations to run FIX_DISK. Alternatively, run full FIX_DISK on all standard partitions and fast FIX_DISK on all robust partitions.

If you *are not* running ROAM-based products,you may attempt to warm start the system; if the warm start fails, follow the above procedure.

Note You cannot use RFS before attempting a warm start.

To help prevent immediate halts that may be caused by ECCU errors, you can use the MEMHLT NO directive in the system configuration file. If MEMHLT NO is configured and the system experiences immediate halts, have the system serviced.

Warm Starts

In general, you may attempt to warm start PRIMOS after these situations:

- Trapped halts (non-ROAM systems only)
- Immediate halts (non-ROAM systems only)

WARNING Do not warm start the system if it is running ROAM-based data management products (such as DISCOVER, PRISAM, or DBMS) or you may lose data. Use a cold start only, so that the ROAM product can perform a rollback of incomplete transactions. (Ask your System Administrator if you are not sure whether ROAM-based products run on your system.)

Use the following procedure to warm start your system. Figure 3-4 is a flow chart of this procedure.

Note You cannot use RFS before attempting a warm start.

---

1. Enter in the system logbook all information displayed at the supervisor terminal and log the values of the DSW registers.
2. Perform a crash dump.
3. Use the WARMSTART command to warm start the system. If the warmstart is successful, PRIMOS is restarted after these messages are displayed:

```
CP1> WARMSTART
DPM006: Central Processor system initialization completed.
        14 May 91 14:05:23 Tuesday
SYSTEM WARM STARTING, PLEASE WAIT

***** WARM START *****
```

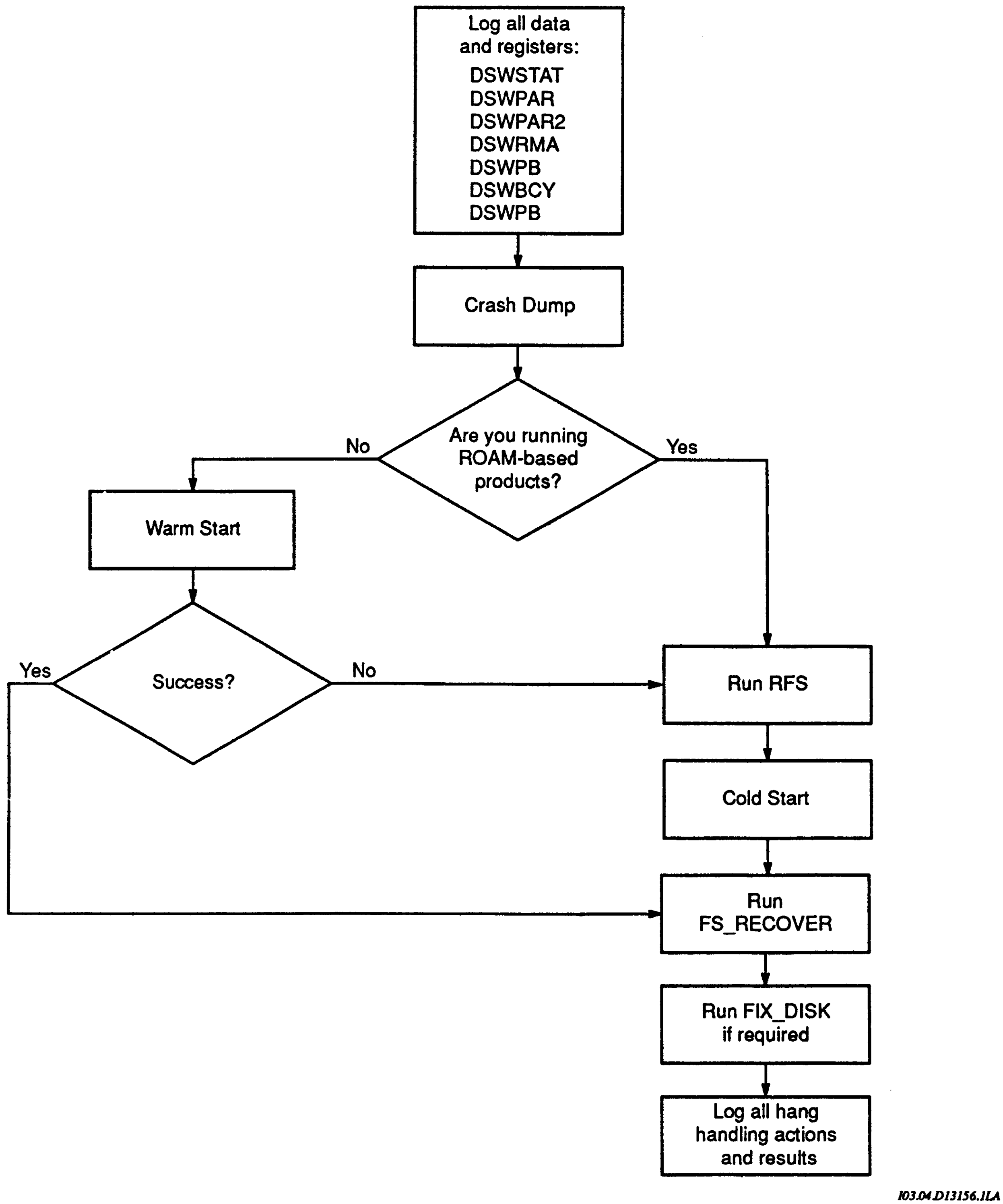


Figure 3-4. Warm Start

A warm start may take about 90 seconds before the WARMSTART message appears at user terminals. (It takes slightly longer for the message to appear at the supervisor terminal.) Do not assume a warm start has failed without waiting at least 90 seconds and checking the user terminals for the WARM START message.

4. If the warm start fails, either no message is displayed or the system halts. In this case, run RFS and then perform a cold start.
5. After the system is running, ensure the integrity of the file system by doing either of the following:
   - o Run FS_RECOVER and follow the recommendations to run FIX_DISK, or
   - o Run full FIX_DISK on standard partitions and fast FIX_DISK on robust partitions.

   Be sure to record all your halt-handling actions and their results in the system logbook.

Cold Starts

In general, cold start PRIMOS after these situations:

- Forced shutdown halts
- Any halt if your system is running database products
- Any time a warm start is unsuccessful
- If you change CPU modes between DUAL and UNI

Use this procedure to cold start your system after a crash:

1. Be sure that you enter in the system logbook all information displayed at the supervisor terminal and log the DSW registers.
2. Perform a crash dump.
3. At this point, you may wish to run RFS from CP mode. (If the system experienced a successful forced shutdown, RFS should not be necessary but you may wish to follow procedure.)

```
CP1> SYSCLR
CP1> RUN 662
```

4. From CP mode, use the BOOTP, BOOTQ, or SYSCLR command, followed by BOOT:

```
CP1> BOOTP
```

5. After the system is running, ensure the integrity of the file system by running FS_RECOVER and following the recommendations to run FIX_DISK, or by noting the RFS messages to run full FIX_DISK on the affected standard partitions and fast FIX_DISK on the affected robust partitions.

   You do not have to run FS_RECOVER after a successful forced shutdown halt, however, unless an ADDISK command displays this message:

```
*** Disk "disk" was not shutdown properly, Run FIX_DISK.***
```

   Note that a robust partition that is improperly shut down cannot be added with the ADDISK command, but instead will produce this message:

```
*** Robust Partition pdev has not been properly shutdown.
*** Fast FIX_DISK has to be run before it can be added.
```

   You must add the robust partition with the –FORCE option, and then run fast FIX_DISK on it as the message states. For details on FIX_DISK and on robust partitions, see Chapter 6 of this guide and the *Operator's Guide to File System Maintenance*.

Caution If you do not heed the message from ADDISK to run FIX_DISK, you run the serious risk of losing data records and files due to file system problems such as unrecoverable disk errors, pointer mismatches, or errors indicated by the message `Directory Damaged.`

---

4 Crash Dump to Disk

A **crash dump** is the writing of the contents of memory to disk or to tape after a system halt. The crash dump preserves a record of the state of the system at the time that the halt occurred. Crash dumps are used by FS_RECOVER in determining which disks need to be fixed. Also, crash dumps are absolutely essential for your PrimeService representative to be able to determine the cause of a halt.

Note A crash dump, which can be performed only from CP mode, must be the first operation performed following a halt after you have recorded the halt information and registers. RUN, BOOT, WARMSTART, or other MP commands cause operations that corrupt the state of the system, thus making the information saved by a subsequent dump less useful. In addition, do not use the MASTER CLEAR button before you have recorded the halt location and determined the recovery actions you will take.

There are two types of crash dumps:

- **Partial crash dumps**, in which the system writes only a part of memory to disk or to tape.
- **Full crash dumps**, in which the system writes the entire contents of memory to disk or to tape. No preparation is required on your part for a full crash dump while PRIMOS is running.

Note Prime recommends that you do a partial crash dump rather than a full dump after a halt because FS_RECOVER and the crash analysis software used by PrimeService need only the partial dump to successfully determine the condition of the file system. Also, a partial dump takes less time and requires less disk space.

Advantages of Crash Dump to Disk

There are three advantages of crash dump to disk over crash dump to tape:

- Crash dump to disk can be performed without operator intervention, because there is no need to mount reels of tape.
- Taking a crash dump to disk is significantly faster than taking a crash dump to tape.

- Both FS_RECOVER and Autopsy, a utility whose use is reserved for PrimeService, can analyze the dump right away, rather than having to wait for a dump from tape.

All of these advantages of crash dump to disk improve system availability by decreasing the time required for collecting crash dump data.

The FS_RECOVER facility can analyze either a crash dump to disk or a crash dump to tape. For further details on crash dump analysis, refer to the *Using FS_RECOVER* manual.

Both the crash dump to disk and the crash dump to tape facilities have been enhanced to write map information as part of the crash dump. Previously, map information was written to the directory SYSTEM_DEBUG*>CRASH>MAPS and had to be separately recovered.

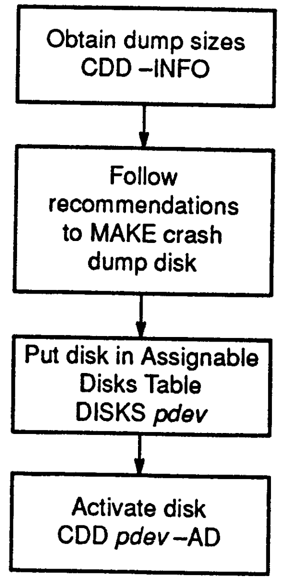
Creating a Crash Dump Disk

Figure 4-1 presents the steps required to create a crash dump disk. Use the CDD command option –INFO (discussed immediately following this section) to determine the disk size necessary for a partial crash dump of your system's memory. Follow the prompts that CDD displays and use the information displayed with the –SPLIT option of MAKE.

A crash disk on a SCSI disk type associated with a Model 7210 (SDTC) disk controller can be created by using only the –SPLIT option; if the disk is on a Model 6580 (IDC1) disk controller, you must also use the –IC option.

At Rev. 23.3, there is no waste of disk space if you use the optimal split value recommended by CDD –INFO; all records not needed by CDD are available to the file system on the other side of the disk. You can add the file system portion of the split partition (using ADDISK) and perform I/O on it without incurring a performance penalty, because file system I/O and crash dump processing do not occur concurrently.

Place the disk in the Assignable Disks Table and activate the disk by using the –ACTIVATE_DISK option of the CDD command.



I04.01.D13156.1LA

Figure 4-1 Creating a Crash Dump Disk

CDD –INFO

In order to determine the record size to allocate for a crash dump disk, use the –INFO option of the CDD command as a planning aid for this task. Use of this option alone gives you the sizes for a full crash dump and for a partial crash dump. (Prime recommends that you use a partial crash dump.) At Rev. 23.3 CDD –INFO provides precise –SPLIT recommendations when you create a crash dump disk.

You can use other options with the –INFO option to specify the disk type you will use for the crash dump and the dump size if you know it. You can also request a table of optimal dump sizes and you can determine the dump sizes for other CPUs and other total memory sizes, for example, for other machines in your network.

Normally you would use the following command format to determine the value to use with the –SPLIT option of MAKE. CDD determines the size of the memory for the system you are on and calculates the required dump sizes:

```
OK, CDD -INFO
[CDD Rev. 23.3 Copyright (c) 1992, Prime Computer, Inc.

-----------------------------------------------------------------------------
| This system has 64 MB of core memory.  Expected total sizes for full and  |
| partial dumps are made up as follows:                                     |
|                                                                           |
|                              FULL DUMP          PARTIAL DUMP              |
|                          ----------------    ----------------             |
|     CORE memory dump :      32767 records       16384 records (approx)   |
|     MAPS dump        :         42 records          42 records            |
|     PIOS dump        :         65 records          65 records            |
|     Safety margin    :        100 records         100 records            |
|                          ----------------    ----------------             |
|     TOTAL DUMP SIZE  :      32974 records       16591 records (approx)   |
|                          ================    ================             |
|                                                                           |
-----------------------------------------------------------------------------

For MAKE recommendations, please specify the disk you intend to use for CDD.
Enter "Q" to quit, or "H" for help.

Enter <pdev> or disk name:
```

For a partial dump, you can now see that you need approximately 16591 records of disk space. Assuming you have a Model 4729 disk, which has 10414 records per surface on the last 27 surfaces, you can dedicate the last three surfaces for the crash dump space (and some file system space) and the remaining surfaces for a file system. (One surface is too small and starting surface numbers must be even so you need three surfaces.) The basic pdev for the last three surfaces (starting surface 28) is 160421. Assuming this disk is on controller 26_8 and drive unit 0, you add 40 for a pdev of 160461. Now specify this information:

```
Enter <pdev> or disk name: 160461

Please specify a MAKE-compatible disk type for disk 160461.
Enter "H" for Help, or "Q" to quit.

Enter disk type (e.g.  "MODEL_4729"): MODEL_4729
-----------------------------------------------------------------------------
| The crash disk you have specified has the following characteristics:     |
|                                                                           |
|     Disk 160461      :     2 heads, starting head 0  (ctlr '26, unit 0)   |
|     Disk model       :     MODEL_4729                                     |
|     Total disk size  :     31242 records                                  |
|                                                                           |
| To MAKE this disk with the maximum possible crash dump capacity:         |
|                                                                           |
|     MAKE disk with   :     -SPLIT 30989          (see note 1 below)       |
|     Maximum dump size:     30988 records         (see note 2 below)       |
|                                                                           |
| ************* This disk is TOO SMALL for a full dump.  *************     
|                                                                           
| For this disk to accommodate a partial dump of the size predicted        |
| earlier, the smallest -SPLIT value you can specify to MAKE is:           |
|                                                                           |
|                              FULL DUMP          PARTIAL DUMP              |
|                          ----------------    ----------------             |
|     MAKE disk with   :    ** TOO SMALL **       -SPLIT 16764              |
|     Maximum dump size:                          (16763 records)           |
|                                                                           |
-----------------------------------------------------------------------------
Type <return> for explanatory notes, or "Q" to quit:
```

You now see that the three surfaces of this disk will accommodate a partial dump (but not a full dump). You then should use MAKE with the –SPLIT option with an argument of 16764. You can use the remaining records on this partition (31242 – 16764 = 14478) for a file system.

If you use only the –INFO option without specifying the pdev or the disk type, CDD prompts you for this additional information in order to recommend the values that MAKE needs to create the crash dump disk.

For example, to determine optimal partial dump size for your system using a Model 4729 disk, you could use this command line:

```
OK, CDD 160461 -DT MODEL_4729 -INFO
```

If you want to depart from the –SPLIT value recommended by CDD –INFO, you should consult a table of optimal dump sizes for your particular system and disk type by using the CDD –DUMP_SIZE_TABLE option (abbreviation –DST). Be sure to use these optimal –SPLIT values. The table appears like this:

```
OK, CDD 160461 -DT MODEL 4729 -DST 14000 1000
[CDD Rev. 23.3 Copyright (c) 1992, Prime Computer, Inc.]
------------------------------------------------------------------------------
-
| The crash disk you have specified has the following characteristics:       |
|                                                                            |
|     Disk 160461       :      3 heads, starting head 28  (ctlr '26, unit 0) |
|     Disk model        :      MODEL_4729                                    |
|     Total disk size   :      31242 records                                 |
|                                                                            |
| To MAKE this disk with the maximum possible crash dump capacity:           |
|                                                                            |
|     MAKE disk with    :      -SPLIT 30989         (see note 1 below)       |
|     Maximum dump size:       30988 records        (see note 2 below)       |
|                                                                            |
| DUMP SIZE TABLE:                                                           |
|                                                                            |
| For this disk, optimal splits are those for which either the maximum dump  |
| size (MDS) or the -SPLIT value (S) is an exact multiple of 254 records,    |
| and S = MDS + 1.  Below is a table of optimal -SPLIT values, beginning     |
| from the dump size closest to 14000 records, and approx 1000 apart:        |
|                                                                            |
|     MAKE with -SPLIT  14224 for a maximum dump size of  14223 records      |
|     MAKE with -SPLIT  15240 for a maximum dump size of  15239 records      |
|     MAKE with -SPLIT  16002 for a maximum dump size of  16001 records      |
|     MAKE with -SPLIT  17018 for a maximum dump size of  17017 records      |
--More--
|     MAKE with -SPLIT  18034 for a maximum dump size of  18033 records      |
                        .                    .                    .
                        .                    .                    .
                        .                    .                    .

|     MAKE with -SPLIT  30226 for a maximum dump size of  30225 records      |
|     MAKE with -SPLIT  30989 for a maximum dump size of  30988 records      |
|     - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -      |
|     End of table - preceding line represents maximum capacity of disk      |
------------------------------------------------------------------------------
Type <return> for explanatory notes, or "Q" to quit: Q
OK,
```

At this point, you can make the crash dump disk based on the recomendations provided by CDD –INFO. For reference material about the CDD command, see the *Operator's Guide to System Commands.*

Activating a Crash Dump Disk

You must activate a crash dump disk before you can use it for crash dump purposes. When you take a crash dump, CDD writes the system crash information into this activated partition. To activate a crash dump disk, perform the following steps:

1. Use the MAKE –SPLIT command to format the disk (only necessary the first time the disk is used).
2. Use the DISKS (or DI) command to add the disk to the Assignable Disks Table.
3. Use the CDD –ACTIVATE_DISK command to activate the crash dump disk. Only one crash dump disk can be activated at a time.

A crash dump disk must be the non-file-system portion of a split partition; it can be a paging partition that is not currently used for paging. The disk must be on a Model 10019 (IDC) or Model 7210 (SDTC) disk controller.

A disk drive in a 75500-6PK device module that contains a crash dump disk cannot be swapped while it is activated. If you wish to perform a disk swap, you can

- Deactivate the crash dump disk
- Activate a crash dump disk on another disk drive
- Issue a SPIN_DOWN or DISK_PAUSE command

If the crash dump disk is a non-SCSI disk, it must have been made with the –DBS ON option of the MAKE command. A SCSI disk on a 7210 controller can be made with either the –IC or –AC option; do not use the –DBS option with a SCSI disk.

Note You cannot activate a partition as a crash dump disk (using CDD –ACTIVATE_DISK) if the partition is currently in use for anything else: paging, assigning, or mirroring a disk. The file system side of the disk, however, is not subject to this restriction, and may be added at the time the disk is activated. Once you have activated a disk for CDD, you cannot use it for anything else because of the initialization information written on it.

Performing a Crash Dump to Disk

Once you have activated a crash dump disk, your system is ready to perform crash dumps to disk when needed. When a system halt occurs, you can perform the actual crash dump to disk in either of two ways:

- Automatically, by using System Recovery from the Maintenance Processor
- Manually, by using the Maintenance Processor command RUN 661

In either case, this operation writes the crash dump information on the crash dump disk. This preserves the crash information so that you may perform a Resident Forced Shutdown (RFS) and a system reboot.

You can manually perform a crash dump to disk immediately following a system crash by issuing the following Maintenance Processor (VCP) commands from the system console:

```
CP> SYSCLR
DPM006:Central Processor system initialization completed.
       02 Aug 91 11:47:00 Fri
CP> RUN 661
Initializing dump disk 120762 .... OK
Beginning partial dump ........
CORE dump done  12591 records written, 20345 left on disk
MAPS dump done     47 records written, 20298 left on disk
PIOS dump done     65 records written, 20233 left on disk
Crash dump to disk 120762 completed.
DPM400: CPU halted at 000014/004707: 003776
       02 Aug 91 11:50:02 Fri
CP>
```

If the activated disk is too small to accommodate the crash dump or unrecoverable problems occur during the crash dump to disk, CDD prompts you to select crash dump to tape rather than crash dump to disk.

Analyzing a Crash Dump to Disk

You can use FS_RECOVER 3.0 or greater to analyze a crash dump disk. FS_RECOVER can analyze a crash dump on the crash dump disk itself, or a crash dump recovered to a file. Although FS_RECOVER can read a crash dump directly from the crash dump disk, it is usually preferable to recover the crash dump before performing FS_RECOVER analysis, for the following reasons.

- If the system crashes again before the dump is moved, the existing dump is overwritten or else the new dump is not taken.
- In order to make a copy of the dump available for use by PrimeService, you must recover the crash dump to a file and then save it using MAGSAV.

Use the CDD –RECOVER_DUMP option to perform this operation. CDD –RECOVER_DUMP copies the crash information stored on the system's crash dump disk into a crash dump file stored in a user-specified file system directory.

Recommendations

Following is a summary of recommendations for the use of crash dumps. More detailed information about making crash dump disks can be found in the *Operator's Guide to System Commands* and the *Operator's Guide to File System Maintenance.*

- Always take a crash dump.
- Take the crash dump immediately after the crash, and before using RFS, so that an accurate representation of the disk subsystem at the time of the crash may be obtained. The best way to control this process is by using Automated System Recovery.
- Use CDD instead of CDT if at all possible.
- Use CDD –INFO to determine how much disk space you need to allocate on your crash dump disk.
- Take a partial crash dump rather than a full dump.
- Recover the crash dump from disk using INIT_RECOVER.CPL before using FS_RECOVER to analyze it so that the crash dump disk is ready to take another crash dump.

5 Crash Recovery Facilities

This chapter documents the crash recovery facilities Resident Forced Shutdown (RFS) and FS_RECOVER.

- Resident Forced Shutdown (RFS) attempts to shut down all local disk partitions following a system halt or hang. It performs a normal shutdown on those disk partitions that were not active at the time of the system crash, and thus do not require FIX_DISK processing. It suggests FIX_DISK processing for those local disk partitions that it could not successfully shut down.
- FS_RECOVER analyzes a crash dump to determine what type of FIX_DISK recovery is necessary. FS_RECOVER works in conjunction with AUTOPSY to analyze crash dumps and determine the integrity of the file system. It reduces the mean time to recover by using partial fixes and temporarily delaying full fixes.

These facilities are generally used together. Following a system crash, perform the following steps:

1. Generate a crash dump.
2. Run RFS.
3. Cold start the system.
4. Use FS_RECOVER to analyze the crash dump and to generate FIX_DISK CPLs, and run FIX_DISK where recommended.

Together, these two facilities can significantly reduce downtime of local disk partitions following an unexpected system event.

Note RFS and FS_RECOVER work together to restore file system integrity. Neither of these facilities guarantees data integrity.

Resident Forced Shutdown (RFS)

Resident Forced Shutdown (RFS) attempts to shut down local disk partitions after a system halt or hang. It successfully shuts down those disk partitions that do not require FIX_DISK processing and identifies those disk partitions that require FIX_DISK processing before adding the disk during restart. It is not necessary to run RFS after a successful forced shutdown.

Only those disk partitions which had file system transactions in progress actually require FIX_DISK processing during restart. (**File system transactions** are automatically defined by PRIMOS system software whenever a file system object is created, deleted, extended, or truncated.) Other active disk partitions not having had ongoing file system transactions at halt time will be shut down and therefore restarted without FIX_DISK processing. (FIX_DISK processing is also required if an uncorrected disk write error occurs, either while the system is running, or while performing RFS processing.) It is estimated that less than 20 percent of active local disk partitions have transactions in progress at any given time. Therefore, limiting FIX_DISK operations to only those partitions can significantly speed the time required to restart the system.

No modification of user programs or procedures is required to use RFS.

Note All disks must be in a stable state for RFS to process them reliably. Therefore, when first installing PRIMOS on your system, you should make sure that no prior file system damage exists on your disks. You can do this by verifying the messages displayed when each disk is added, or by running FIX_DISK on all local disk partitions.

Running RFS

Following a system halt or hang, you may run the RFS procedure from the supervisor terminal or, if you have configured ASR, RFS will be initiated automatically following the crash dump. The latter strategy is recommended by Prime. If you intend to use FS_RECOVER, you must generate a crash dump *before* running RFS because, otherwise, FS_RECOVER would have no way of determining the exact state of the file system at the time of the crash and, therefore, its recommendations would be suspect. Remember that RFS and FS_RECOVER were designed to work together.

To manually invoke RFS, do so in the following manner:

```
CP> SYSCLR
CP> RUN 662
```

If the system is hung, you must first stop the main processor. Press the escape key twice (<esc><esc>) to enter the Maintenance Processor, then issue the Maintenance Processor command STOP. If the system is already halted, you can omit these steps. Then execute the RFS procedure by issuing the SYSCLR command, then RUN 662. If RFS halts while executing, it can be restarted; it continues execution on the next disk partition.

RFS performs the following steps.

1. The RFS routine flushes all modified locate buffers. This ensures that all disk partitions that do not have transactions in progress will be up-to-date when they are shut down. It also increases the chances of maintaining user data integrity on all other partitions even though they will not be able to be shut down properly. RFS restores file system integrity; it may not always restore data integrity.

2. RFS displays a partition status message on the system console as it processes each local disk partition. This message contains the partition's name and pdev. RFS then displays a message that describes the status of each disk partition:

```
*** From RFS: Forced shutdown started!
        Shutting down partition    2060 ... OK
        Shutting down partition    3062 ... OK
        Shutting down partition    3560 ... OK
        Shutting down partition    2266 ... OK
        Shutting down partition    6260 ... OK
        Shutting down partition    2264 ... run FIX_DISK
        Shutting down partition   41666 ... OK
                                .
                                .
                                .
```

3. When RFS has completed, it displays the following message at the system console:

   ```
   *** From RFS: Shutdown completed.
   ```

   and then halts the system. Follow standard procedures for crash dump analysis and/or re-booting the system.

Note A warm start is not permitted after running RFS; you must cold start the system.

Summary

Always use RFS, even if for some reason you are not planning to use FS_RECOVER. Prime estimates that the probability of PRIMOS file system corruption is reduced from 33 percent to 1 percent with RFS. In addition, the probability of database damage is reduced from 17 percent to 4 percent.

What Is FS_RECOVER?

FS_RECOVER is a crash recovery tool provided since Rev. 23.1. It is an Independent Product Release (IPR) that is supported on all PRIMOS revisions 21.0 and higher (model number 8503FSR). It is an optional product at Rev. 23.1, functionally independent of Rev. 23.1 and installed separately. Installation instructions are provided in this chapter.

This section describes

- The effects of a system crash on your file system
- What FS_RECOVER does
- How FS_RECOVER works
- Some caveats related to FS_RECOVER

Effects of a System Crash on Your File System

A system crash is an unexpected event. It can happen while PRIMOS is updating or changing the file system. If it does, it may be impossible to access some or all of the files on the partitions that were active at the time of the crash. The only way to correct this problem is to run FIX_DISK on the affected partitions.

Note The term *file system*, as used here, refers to the data structures used by PRIMOS to find all the records for files on a partition.

What Does FS_RECOVER Do?

The main goal of FS_RECOVER is to reduce file system recovery time following a system crash. This allows you to make the file system available to users sooner. FS_RECOVER can also assess the general state of your file system and provide an automated interface to FIX_DISK, even if your system has not crashed.

If your system did crash and you took a crash dump, you can use FS_RECOVER to read and analyze the crash dump. FS_RECOVER determines

- Which partitions need to be fixed *immediately*
- Which partitions need fixing that can be *deferred* to a more convenient time
- Which partitions were unaffected by the crash

FS_RECOVER also determines the correct FIX_DISK options for those partitions that must be fixed immediately and provides an automated facility for running FIX_DISK.

If your system has not crashed or if your system crashed but you did not take a crash dump, you can use FS_RECOVER to make a generalized assessment of the state of your partitions. FS_RECOVER determines which partitions are damaged, and which partitions are clean. (The term *clean partition*, as used here, refers to a partition which does not cause PRIMOS to generate a warning message at the time it is mounted, or added. Refer to Appendix C of the *Using FS_RECOVER* manual for a listing of these warning messages.)

FS_RECOVER also determines the correct FIX_DISK options for the damaged partitions and provides an automated facility for running FIX_DISK.

FS_RECOVER Using a Crash Dump

When you reboot your system after a crash, you should allow PRIMOS.COMI to mount all your local disk partitions, but do not start any disk mirrors and do not allow users to log in.

Note If you correctly placed INIT_RECOVER.CPL within PRIMOS.COMI (when installing FS_RECOVER) this is automatically accomplished, and you can invoke FS_RECOVER by pressing Control-P when prompted to do so at cold start.

When the system is running, use FS_RECOVER to read the crash dump and perform the recovery analysis.

When performing a crash dump recovery analysis, FS_RECOVER uses two major sources of data.

- The crash dump itself, which contains detailed information about what was happening on your system at the time of the crash
- The *current* state of the disk partitions

The current state of the disk partitions is available only if each disk is added. The current state information is merged with the crash dump information to form a recommendation for each partition that was mounted at the time of crash.

When analyzing the crash dump, FS_RECOVER looks for three types of information, as follows.

Crash type	The type of crash, which affects the types of recommendations FS_RECOVER makes for running FIX_DISK, is determined from the machine state.
Activity	FS_RECOVER identifies file system activity at the time of the crash in order to indicate where damage to the integrity of the file system may be.

Prior Corruption	FS_RECOVER looks for any information that might indicate that file system damage existed *prior* to the crash, such as flag bits set in the DSKRAT indicating that a disk was not cleanly shutdown on some previous occasion.

Note Be aware that all indications of prior damage are *not* guaranteed to be in the crash dump. This is the most important reason why you should follow the FS_RECOVER recommendations and perform the deferred fixes as soon as you can.

Generally, FS_RECOVER analyzes all this information in less than ten minutes.

After the analysis is complete, FS_RECOVER displays a recommendation for each partition that was mounted at the time of the crash. Each recommendation includes three pieces of information:

- A list of pathnames for any files on the partition that were active at the time of the crash. The pathnames may or may not be complete, depending on the amount of file system information in the locate buffers at the time of the crash.
- A statement telling you
  - If FIX_DISK needs to be run on the partition
  - What FIX_DISK options should be used
  - Whether you should run FIX_DISK immediately or if you can defer running FIX_DISK to a more convenient time

  A facility is provided to change the FIX_DISK recommendation, should you decide to do so.
- If a partition was mirrored, the recommendation will tell you which half of the mirrored pair is to be used as the primary when you restart the mirror with the MIRROR_ON command.

When the recommendations are complete, FS_RECOVER builds a CPL program for each partition requiring immediate FIX_DISK. These CPL programs are designed to be run by phantoms. FS_RECOVER then determines how many phantoms will be needed to execute all the CPL programs. This determination will take into account the number of available phantoms, the number of FIX_DISK sessions required, the number of disk drives containing partitions requiring FIX_DISK, and the PRIMOS limit on the number of assignable disks.

FS_RECOVER then tells you how many phantoms are required, and asks you how many phantoms you wish to use. After you have made that decision, FS_RECOVER creates a phantom called the FIX_DISK Monitor that controls the phantoms that perform the FIX_DISK sessions. These phantoms keep separate, date stamped, COMO files for each FIX_DISK session so you can monitor their progress and results. When all of the FIX_DISK sessions have completed, the FIX_DISK Monitor phantom logs out.

FS_RECOVER Without a Crash Dump

You can use FS_RECOVER to make a generalized assessment of the state of your locally mounted partitions. If any one of these partitions are damaged, FS_RECOVER asks if you want to run FIX_DISK on the damaged partitions. If you answer yes, FS_RECOVER sets up for automated FIX_DISK the same way it does for a crash dump recovery analysis.

You can use FS_RECOVER without a crash dump. For example, if you just had a system crash but were unable to get a crash dump, you can take advantage of the automated FIX_DISK facilities of FS_RECOVER. You can also identify and repair partitions that had a defer recommendation from a previous crash dump analysis.

Considerations When Using FS_RECOVER

The crash dump recovery analysis portion of FS_RECOVER works best if you use it immediately after each crash. FS_RECOVER may not work correctly if you attempt to analyze an old crash dump or a crash dump that was taken before other crashes.

The following are other considerations for using FS_RECOVER.

- FS_RECOVER cannot always display the full pathnames of every file affected by a crash. The pathnames are generated using the contents of the locate buffers found in the crash dump. The more pathname information found in the locate buffers,the more complete the pathnames FS_RECOVER can display. Pathnames cannot be generated for CAM files on robust partitions; however, you may use RECORD_TO_PATH.
- The automated FIX_DISK facilities of FS_RECOVER cannot be used to repair the command device (COMDEV). File system damage on the command device must be repaired by running FIX_DISK with the –COMDEV option at the supervisor terminal.
- FS_RECOVER cannot be run by phantoms.

Installing FS_RECOVER

This section discusses installation of FS_RECOVER on your system, including any changes you may have to make to the system.

FS_RECOVER Installation Tape

Prime distributes FS_RECOVER on a standard 1600bpi, MAGSAV-format tape. This tape is included in your Rev. 23.3 package. You mount the tape on any tape drive and restore the contents into any convenient partition. Restoring the tape contents creates a directory named FS_RECOVER, which contains about 1500 disk records. You install FS_RECOVER from that directory.

Using FS_RECOVER.INSTALL.CPL

To install FS_RECOVER, attach to the FS_RECOVER directory and execute the FS_RECOVER.INSTALL.CPL file. The installation file copies FS_RECOVER>SYSTEM_DEBUG* to a top-level directory named SYSTEM_DEBUG* on your command device (COMDEV). If you have several command devices, you may want to modify FS_RECOVER.INSTALL.CPL to install FS_RECOVER on all of them. The installation process also copies two new search rules files into SEARCH_RULES*.

Changes to Search Rules

FS_RECOVER uses four search rules files:

AUTOPSY.SR
MAPS.SR
COMMAND$.SR
ENTRY$.SR

The FS_RECOVER.INSTALL.CPL file automatically installs the first two files in SEARCH_RULES*. The last two search rules files are part of standard PRIMOS and already exist. The installation modifies these two files as follows.

- The COMMAND$.SR search rule defines where PRIMOS looks for external commands. The default is the directory CMDNC0 on the COMDEV. The installation adds SYSTEM_DEBUG* to the list so that, as a minimum, COMMAND$.SR contains CMDNC0 and SYSTEM_DEBUG*.
- The ENTRY$.SR search rule defines where PRIMOS looks when it attempts to resolve a dynamic link. The installation adds SYSTEM_DEBUG*>AUTOPSY.RUN.

ACL Requirements

FS_RECOVER contains security checks to ensure that only the supervisor terminal (User 1), the user ID SYSTEM, or the System Administrator use FS_RECOVER. In addition, SYSTEM_DEBUG* and SYSTEM_DEBUG*>CRASH require some specific ACLs. These ACLs are shown in the following example.

When you follow this example, substitute your System Administrator ID for *system_admin*. To set the correct ACLs, enter the following:

```
OK, SAC <0>SYSTEM_DEBUG* SYSTEM:ALL system_admin:ALL $REST:LURX
OK, SAC <0>SYSTEM_DEBUG*>CRASH SYSTEM:ALL system_admin:ALL $REST:NONE
```

Segment Requirements

Caution The user ID SYSTEM and the System Administrator's ID must be configured for at least 128 dynamic segments. Failure to provide this minimum limit may cause unpredictable results. When you invoke FS_RECOVER, it checks the number of dynamic segments configured and prints warning messages if the number is too small.

Changes to PRIMOS.COMI

In order to complete the installation of FS_RECOVER, you must change your PRIMOS.COMI to include running the INIT_RECOVER.CPL program in SYSTEM_DEBUG*. The placement of INIT_RECOVER.CPL within PRIMOS.COMI must occur *after* all local disk partitions are mounted, but *before* user logins are allowed:

```
STI -TZ 0500 -DLST YES                      /* Sets up time-zone information.
START_DSM                                   /* Startup DSM.
ADD_DISKS.CPL                                         /* Mount local disks.
R SYSTEM_DEBUG*>INIT_RECOVER.CPL -PAUSE     /* Invoke FS_RECOVER, if needed.
MAXUSR                                                /* Allow user logins.
```

Note If you omit the -PAUSE option, you will not be able to invoke FS_RECOVER while PRIMOS.COMI is running.

When PRIMOS.COMI invokes INIT_RECOVER.CPL, INIT_RECOVER first displays a header and then saves the PRIMOS maps. The -PAUSE option causes PRIMOS.COMI to display the following message and pause for thirty seconds to allow you to press Control-P, aborting PRIMOS.COMI and automatically invoking FS_RECOVER.

```
Pausing briefly to allow you to enter CONTROL-P to invoke FS_RECOVER.
Otherwise, PRIMOS.COMI will continue.
```

If you do not press Control-P at this time to invoke FS_RECOVER, this message displays and PRIMOS.COMI continues.

```
Wait completed, continuing with coldstart.
```

Allocating Disk Records for Tape Dumps

If you must use crash dump to tape, remember that FS_RECOVER cannot work on the raw data contained on the crash dump tape. You must first put the data on the disk into the file system. FS_RECOVER has special facilities to do this, but sufficient free disk records must exist. Since crash dump files can be rather large, you should set aside some dedicated space on a partition. It is recommended that you set aside this disk space in the <0>SYSTEM_DEBUG*>CRASH directory, but this is not a requirement; you can put the crash data file on any partition.

The amount of disk space required for a crash dump file varies with the system configuration and the type of crash dump. The crash dump procedure is for a *partial* dump, which is all that FS_RECOVER usually needs. *Full* crash dumps are virtually never needed and take up considerably more disk space. In either case, you can use the following guidelines for disk space planning.

1. Use the STATUS SYSTEM command at the supervisor terminal to determine the kilobytes (KB) of memory in your system.

```
OK, STATUS SYSTEM

System STAN is currently running PRIMOS rev. 23.3
Copyright (c) Prime Computer, Inc. 1991
32768K bytes memory in use

OK,
```

2. If you are generating *partial* crash dumps go to Step 3.

   For *full* crash dumps, calculate the base number of disk records required, as follows, and go to Step 4. The base number of records for a full tape dump is equal to the KB of memory divided by two:

```
KB of memory
in the system
--------------        =  base number of disk records
           2
```

3. For *partial* crash dumps, calculate the base number of disk records required by using one of the following formulas and then go to Step 4.

   Use this formula if your system has 32768 KB *or less*:

```
(KB of memory) * (0.35) = base number of disk records
```

   Use this formula if your system has *more than* 32768 KB:

```
(KB of memory) / 4  =  base number of disk records
```

4. If your system is a 6150™, 6350™, 6450™, 6550™, or a 6650™ (a 6000 series machine), add 66 to the base number of disk records calculated in either Step 2 or Step 3. This number represents the total number of disk records you should set aside for a crash dump on these machines. (You can use the SYSTEM_INFO command function, described in this document, to determine the model of your system.)

Examples of Calculating Required Disk Records: If your system is a 6350 with 65536 KB of memory and you use partial crash dumps, the number of disk records to set aside is as follows:

```
( 65536 / 4 ) + 66  =  16450 disk records
```

If your system is a 2550™ with 8192 KB of memory and you use partial crash dumps, the number of disk records to set aside is as follows:

```
8192 * 0.35  =  2868 disk records
```

Using FS_RECOVER

The recommended method to use FS_RECOVER is to invoke it from within the INIT_RECOVER.CPL routine as PRIMOS.COMI is booting the system. You can also manually invoke FS_RECOVER in three ways:

- at the supervisor terminal
- while logged in as the System Administrator
- under the user ID SYSTEM

After invocation, FS_RECOVER makes several integrity checks to ensure that it was installed correctly. If any of the checks fails, FS_RECOVER displays an error message and returns you to PRIMOS command level.

Recommended Strategy After a System Crash

If your system crashes, follow this procedure:

1. Generate a crash dump.
2. Run RFS *after* generating the crash dump to disk. (RFS accomplishes a forced shutdown of PRIMOS and shuts down each partition in an orderly manner.)
3. Cold start your system.

   (If you are using ASR, the three steps listed above are accomplished automatically.)

4. PRIMOS.COMI executes until it encounters the INIT_RECOVER.CPL command line. It then displays the following message and pauses for 30 seconds.

```
Pausing briefly to allow you to enter CONTROL-P to invoke FS_RECOVER.
Otherwise, PRIMOS.COMI will continue.
```

Press Control-P to abort PRIMOS.COMI and to invoke FS_RECOVER.

Note If you use the –AA option of the SYSTEM_RECOVER command as part of ASR, you will not have a chance to enter CONTROL–P to interrupt PRIMOS>COMI, and you will not be prompted to enter CONTROL–P in any way. Use of the –AA option assumes that you wish to have fully-automated recovery of your system.

---

5. FS_RECOVER displays its Main Menu. Use Main Menu Option 3 to assess the health of your disk partitions.

   A. If your system crashed because of a forced shutdown or if you successfully ran RFS, all the partitions may be clean. If all the partitions are clean, exit FS_RECOVER and continue PRIMOS.COMI by entering CO CONTINUE 6.

   B. If any of the partitions are damaged, do *not* initiate automated FIX_DISK while you are in Main Menu Option 3. Instead, go back to the Main Menu and select Option 1 to read the crash tape. Then select Main Menu Option 2 to analyze the crash dump file. Execute all recommended *immediate* FIX_DISK sessions and then continue PRIMOS.COMI by entering CO CONTINUE 6.

   C. If the crash dump analysis indicates that there are deferrable FIX_DISK sessions, you can reinvoke FS_RECOVER at a convenient time later and use Main Menu Option 3 to repair the damaged partitions. Continue PRIMOS.COMI by entering CO CONTINUE 6 at this time.

If your command device (COMDEV) is damaged, you must use FIX_DISK at the supervisor terminal.

FS_RECOVER Main Menu

If the installation integrity checks pass when you invoke FS_RECOVER, FS_RECOVER displays its Main Menu and prompts you for a choice:

```
[FS_RECOVER Rev 3.0 Copyright (c) 1991, Prime Computer Inc.]

MAIN MENU:

(1) Read crash tapes
(2) Perform crash recovery analysis
(3) Display state of currently mounted disks
```

```
Enter a menu number, or (Q)uit or (M)enu:
```

You have several choices, as follows:

- Use Option 1 when you want to read a crash dump tape into a disk file.
- Use Option 2 to perform a file system recovery analysis on a crash dump file that you created with Option 1. You can then invoke automated FIX_DISK.
- Use Option 3 to assess the state of all currently-mounted local disk partitions. You can then invoke automated FIX_DISK.
- Enter ! *<PRIMOS command line>* to execute a PRIMOS command without leaving FS_RECOVER.
- Enter M to cause FS_RECOVER to redisplay the menu.
- Enter Q to leave FS_RECOVER and exit to PRIMOS command level.

Breaking Out of FS_RECOVER: When you select a Main Menu option, you can stop execution of FS_RECOVER at any time by using Control-P. The *only* exception to this is when you are selecting a choice from the FIX_DISK Menu. While you are in the FIX_DISK Menu, Control-P, ECL support, and PRIMOS command line support are disabled. If you do stop FS_RECOVER by pressing Control-P, you see the following:

```
**** Break! ****
(A)bort, (C)ontinue, or (R)eturn to Main Menu? A
OK,
```

You can abort FS_RECOVER, continue with the interrupted selection, or go back to the Main Menu. You can get back to the Main Menu also by simply entering Q or QUIT in most cases. For example:

```
Enter a menu number, or (Q)uit or (M)enu: 1

Mount the first reel of the crash tape(s) and enter the magtape unit
number.
You may also enter:
                    -"! <PRIMOS command>"
                    -"Q" or "QUIT" to return to the main menu.
Tape unit (9 track): Q

MAIN MENU:
```

Executing PRIMOS Commands Within FS_RECOVER: In some places where FS_RECOVER prompts you for input, you can also enter PRIMOS commands. In many instances, as in the previous example, FS_RECOVER explicitly tells you that you may enter PRIMOS commands. To enter a PRIMOS command line from an FS_RECOVER prompt, precede the PRIMOS command line with ! (an exclamation point). Abbreviations, wildcarding, and iteration lists are fully supported. After the PRIMOS command completes, FS_RECOVER prompts for input.

Using ECL Within FS_RECOVER: The ECL environment within FS_RECOVER is totally separate from your PRIMOS ECL environment.

ECL is automatically enabled within FS_RECOVER except in these cases:

- ECL is not installed.
- You invoke FS_RECOVER from the supervisor terminal on a system running a PRIMOS revision *prior* to Rev. 22.1.

Reading Crash Dump Tapes

FS_RECOVER cannot read the raw data on the crash dump tapes. You must use Main Menu Option 1 to read the data from tape into a disk file before FS_RECOVER can analyze the data. The tapes need to be successfully read only once, but individual reels with unrecovered tape errors may be reread as many times as necessary. If you stop reading tapes at the end of a reel, you can leave FS_RECOVER and then come back at some later time and continue reading the tapes, starting with the next reel. Reels must be read in the order that they were written.

To read crash dump tapes, select Option 1 from the Main Menu. Follow the prompts to mount the first reel of the crash dump tapes on a tape drive and enter the tape drive unit number:

```
[FS_RECOVER Rev 3.0 Copyright (c) 1991, Prime Computer Inc.]

MAIN MENU:

(1) Read crash tapes
(2) Perform crash recovery analysis
(3) Display state of currently mounted disks

Enter a menu number, or (Q)uit or (M)enu: 1
Mount the first reel of the crash tape(s) and enter the magtape unit number.
You may also enter:
                      -"! <PRIMOS command>"
                      -"Q" or "QUIT" to return to the main menu.
Tape unit (9 track): 0
```

Checking the Tape Drive: When you enter a magtape unit number, FS_RECOVER attempts to assign the tape drive. If the assign fails, you get an error message followed by another prompt for a magtape unit:

```
Tape unit (9 track): 0
PRIMOS error code 39 while assigning MT0. Device in use.
Mount the first reel of the crash tape(s) and enter the magtape unit number.
You may also enter:
                      -"! <PRIMOS command>"
```

```
                          -"Q" or "QUIT" to return to the main menu.
Tape unit (9 track):
```

After assigning the tape drive, FS_RECOVER checks to ensure that a tape is mounted on the tape drive and that the drive is online and ready. If any of these checks fail, you get an error message followed by the magtape unit prompt.

```
Tape unit (9 track): 0
Device offline or not ready.
Mount the first reel of the crash tape(s) and enter the magtape unit number.
You may also enter:
                          -"! <PRIMOS command>"
                          -"Q" or "QUIT" to return to the main menu.
Tape unit (9 track):
```

Crash Dump File: When the magtape drive is online and ready, FS_RECOVER prompts for the pathname of the file you want to put the crash dump data into. Ideally, this should be a file in SYSTEM_DEBUG*>CRASH, but this is not a requirement; you can put the crash dump data file on any partition. Use a unique name for each crash dump file so that the file is easy to identify. The recommended naming convention includes the system name, followed by a date/time stamp. For example, if your system is named MOLLY and the crash occurred on April 19, 1992 at 1:30 p.m., the recommended name for the crash dump data file is one of the following:

MOLLY.92.0419.1330

filename.[DATE -FTAG]

Reading the Tape: After you enter the crash dump pathname, FS_RECOVER reads the tape. When the end of the crash dump is detected on tape, FS_RECOVER returns you to the Main Menu. If an end-of-tape occurs before the end of the crash dump, FS_RECOVER prompts for the next reel. At this point, you can mount the next reel and enter the magtape unit number:

```
End of reel 1; 32766 records read; 32766 records dumped; 0 errors.
Are there any more reels? YES
Tape unit number (9 track): 0
```

Performing the Recovery Analysis

After FS_RECOVER reads the tape, select Main Menu Option 2 (Perform Crash Recovery Analysis) after being sure that you meet the following requirements.

- A crash dump file must exist. That is, at some point you must have used Main Menu Option 1.
- When you select Main Menu Option 2, you must know the pathname of the working directory (the directory containing the FS_RECOVER CPL programs and the crash dump file) that you want FS_RECOVER to use.

FS_RECOVER Working Directory: The FS_RECOVER working directory is where FS_RECOVER expects to find the two CPL programs, RUN_FIX_DISK.CPL and FIX_DISK_MONITOR.CPL. FS_RECOVER also uses the working directory to keep COMO files and to build CPL programs for automated FIX_DISK. Prime recommends that you keep all your crash dump files in the working directory also, but this is not a requirement.

The default working directory is SYSTEM_DEBUG*>CRASH. However, you can create and use a different working directory. If you do, copy RUN_FIX_DISK.CPL and FIX_DISK_MONITOR.CPL from SYSTEM_DEBUG*>CRASH into the new working directory.

Here is an example of how to create a new working directory.

```
OK, A MFD 1
OK, CREATE CRASH.NEW
OK, COPY SYSTEM_DEBUG*>CRASH>RUN_FIX_DISK.CPL *>CRASH.NEW>==
OK, COPY SYSTEM_DEBUG*>CRASH>FIX_DISK_MONITOR.CPL *>CRASH.NEW>==
OK,
```

When you select Main Menu Option 2, FS_RECOVER prompts you to enter the pathname of the working directory and displays a default working directory pathname. To select the default working directory, simply press Return.

```
Enter pathname of working directory (default="<0>SYSTEM_DEBUG*>CRASH"): <cr>
```

Pathname of the Crash Dump File: Next, FS_RECOVER prompts you to enter the pathname of the crash dump file you want to analyze. If you just finished using Main Menu Option 1 to read crash dump tapes, FS_RECOVER uses the pathname of the file you read the tapes into as the default pathname. If you want to use the default pathname, simply press Return. Otherwise, enter the pathname of the crash dump file you want to analyze.

FS_RECOVER then attempts to load the crash dump, which takes about one minute.

Example of Doing the Analysis: Following is an example of the display when you select Option 2.

```
MAIN MENU:

(1) Read crash tapes
(2) Perform crash recovery analysis
(3) Display state of currently mounted disks

Enter a menu number, or (Q)uit or (M)enu: 2

---------------------------------------------------------------------------
                                 *** RECOVERY ANALYSIS ***

Enter pathname of working directory (default="<0>SYSTEM_DEBUG*>CRASH"): <cr>
Crashdump pathname: SYSTEM_DEBUG*>CRASH>MILO.121291.0100

(Beginning crashdump load, please wait...)
```

Session COMO File: After FS_RECOVER successfully loads the crash dump, it starts a session COMO file in the working directory. The name of the COMO file is always unique and consists of the crashed system's name and a date/time stamp.

```
(Beginning crashdump load, please wait...)

Your session COMO file is <0>SYSTEM_DEBUG*>CRASH>RES-C4.910405.100048.
```

Messages Indicating the Machine State: After FS_RECOVER starts the session COMO file, FS_RECOVER determines the machine state at the time of the crash. *Record this information in your System Log Book.*

The following messages indicate possible machine states:

```
The machine was stopped by a MASTER CLEAR.

The machine did not halt; it was STOPPED by the
Maintenance Processor.

The machine halted at x(0)/xxxxxx; xxxxxx+'0

PRIMOS executed a Slow Halt at x(0)/xxxxxx; xxxxxx+'0

PRIMOS stopped the machine using a Forced Shutdown.

The machine was stopped using the "SHUT ALL" command at
the System Console.
```

Messages During Analysis of Data: After determining the machine state, FS_RECOVER begins analysis of the data. Analysis can take up to ten minutes. During this time, you see several informational messages:

```
(Building Unit Info table, please wait...)
(Validating Disk Driver data structures, please wait...)
(Validating state of the Locate subsystem, please wait...)
(Validating Unit Table Hash, please wait...)
(Building nllock LOCKLIST database, please wait...)
(Building nllock owners database, please wait...)
(Validating any resident DSKRATs, please wait...)
```

Occasionally you may see other warning or caution messages interspersed with the informational messages. Refer to Appendix B of *Using FS_RECOVER* for more information.

Recommendations for Running FIX_DISK

After FS_RECOVER completes the analysis, it presents a summary of each partition with a recommendation to run FIX_DISK. Prior to displaying the recommendations, FS_RECOVER displays this information:

```
You will now be shown an individual summary of activity for each partition
that was mounted at the time of the crash.  After each summary there will
be a FIX_DISK recommendation.  To accept the recommendation simply answer
"YES" or press <RETURN>).  If you do *not* want to accept the recommendation
enter one of the following:
"SKip"          to do nothing to the partition.
"CHeck"         to run "FIX_DISK" (without the "-FIX" option)
"FUll"          to run "FIX_DISK -FIX".
"PArtial"       to run "FIX_DISK -FIX -PARTIAL"
"FAst"          to run "FIX_DISK -FIX -FAST"

"HELP"          to see this screen.
"QUIT"          to return to the Main Menu.

Press <RETURN> when you are ready to see the partition state summary:
```

The recommendation falls into one of four categories:

Immediate FIX_DISK

You should run FIX_DISK before using the partition; file system and data integrity are compromised. FS_RECOVER will attempt to use either the –FAST option or the –PARTIAL option to minimize FIX_DISK session time. (The -PARTIAL option is supported but undocumented.) By default, FS_RECOVER builds CPL files to run any immediate FIX_DISK.

Deferred FIX_DISK

You can add the partition but file system integrity may be compromised. If no database recovery is required for the files on the partition, you can make the partition available for use immediately. However, at some convenient time, you *must* run full FIX_DISK on the partition.

Not Required

The partition was clean before the crash and the crash did not damage the partition. You should find all your partitions in this state after a successful forced shutdown or a successful invocation of RFS.

Note If no database recovery is required for the files on the partition, you can make the partition available for use immediately.

No Recommendation

If FS_RECOVER detects that a disk drive containing a partition that was mounted at the time of the crash has been repartitioned, no recommendation will be given.

Example of Immediate FIX_DISK: Here is an example of a partition requiring immediate FIX_DISK:

```
LDEV:    '1    PDEV: '6062      NAME: <BAYGRP> (robust)
Warning: The crashdump indicates 2 serious problems with this partition:
         A file system transaction was in progress at the time of the crash.
```

```
        Portions of the DSKRAT were modified, but not written to the disk.

Activity  File Type   Pathname
--------  ----------  --------
L         SAM file    <BAYGRP>UNIX01
L         ACL dir     <BAYGRP>ANDYG.RSRCH
LT        DAM file    <BAYGRP>ANDYG.RSRCH>TABLE

File Activity Codes:
L : file had modified unflushed records in Locate subsystem.
T : file may have had an in-progress transaction.
RECOMMENDATION: run "FIX_DISK -FIX".
Is this what you want to do? Y
Currently, PDEV '6062 <BAYGRP> is not mounted.
Do you want it mounted after the FIX_DISK completes? Y
```

Example of Deferred FIX_DISK: Here is an example of the summary for one partition requiring a deferred FIX_DISK:

```
--------------------------------------------------------------------------
LDEV: '2 PDEV: '3462 NAME: <QUALF2>


No file system activity indicated; schedule a FIX_DISK at your convenience.
--------------------------------------------------------------------------
```

In this example, there was no indication of file system activity or serious problems; an immediate FIX_DISK is not required. If no special database recovery is needed for the files on this partition, you can make it available to users. However, at some convenient time, you must run FIX_DISK to maintain the integrity of the partition's file system.

Changing a FIX_DISK Recommendation: After FS_RECOVER displays the summary and recommendation for a partition it asks you if you agree with the recommendation. If you answer YES, FS_RECOVER continues with the next partition summary. If you answer NO, FS_RECOVER enters the FIX_DISK Menu, which then asks you what you want to do with the partition. While you are in the FIX_DISK Menu, Control-P, ECL support, and PRIMOS command line support are disabled.

```
--------------------------------------------------------------------------
LDEV: '2 PDEV: '3164 NAME: <DISK02>

RECOMMENDATION: run "FIX_DISK -FIX".
Is this what you want to do?
```

At this point, enter a valid choice from the summary menu shown previously in Recommendations for Running FIX_DISK or enter NO to see a list of valid choices:

```
Valid choices are:
"SKip"          to do nothing to the partition.
"CHeck"         to run "FIX_DISK" (without the "-FIX" option)
"FUll"          to run "FIX_DISK -FIX".
"PArtial"       to run "FIX_DISK -FIX -PARTIAL"
"FAst"          to run "FIX_DISK -FIX -FAST"
```

```
"HELP"          to see this screen.
"QUIT"          to return to the Main Menu.

Is this what you want to do?
```

After you enter a valid choice, FS_RECOVER continues with the next partition. After you have answered the queries for all affected partitions, FS_RECOVER summarizes your choices.

FS_RECOVER Summary Display

After all the partitions have been individually summarized, FS_RECOVER displays a general summary of all the FIX_DISK recommendations. FS_RECOVER then asks you if all the recommendations are satisfactory. If you answer NO, FS_RECOVER repeats the individual partition summaries so that you can change recommendations for running FIX_DISK.

```
                *CURRENT*                       CURRENTLY      TYPE OF
LDEV            PDEV           NAME             MOUNTED?       FIX_DISK NEEDED             COMMENTS
----            ----           -----            -----------    ------------------          --------
                0                               6060           <UNIX00>       yes          none
COMDEV
                1                               6062           <UNIX01>       yes          immediate, full

                2                               3164           <UNIX02>       yes          immediate, full

3 partitions analyzed, 2 partitions require FIX_DISK.
2 immediate FIX_DISKs, 0 deferrable FIX_DISKs.

Are these FIX_DISK recommendations satisfactory? YES
```

Automated FIX_DISK

If there are no recommendations for running immediate or deferred FIX_DISK, FS_RECOVER returns to the Main Menu. If there are deferred or immediate FIX_DISK recommendations and you answer YES, indicating that you are satisfied with the FIX_DISK recommendations, FS_RECOVER asks if you want to initiate automated FIX_DISK on all partitions requiring immediate FIX_DISK (except the Command Device (COMDEV)):

```
3 partitions analyzed, 2 partitions require FIX_DISK.
2 immediate FIX_DISKs, 0 deferrable FIX_DISKs.

Are these FIX_DISK recommendations satisfactory? YES
Do you want to initiate the immediate FIX_DISKs? YES
```

If all recommendations were for deferred FIX_DISK, or if the only recommendation for immediate FIX_DISK was for the command device, FS_RECOVER returns to the Main Menu.

Administrative Setup for Automated FIX_DISK

If you answer YES indicating that you want to initiate the immediate FIX_DISK recommendations, FS_RECOVER displays an Administrative Setup screen. In addition, if you are running FS_RECOVER from the supervisor terminal, the Administrative Setup screen asks if you want to stop the LOGIN_SERVER and reminds you to break any existing mirrors with the MIRROR_OFF command. Answering YES prohibits user logins after FS_RECOVER enables MAXUSR for the FIX_DISK phantoms. The default answer is YES.

```
                                   *** ADMINISTRATIVE SETUP ***

Do you want to stop the LOGIN_SERVER before starting FIX_DISK? NO
Forcing "MAXUSR ALL" for FIX_DISK sessions.
Attempting to startup the DISK_MANAGER.

Reminder: If any of the partitions which are about to be repaired
          are currently mirrored you must break those mirrors with
          the "MIRROR_OFF" command prior initiating automated FIX_DISK.
```

If you are *not* running FS_RECOVER from the supervisor terminal, FS_RECOVER tells you to go to the supervisor terminal and enter the following commands:

```
                                   *** ADMINISTRATIVE SETUP ***

The DISK_MANAGER must be started up prior to initiating FIX_DISK phantoms.
Enter the following command at the System Console:

"ECL -OFF"
"DISK_MANAGER -START"


In order to allow FIX_DISK phantoms to login, enter the following command
at the System Console:

"MAXUSR -PUSR 222"

If you want to prohibit user logins while FIX_DISK is running, enter the
following command at the System Console:

"STOP_LSR"

Press <RETURN> after this is done and/or you are ready to proceed: <cr>
```

Automated FIX_DISK Configuration: After you leave the Administrative Setup display, FS_RECOVER creates a subdirectory within the working directory. FS_RECOVER then builds the CPL programs for automated FIX_DISK in this subdirectory.

Next, FS_RECOVER determines how many phantoms are necessary to execute all the CPL programs. It takes into account the number of available phantoms, the number of FIX_DISK sessions required, the number of disk drives containing partitions requiring FIX_DISK, and the PRIMOS limit on the number of assignable disks.

FS_RECOVER then asks how many phantoms you would like to use:

```
                          *** FIX_DISK SETUP ***

(Building CPL programs for automated FIX_DISK, please wait...)

All the programs which will control the FIX_DISK sessions are located in:
<0>SYSTEM_DEBUG*>CRASH>FIX.RES-C4.910319.164508

The 2 partitions requiring FIX_DISK reside on 2 different disk drives.
Both of these disk drives can be worked on in parallel.  This requires one
phantom per disk drive (each phantom will do ALL the required FIX_DISKs for
a given disk drive), plus one phantom to drive the FIX_DISK_MONITOR program.
If 3 phantoms are too much, fewer (down to a minimum of 2) may be used.

Enter the number of phantoms to use (2-3) or (Q)uit: 3
```

The INIT_RECOVER –AUTO_ANALYSIS Option

FS_RECOVER does not query you, as in the preceding sections, when you use the –AUTO_ANALYSIS option and place FS_RECOVER in automated analysis mode. Instead, it analyzes the pre-configured CDD partition and automatically invokes FIX_DISK sessions on those file system partitions that it determines need immediate file structure repair.

FIX_DISK Manager Phantom

After you tell FS_RECOVER how many phantoms to use, you are prompted to begin automated FIX_DISK. You can also quit or execute PRIMOS commands prior to beginning automated FIX_DISK.

```
FIX_DISK setup is now complete, and we're ready to begin.
Enter <RETURN> to begin, "QUIT", or "! <command>": ! m -all -now -force
The system will be available in about 20 minutes. Please standby...

Enter <RETURN> to begin, "QUIT", or "! <command>": <cr>
```

When you press the Return key, FS_RECOVER initiates the FIX_DISK Monitor phantom. The FIX_DISK Monitor then begins creating phantoms to run the FIX_DISK sessions.

Disk Manager Subsystem

When running FS_RECOVER from the supervisor terminal, FS_RECOVER automatically initiates a program called the DISK_MANAGER while you are in the Administrative Setup screen. If you are *not* running from the supervisor terminal, FS_RECOVER instructs you to manually initiate the DISK_MANAGER at the supervisor terminal.

The DISK_MANAGER program services certain commands for the FIX_DISK phantoms. Due to PRIMOS restrictions, commands such as ADDISK, SHUTDN, and DISKS are *privileged* and can only be executed from the supervisor terminal. Whenever a FIX_DISK phantom needs one of these privileged commands executed, it calls the supervisor terminal. The DISK_MANAGER program allows the supervisor terminal to *listen* for these commands and then execute them on behalf of the FIX_DISK phantom.

You can still use the supervisor terminal to execute PRIMOS commands with the exception of DELSEG, ICE, and ECL, but do not enter commands that take longer than a few seconds to execute, because the DISK_MANAGER can *listen* for commands from the FIX_DISK phantoms only when the supervisor terminal is not busy.

When the DISK_MANAGER program receives a command from one of the FIX_DISK phantoms, it displays the command, along with the results, on the supervisor terminal:

```
*** DISK_MANAGER at 12 March 91 15:32
*** Starting "AD 6062" for SYSTEM (user 110).
*** Finished "AD 6062" for SYSTEM (user 110).
```

Displaying the State of Currently Mounted Disks

Main Menu Option 3 is used to make a generalized assessment of the health of all currently mounted local partitions. During this assessment, FS_RECOVER recognizes only two states that a partition can be in, as follows:

Clean A clean partition is one in which the file system structures on the partition are completely intact. This is indicated by bits set in the partition's DSKRAT that tell PRIMOS whether or not the partition had been cleanly shutdown since its last full FIX_DISK session. If the bits are not set, PRIMOS displays a warning message when the partition is mounted. (Refer to Appendix C) However, there are exceptional instances when a clean partition can become damaged *after* it is mounted. As of Rev. 23.1, PRIMOS has specialized support to make information about these exceptions available to FS_RECOVER.

Damaged A damaged partition is one that was either not clean at the time it was mounted, or it was damaged after it was mounted. If the damage occurred after the partition was mounted and you are running PRIMOS Rev. 23.1 or later, FS_RECOVER will tell you the type of problem that damaged the partition.

The following is an example of the use of Main Menu Option 3.

```
MAIN MENU:

(1) Read crash tapes
(2) Perform crash recovery analysis
(3) Display state of currently mounted disks

Enter a menu number, or (Q)uit or (M)enu: 3
--------------------------------------------------------------------------------
                                   *** SHOW LOCAL DISKS ***

                                    FIX_DISK
LDEV        PDEV        NAME        NEEDED?     COMMENTS
----        -----       ----        -------     --------
0           6060        <DISK00>    no          COMDEV
1           6062        <DISK01>    full        *Not Clean*
2           3164        <DISK02>    full        *Not Clean*

3 partitions displayed, 2 require full FIX_DISK.
```

FS_RECOVER now asks if you wish to run FIX_DISK on all partitions except the command device (COMDEV). If you answer NO, FS_RECOVER then asks if you want to run FIX_DISK on any partition. If you answer YES, FS_RECOVER sets up for automated FIX_DISK.

```
Initiate "FIX_DISK -FIX" on *ALL* disk partitions, except the COMDEV? N
Initiate "FIX_DISK -FIX" on the partitions which are not "clean"? N
Do you want to run FIX_DISK on some of these "unclean" partitions? N
Do you want to run FIX_DISK on any disk partitions, except the COMDEV? Y

You will prompted once for each partition that is not "clean".  To run
"FIX_DISK -FIX" on that partition simply answer "YES" (or press <RETURN>).
To avoid running FIX_DISK on a partition, or to run FIX_DISK with other
options, enter one of the following:
"SKip"           to do nothing to the partition.
"CHeck"          to run "FIX_DISK" (without the "-FIX" option)
"FUll" or "YES"  to run "FIX_DISK -FIX".
"PArtial"        to run "FIX_DISK -FIX -PARTIAL"
"FAst"           to run "FIX_DISK -FIX -FAST"

"HELP"           to see this screen.
"QUIT"           to return to the Main Menu.
Run "FIX_DISK -FIX" on PDEV '6062 <OSGRP1>? SK
Run "FIX_DISK -FIX" on PDEV '6164 <OSGRP2>? SK
Run "FIX_DISK -FIX" on PDEV '6160 <OSGRP3>? SK
Run "FIX_DISK -FIX" on PDEV '4162 <OSGRP4>? SK
Run "FIX_DISK -FIX" on PDEV '6362 <CHUM1>? FULL
Run "FIX_DISK -FIX" on PDEV '5120 <CHUM2>? SK
Run "FIX_DISK -FIX" on PDEV '6122 <CHUM3>? SK
Run "FIX_DISK -FIX" on PDEV '5527 <EAF1>? SK

                              FIX_DISK
LDEV  PDEV    NAME       RECOMMEND  ACTUAL        COMMENTS
----  ------  --------   ---------  ------        --------
   0    6060  <OSGRP0>   full       none          COMDEV NC
   1    6062  <OSGRP1>   none       none          Robust
   2    6164  <OSGRP2>   none       none          Robust
   3    6160  <OSGRP3>   none       none
   4    4162  <OSGRP4>   none       none
   5    6362  <CHUM1>    full       full          NC
   6    5120  <CHUM2>    full       none          NC
   7    6122  <CHUM3>    none       none
  10    5527  <EAF1>     none       none          Robust
Are these FIX_DISK recommendations satisfactory? y
Enter pathname of working directory (default="<0>SYSTEM_DEBUG*>CRASH"):
```

Other RAS Features

6

Introduction

The term *RAS*, as its name implies, incorporates an array of hardware and software products designed to make Prime equipment not only much less likely to fail, but also easier to fix and faster to bring back up. This chapter discusses some of the other RAS features, including

- Disk mirroring
- Spin down
- Robust partitions
- VCP–V Maintenance Processor (Quick Boot mode)

Disk Mirroring

Disk mirroring increases system availability by making it possible to process with pairs of logical disks. These logical disks are equivalent: if one fails, the other is an exact duplicate and is available for use. The transition to the use of the duplicate disk is automatic.

This is especially useful in a heavy-usage database environment where data access is critical. Prime presently estimates that the Mean Time to Data Outage (MTDO), the average time between loss of physical data, increases on an SMD disk from approximately 30,000 hours on an unmirrored disk to 2.7 million hours on a mirrored disk, and that MTDO on a SCSI disk increases from approximately 150,000 hours to 67 million hours using mirroring.

Disk mirroring allows PRIMOS to

- Mirror partitions on different disk drive units (which thus have different disk drive unit numbers) of the same disk controller
- Mirror partitions on disk drive units that have the same disk drive unit numbers but are on different disk controllers

- Mirror partitions on different disk drive units of different disk controllers
- Continue processing on one partition if the other fails
- Copy a partition as a background process while the partition pair is being mirrored (a catch-up copy)

When you mirror partitions, all records written to a partition, called the **primary partition**, are also written to another partition, called the **secondary partition**. Thus, all write operations are duplicated.

Reading of records is not duplicated. Reading is split so that the records in the first half of the partition are read from the primary partition and the records in the second half are read from the secondary partition. This process reduces the average time it takes to read a record (compared to reading all records from one of the partitions) because the average seek time is reduced.

Mirroring Requirements

The requirements for disk mirroring are as follows:

- Both the primary partition and the secondary partition must be in disk drives associated with downloaded intelligent disk controllers that are capable of dynamic badspot handling; that is, the Model 6580 (IDC1) for SMD disks and the Model 7210 SCSI disk/tape controller downloaded with ICOP+ for SCSI disks.

Note If the primary partition and the secondary partition are on different disk controllers, the controller is eliminated as a common point of failure. In addition, performance improves when you are mirroring partitions on different disk controllers.

- The two partitions must be Rev. 21.0 or later partitions.
- The two partitions must be in Dynamic Badspot Handling (–IC) mode if they are associated with a Model 6580 (IDC1) disk controller or they must be on a Model 7210 SCSI disk/tape controller downloaded with ICOP+ so that Dynamic Badspot Handling can take place on them.
- The two partitions must be on the same model disk; that is, they must be on the same physical disk, or spindle, types.
- The two partitions must be identical with respect to size (number of surfaces) and position (starting surface number) on the spindles. (They thus will have identical basic pdevs before the pdev is modified for disk drive unit number and disk controller address.)

- A maximum of 128 partitions can be mirrored at one time; that is, there can be a maximum of 64 pairs of mirrored partitions.
- Assigned partitions cannot be mirrored.
- It is not possible to mirror both the paging portion and the file system portion of a split partition. Generally this means that only the paging portion can be mirrored because you start the paging mirror at system startup by a configuration directive. In addition, if the paging portions of two partitions are mirrored, it is *not* possible to add the file system portion of either partition with the ADDISK command.
- One or more of the following directives must be in the configuration file. (See the section Configuration Directives for Mirroring below.)

```
MIRROR
COMDVM pdev
PAGINM pdev1 [. . . pdev8]
```

- You can mirror robust partitions; however, the type of partition that results (either standard or robust) depends on what the primary partition is. See Mirroring and Robust Partitions in Chapter 7 of the *Operator's Guide to File System Maintenance* for more information.

Since the catch-up copy facility in the mirroring process makes a physical copy of the primary partition that you want to mirror to the secondary partition, the resulting secondary partition becomes the same revision (either Rev. 21.0, Rev. 22.0, or Rev. 22.1) and the same type of partition (standard or robust) as the primary partition.

Performance

If you mirror one partition of a spindle, you should mirror every partition on that spindle for best performance. In addition, configure each of the two partitions of a mirrored pair on different disk controllers, if possible. This provides better reliability and performance because if mirrored partitions, and thus their spindles, are associated with a single controller, the controller can be a single failure point for both partitions.

Caution You can mirror only some of the logical partitions on a spindle. However, doing this will have a negative performance impact if there is much activity on the nonmirrored partitions. It is thus strongly recommended that you mirror all the partitions on a spindle if you plan to mirror any partitions on that spindle.

For more information on mirroring, see the *Operator's Guide to File System Maintenance.*

SPIN_DOWN Command

SPIN_DOWN is a supervisor terminal command that stops (spins down) a disk. The principal use for this command is to take offline a malfunctioning disk until it can be repaired or replaced.

Issue the SPIN_DOWN command to stop a disk drive when you notice it malfunctioning. SPIN_DOWN is presently used with SCSI disk drives in a Model 75500-6PK device module that are controlled by a Model 7210 (SDTC) disk controller using ICOP+.

```
SPIN_DOWN pdev
```

pdev is the physical device number (in octal) of the disk drive. You can only spin down a disk that is not in use; you cannot spin down a physical disk containing COMDEV (unless COMDEV is mirrored), a paging, added, or assigned partition, or a partition activated for crash dump to disk.

Following a successful spindown, an amber LED light is displayed on the specified disk drive in the Model 75500-6PK device module, indicating that the disk has spun down. After successfully issuing the SPIN_DOWN command, turn off the power switch located on the front of the disk drive.

If you attempt to spin down a disk that is either already spun down or nonexistent, SPIN_DOWN performs no operation but returns an OK prompt. If you attempt to spin down a disk for which spindown is not permitted, the system returns the following message:

```
Physical device number pdev conflicts with an active file
system partition, assigned disk, or paging disk.  Please
verify the physical device number and check for
conflicts.
Physical device number pdev is:
CONTROLLER ADDRESS: nn
UNIT NUMBER: n
```

The Controller address *nn* is either 22, 23, 24, 25, 26, 27, 45, or 46 (octal) and the unit number *n* is an octal number 0 through 7 (inclusive), as shown on the front of the disk drive itself. This message is also displayed if the disk contains an activated partition for crash dump to disk.

Robust Partitions

A **robust partition** is a type of disk partition introduced at Rev. 22.1. Robust partitions reduce the time that it takes to recover from a system halt. All files and segment directory subfiles on a robust partition are physically stored as CAM files. The CAM file structure allows the –FAST option of FIX_DISK to quickly check the extent map and verify the physical structure of the CAM file. This same capability is not available on a **standard** (non-robust) partition.

Another major advantage of robust partitions is that PRIMOS advises you whenever the result of a system halt requires you to run FIX_DISK on a partition. PRIMOS cannot require you to run FIX_DISK on a standard partition after a system halt nor can FIX_DISK indicate when it should be run except in the case of an incorrect quota system.

Understanding The Robust Partition File System

The robustness of a robust partition is transparent to nearly all software. Robust partitions introduce a new concept called **logical file typing**. In previous revisions of PRIMOS there were three types of physical files. A file could be a physical SAM file, a physical DAM file, or a physical CAM file. This physical typing determines exactly how the file is strung together to make it an entity. Robust partitions separate the physical file structure from the logical, or application-level, file structure.

Every file that is created on a robust partition is physically organized as a CAM file. This means that every file on a robust partition has an extent map that tells PRIMOS where the actual data records are stored. All of this is transparent to higher levels of software. LD, for example, reports the existence of SAM, DAM, and CAM files on a robust partition. If your application opens a SAM file on a robust partition, it appears to be a SAM file. This is the logical file type and it determines which application-level operations are possible. Underneath the application, however, PRIMOS converts the operations into the proper steps to access the correct data record in the physical CAM file that actually exists.

What Robust Partitions Can Provide

Robust partitions offer several advantages that can significantly reduce the length of time that is required for you to resume normal operations after a system halt. Some of these advantages derive from the robust partition structure. A few of the advantages are based upon the inherent characteristics of CAM files. The purpose of this subsection is to explain the nature of the advantages that robust partitions offer.

Advantages: Advantages of using robust partitions include

- System availability is improved because some halts do not require FIX_DISK to be run and others require only fast FIX_DISK (FIX_DISK -FAST) in place of full FIX_DISK.
- PRIMOS tells you whether or not you must run FIX_DISK on the partition when you use the ADDISK -FORCE command. This saves you the time of running FIX_DISK unnecessarily.

- Robust partitions can improve upon your ability to resume operations after some system halts.
- File deletions and truncations are faster since it is necessary to read only extent maps rather than every data record.
- Writing out full records using the PRIMOS subroutine PRWF$$ is 50% faster.
- Robust partitions offer a faster record access mechanism for some environments.
- Robust partitions offer the most advantage when you have large files or segment directories with large subfiles.

Because the design of robust partitions specifically improves the ability to recover from a system halt, the disk format is less likely to suffer from some types of directory corruption that can occur on a standard partition. Because of the file system structure implemented on a robust partition, fast FIX_DISK can verify the integrity of the user directories. This can greatly reduce the length of time that is required to run FIX_DISK. As a result, you can quickly check the directory structure.

Logical File Types: Robust partitions include a concept called **logical file typing**. All files stored on a robust partition are physically stored as CAM files. For example, although you might open a file with a logical file type of SAM, PRIMOS physically creates the file as a CAM file. This is transparent to all higher levels of software and allows you to move existing applications to a robust partition without modification. This logical-to-physical mapping also allows PRIMOS to more tightly control the file structure on a robust partition, without changing the logical appearance of that file structure.

Because every file and every segment directory subfile on a robust partition is physically stored as a CAM file, there is less likelihood that a file will be damaged by a corrupt record header chain. Since CAM file data records are not chained through the record headers, corruption of a data record header does not cause the remainder of the file to be lost. Also, the extent map mechanism means that fast FIX_DISK is able to detect file structure corruption very quickly by checking the extent map.

Record Errors: The introduction of robust partitions offers a new method of responding to a corrupted data record. On a standard disk partition, a pointer mismatch (e$ptrm) error occurs if the record header chaining is corrupt. This error is fatal to the application and can be corrected only by running FIX_DISK. This same error can occur on a robust partition, but PRIMOS reports it as an uninitialized block (e$zero) and re-initializes the data record header, filling the data record with nulls. Although, there is now a null data record, the file can still be accessed without requiring you to run FIX_DISK to correct the error. If the application detects this error, it can take its own corrective action, which may include a data-management rollback procedure to correct the data integrity of the database. (Prime DBMS, Prime ORACLE™, MIDASPLUS™, and PRISAM™

all treat the uninitialized block as a fatal error; the application fails and returns to PRIMOS.)

Record Access: Robust partitions also offer a faster record access mechanism for some environments. Typically, a large CAM file provides faster data access than a large DAM file. This is noticeable when you have multiple users accessing the same file simultaneously and when the file is larger than 512 disk records (1 megabyte). This faster access can be an advantage if your application does not already use CAM files.

File Deletion: Deleting a large file is always significantly faster on a robust partition than on a standard partition. Two files cannot claim the same data record on a robust partition. On a standard partition, PRIMOS must verify that all of the records within the file actually belong to the file. Verification is not necessary on a robust partition.

Restrictions on the Use of Robust Partitions

There are a few restrictions on when you can use a robust partition.

Shutdowns: Because the ADDISK command checks a robust partition, you must run FIX_DISK if the partition was not cleanly shut down. This can be inconvenient if you do not regularly run FIX_DISK after every system halt. Forcing you to run FIX_DISK in this case, however, provides better assurance of file structure integrity.

Note Be aware that, following a halt, you should add robust partitions with the -FORCE option. If the disk is clean, then using the -FORCE option has no effect upon the disk. If the disk is not clean, then using -FORCE has the effect of ADDISK -PROTECT; that is, the disk is added in read-only mode, but it has been added nonetheless. This way, FS_RECOVER can analyze the disk.

In order to reduce the time necessary to recover from a system halt, you need to use the -FAST option of FIX_DISK (fast FIX_DISK). Fast FIX_DISK checks the directory structure and CAM file extent maps only.

Booting: The boot procedure can only access files stored as SAM files. All files on a robust partition are stored as CAM files. The PRIMOS boot procedure cannot access any file stored on a robust partition. This means that you should not convert your command partition to a robust partition. This also means that you cannot use a robust partition as an alternate boot device.

Disk Space Required: Sometimes a robust partition requires more disk space than a standard partition to store the same amount of data. A SAM file on a standard partition contains only data records. When you move the SAM file to a robust partition, the file requires an additional record for the extent map. This means that a file that was stored as a single-record SAM file on a standard partition becomes a two-record CAM file on a robust partition. You must allow enough additional space for the conversion.

The amount of additional space required depends on the file type. ACLs and ACATs do not require additional space. Each SAM file requires one additional disk record for an extent map. DAM files might not require any additional disk space. CAM files do not require any additional disk space. Remember however, that CAM files allocate data records in blocks called extents. There are occasions when PRIMOS appends unused data records to the end of a CAM file. These records occupy additional disk space. Generally, you can minimize all of these considerations by placing only large database files on a robust partition.

The size of a segment directory is not significant to the discussion about robust partitions. The segment directory structure is itself not changed. Size considerations instead focus on the size of the individual subfiles within a segment directory.

Directories: The directory structure itself is changed on a robust partition. A Rev. 22.1 standard partition uses a hashed directory structure. A robust partition uses a linear directory structure. Entries should consist of only a small number of large files in each directory on the robust partition to maintain the directory search time.

Sectoring: Robust partitions do not support reverse sectoring. Whenever you convert a partition to the robust format, sectoring is automatically set to forward.

Accessing Rev. 22.1 Format Disks: Rev. 22.1 and later disks are a new format. To locally access either standard or robust Rev. 22.1 format partitions, you must be running Rev. 22.1 or later PRIMOS. You can access Rev. 22.1 format partitions remotely on a network, however, such as through PRIMENET. This means that you should not reboot your local system to an earlier version of PRIMOS. Insure that all of the PRIMOS upgrade has been successfully completed before you begin the conversion to robust partitions.

Null Records: Finally, understand that PRIMOS can insert a null-filled data record into your database as a result of a system halt. This rare event would cause a fatal error on a standard partition.

Understanding the Concept of Recoverability

You should understand one essential concept before deciding whether or not to use robust partitions. Robust partitions improve **recoverability**, or your ability to resume operations after a system halt. Similar to FIX_DISK, robust partitions do not offer any protection against disk corruption; they offer only an improvement in your ability to detect disk corruption. This is one reason why it is important to use robust partitions only for files that an application-level data verification routine can properly check.

In many cases, you can find a degree of data integrity corruption by running full FIX_DISK. This is not the reason for running FIX_DISK; FIX_DISK was designed to check file system integrity and does not check data integrity.

Nevertheless, many locations rely on FIX_DISK to indicate whether the data integrity of a file has been compromised. This appears to work on a standard partition because full FIX_DISK detects corrupted data record headers. The assumption is made that if the data record headers are not corrupt, the data records are probably not corrupt either. Some of this ability to detect data corruption is lost when fast FIX_DISK (FIX_DISK –FAST) is used on a robust partition because fast FIX_DISK will not read any data record headers and therefore cannot verify the validity of the data record headers. Used properly, fast FIX_DISK offers the advantage of rapidly repairing your partitions but this can only be an advantage when you have an alternative process in place to verify data record integrity.

Robust partitions offer help in minimizing the inconvenience caused by a hardware failure, which can cause data loss. Recommendations for Using FIX_DISK, in Chapter 5 of this manual, summarize types of system halts and the necessary action to properly respond to those halts. These recommendations are applicable to systems using either standard or robust partitions. You can see that robust partitions offer the advantage of effectively utilizing the –FAST option of FIX_DISK (fast FIX_DISK) for those system halts that are trapped and processed through the PRIMOS slow-halt mechanism.

Understanding the –FAST Option of FIX_DISK

The –FAST option of FIX_DISK (fast FIX_DISK) allows the System Operator to quickly verify the integrity of the file structure. FIX_DISK does not provide any check on the integrity of the data contained within the files. Only a utility that understands the data management application can verify the data within a file.

This section explains the functionality of fast FIX_DISK on a robust partition and then briefly compares the functionality when you run fast FIX_DISK on a standard partition.

Both robust partitions and standard (nonrobust) partitions support the –FAST option. The –FAST option is less useful, however, on a standard partition because it can be used only if the partition was cleanly shut down.

FIX_DISK Action: FIX_DISK acts identically on the file system directory structure on both standard and robust partitions whether you enable the –FAST option or not. FIX_DISK checks the entire directory structure and verifies the integrity of every directory and segment directory entry. Use of fast FIX_DISK, however, limits the degree of verification on files within directories.

Use of fast FIX_DISK also limits the degree of verification of subfiles within a segment directory. This is an important technical detail. A segment directory is a special type of directory structure that contains a set of subfiles. All of the data is contained within the subfiles. Like any directory, there is a directory header that contains all of the information about the contents of that directory. A segment directory can contain many subfiles. Both full and fast FIX_DISK verify every directory header and every segment directory header. Use of the –FAST option allows FIX_DISK to provide directory structure verification more quickly.

When fast FIX_DISK completes without finding any mismatches, it has checked that the directory structure is intact and that the correct number of disk records have been allocated for the data files. You cannot be sure, however, that the data records actually have the correct data within them. To verify the data record content, you must run a verification routine of a data management package on any data management files.

Full FIX_DISK provides one additional level of verification that fast FIX_DISK does not provide. Full FIX_DISK reads every data record header within every file. Full FIX_DISK then verifies that the record header is properly initialized. Do not, however, rely on FIX_DISK as an indicator of the integrity of the data in a disk record.

Full and Fast FIX_DISK Comparison: To better understand the benefits robust partitions offer, we must distinguish between CAM file functionality and robust partition functionality. Full FIX_DISK processes a CAM file identically whether it is on a robust partition or on a standard partition. The operation of fast FIX_DISK depends whether the CAM file is on a standard partition or on a robust partition. On a standard partition, fast FIX_DISK verifies the last two data records within every CAM file extent. On a robust partition, fast FIX_DISK verifies only the extent map.

On a robust partition, all files are automatically stored as CAM files. Through the logical file typing mechanism, the physical file type is transparent to all higher levels of software. It is the physical typing, however, that is important to FIX_DISK.

In order for FIX_DISK to know which disk records a physical SAM file on a standard partition uses, FIX_DISK must check every record because SAM files do not have an index or an extent map. When FIX_DISK encounters a SAM file, it must read a record header, find the pointer to the following record, and then repeat the process. Thus, both full FIX_DISK and fast FIX_DISK must read through the entire SAM file. PRIMOS physically stores all SAM files as CAM files on a robust partition and, thus, FIX_DISK needs to check only the extent map.

In conclusion, the –FAST option is available on both standard and robust partitions. Fast FIX_DISK verifies the full directory structure on both standard and robust partitions. You can run fast FIX_DISK on a standard partition only when the partition has been cleanly shut down. If you need to run FIX_DISK on a regular basis, robust partitions can reduce the time required.

For additional information about robust partitions, see the *Operator's Guide to File System Maintenance*.

VCP–V Maintenance Processor

The VCP–V is the maintenance processor (MP) for the 2850, 2950, 4050, 4150, 6150, 6350, 6450, 6550 and the 6650 systems. Changes have been made to increase the availability and improve the serviceability of these systems.

Quick Boot

Usually when a system is starting up, it is fully functional and does not require the internal integrity tests that are automatically performed at startup time. Prime has addressed this issue with a new mode of system startup called **Quick Boot**. In Quick Boot mode, the MP reduces the time it takes to start a system from power-up by bypassing most of the reliability tests.

Quick Boot implements:

- A new boot option, called Quick Boot, that decreases the time it takes to boot a system from power-on state.
- A new abbreviated boot code that is read from the floppy disk each time the system is booted, thereby reducing re-boot time.

In Quick Boot mode, the typical elapsed time from power-up to the printing of the disk boot header has decreased from 8 – 12 minutes to 2 – 3 minutes. The message

```
WRN101: Quick Boot option enabled.   Bypassing CPU integrity tests.
```

is printed on the supervisor terminal during power-up, and is also printed when the command is entered that enables Quick Boot mode.

The new boot code, identified as QBOOT on the floppy disk, loads and executes faster than the standard boot code, which is now identified on the floppy disk as CPBOOT. You can load or run either of these programs, regardless of the current boot mode, when you specify the MP commands LOADTM or RUNTM.

Note QBOOT, unlike CPBOOT, can only boot from disk controllers with a device address of '26 or '27 and a unit number of 0, 1, 2 or 3, or from a tape unit number 0. In addition, be aware that QBOOT does not presently have the resilience of CPBOOT, so that booting from a non-existent or defective controller, or with invalid sense switch or data switch settings, causes a program hang without any error indications.

BOOTQ Command

The Quick Boot mode option is enabled by the new MP command BOOTQ, and is disabled by the command BOOTP. The BOOTT command has been eliminated.

Issuing the BOOTQ command initiates the following actions:

1. The MP determines if the functional microcode and the decode net have been loaded. If not, they are loaded and the MP performs a SYSCLR. If the microcode and the decode net have already been loaded, the MP performs a SYSCLR (if it has not yet been performed).
2. The MP loads QBOOT code into main memory and starts the CPU.

The BOOTP command functions as it has in the past, and has the following effect:

1. The MP begins by testing the Control Store on the CPU, and then it loads and runs the SYSVFY microdiagnostics.
2. The microcode and decode net are loaded, and a SYSCLR is performed.
3. The CPBOOT program is loaded into main memory and the CPU is started.

On the BOOT command, the MP loads either CPBOOT or QBOOT, depending on the boot mode, into main memory and then starts the CPU.

New Switch Settings

In addition to setting the mode of power-up boot, the BOOTQ or BOOTP commands can now change the default power-up boot sense switch and data switch settings. This means that you can boot the system on power-up from disks other than device address '26, unit number 0.

Adding a sense switch argument or sense switch and data switch arguments to either the BOOTP or the BOOTQ command defines new switch settings to be used during the power-up boot. Issuing the BOOTQ or the BOOTP command without arguments boots the CPU with the same switch settings that are defined for the power-up boot. The BOOT command, without arguments, defaults unspecified sense switch and data switch settings to 0.

A system will fail to boot from disk if the Quick Boot option is enabled and the data switch setting is not zero. In the QBOOT code, a data switch setting other than zero specifies loading from a diagnostic test board used in manufacturing. For example, the following would cause the CPU to hang:

```
CP> BOOTP 14114 12000
CP> BOOTQ
```

To remedy the situation, issue the following sequence:

```
{ESC} {ESC}
CP> STOP
CP> BOOTQ 14114 0
```

Either the BOOTP or BOOTQ command can be entered at the CP> prompt while the CPU is running. This will allow the boot mode or the default boot switch settings to be changed at any time. The operation will abort, with an error message, after the mode, the sense switch and the data switch settings have been updated.

Microdiagnostics

Be aware that PRIMOS is not always the best diagnostic for determining system status; hardware failures can be quite subtle in the ways in which they manifest themselves. It is true that most component parts of a system must function correctly in order to boot the operating system, but there are many parts of the CPU which were designed for specific functions or conditions. Some of these components are not used in the boot or during normal operation.

The microdiagnostics were designed to test each block of logic on the CPU. While the successful completion of microdiagnostics does not imply that system will boot, it can identify problems that may go undetected until application program failures are discovered. Use caution when deciding whether to run microdiagnostics and, if you do not run diagnostics by default, stay alert for possible consequences, especially if you change your CPU hardware, or if you encounter unexpected errors.

Index

Index

Symbols

! character, before commands in FS_RECOVER, 5-13

A

ACL protection
- FS_RECOVER, 5-8
- SYSTEM_DEBUG* directory, 5-8

Added disks
- crash dump disk restriction, 4-6
- SPIN_DOWN restriction, 6-4

ADDISK command
- –FORCE option, 3-19, 6-7
- robust partition modification for, 6-5
- robust partitions, force FIX_DISK, 6-7

ASR
- described, 1-5
- guidelines, 2-1
- recommendations for, 1-6

Assigned disks, SPIN_DOWN restriction, 6-4

Auto Restart, 2-4

Automated System Recovery. *See* ASR

AUTOPSY
- run file, 5-8
- search rules file, 5-8

B

Boot sense switch settings, 6-12

Booting
- halts and hangs during, 3-8
- robust partitions and, 6-7

BOOTQ command, 6-11

Buffering. *See* Locate buffers

C

CAM files
- deletion of on robust partition, 6-7
- fast FIX_DISK on, 6-10
- following system crash, 5-7
- speed of data access, 6-7

Cautions, mirroring part of physical disk, 6-3

CDD command
- –INFO option, 4-3
  - creating crash dump disks, 4-3
- –RECOVER_DUMP option, 4-7
- –DUMP_SIZE_TABLE option, 4-5

Clean partitions
- assessing, 5-23
- defined, 1-3

Cold starts
- following crash, 5-1
- halts, use of, 3-18
- mode DUAL/UNI, changing, 3-18
- pausing to invoke FS_RECOVER, 5-9
- required after RFS, 5-3
- risk due to, after halt, 3-2

COMDEV
- damaged by crash, 5-7
- mirroring, 6-4
- repairing using FIX_DISK, 5-7
- SPIN_DOWN restriction, 6-4
- SYSTEM_DEBUG*, 5-8

Command line, within FS_RECOVER, 5-13

Commands
- *See also* MP commands; VCP commands
- exclamation point character, 5-13
- executing within FS_RECOVER, 5-13
- INIT_RECOVER, –AUTO_ANALYSIS option, 5-22

Commands (VCP). *See* VCP commands

COMO files
- crash recovery session, 5-17
- FIX_DISK, 5-6
- FS_RECOVER, 5-16, 5-17

CONFIG directives, required for mirroring, 6-3

Configuring, FS_RECOVER segments, 5-9

Control codes, during FS_RECOVER, 5-13

Controllers, disk, mirroring performance, 6-3

CPBOOT command, 6-11

CPL
- for automated FIX_DISK, 5-16
- for running FIX_DISK, 5-6

CPL files
- FIX_DISK_MONITOR.CPL, 5-16
- FS_RECOVER.INSTALL.CPL, 5-8
- INIT_RECOVER.CPL, 5-9
- RUN_FIX_DISK.CPL, 5-16

Crash dump, partial, 4-4

Crash dump disk
- activating, 4-6
- controller support, 4-6
- creating, 4-2
- disk type, 4-6
- file system portion, 4-6
- FS_RECOVER access, 4-7
- MAKE requirements, 4-6
- paging restriction, 4-6
- recovering dump from, 4-8

Crash dump disks
- sizes, optimal, 4-5
- SPIN_DOWN restriction, 6-4

Crash dump to disk
- *See also* CDD
- activating, 4-6
- analyzing, 4-7
- defined, 1-2
- disk too small, 4-7

Crash dump to disk (Continued)
- disk type restrictions, 4-6
- MAPS information, 4-2
- performing, 4-7
- recommendations, 4-8

Crash dump to tape
- disk dump failure, 4-7
- MAPS information, 4-2
- multi-reel, 5-15
- tape drive checking, 5-14
- tape reading, 5-14, 5-15

Crash dumps
- allocating records for, 5-10
- analyzing, 5-15
- calculating record requirements, 5-10
- creating files, 5-15
- disk space for analysis, 5-10
- file pathname, 5-16
- full, 5-10
- partial, 5-10
- performing, 5-1

Crash recovery tools
- FS_RECOVER, 5-4
- RFS, 5-2

D

Damaged partitions, assessing, 5-23
Data integrity, following system crash, 5-3
Data sense switch settings, 6-12
Directives, configuration, required for mirroring, 6-3
Directories
- segment, structure of, 6-9
- segment and FIX_DISK, 6-9

Disk and tape controllers
- 10019 for crash dump disk, 4-6
- 7210 for crash dump disk, 4-6
- 7210 SPIN_DOWN support, 6-4
- 7210 with 75500-6PK disks, 6-4

Disk mirroring, COMDEV, 6-4
DISK_MANAGER program, 5-22
Disks
- crash analysis space, 5-10
- FIX_DISK required, 5-2
- initial state, 5-2
- recovering, 5-1
- shutting down, 5-2

DSW registers, displayed at halts, 3-7
Dynamic segments, FS_RECOVER requirements, 5-9

E

ECCU halts, discussion of, 3-7
ECL, using within FS_RECOVER, 5-14
Errors
- null–filled records, 6-8
- pointer mismatch, 6-6
- uninitialized block, 6-6

F

Fast shutdown, 1-2
File system
- recovering from halts, 1-4
- recovery recommendations, 1-6
- recovery using RAS features, 1-5

File system cache. *See* Locate buffers
File system integrity, 5-1
Files
- *See also* PRIMOS.COMI file
- affected by a crash, 5-7
- crash dump, 5-15, 5-16
- deletion of on standard and robust partitions, 6-7
- logical types, 6-6
- logical typing, 6-5
- logical–to–physical mapping, 6-6
- organization on robust partitions, 6-5
- search rules, 5-8

FIX_DISK
- automated, 5-20, 5-22
- COMO files, 5-6
- deferred, 5-18
- determining if required, 5-2
- file system integrity, 3-10, 3-13, 3-15, 3-18, 3-19
- FS_RECOVER recommendation examples, 5-19
- immediate, 5-18
- manager phantom, 5-22
- monitor, 5-16
- monitor phantom, 5-6
- not required, 5-18

FIX_DISK command
- –FAST option, 6-9
- –COMDEV option, 5-7

FIX_DISK utility
- design of, 6-8
- detecting file structure corruption, 6-6
- fast
  - data record headers, 6-9
  - integrity verification, 6-6
  - operation of, 6-9
  - use of, 6-7
- full, 6-10

Forced shutdown, procedure for, 5-12
Forced shutdown halts
- cold starts, use of, 3-18
- discussion of, 3-5
- messages, 3-4
- recovery procedure, 3-15
- unsuccessful, 3-6

FS_RECOVER, 5-4
- ACL requirements, 5-8
- breaking out of, 5-13
- COMO files, 5-16, 5-17
- Control-P during, 5-13
- crash analysis (example), 5-16
- crash dump file, 5-15, 5-16
- crash dump to disk, 4-7
- crash dump using, 5-5
- crash recovery analysis, 5-15
- data analysis messages, 5-17
- deferred FIX_DISK, 5-18
- defined, 1-3
- directory, 5-8
- disk manager, 5-22
- ECL environment, 5-14
- error messages, 5-11
- executing PRIMOS commands during, 5-13
- file system integrity, 3-10, 3-13, 3-15, 3-18, 3-19
- FIX_DISK not required, 5-18
- forced shutdown following, 5-12
- immediate FIX_DISK, 5-18
- installation errors, 5-11
- installation of, 5-7
- invoking at coldstart, 5-9
- machine state, 5-17
- main menu, 5-12
- options, 5-13
- phantoms, 5-7
- PRIMOS.COMI changes, 5-9
- reading crash dump tapes, 5-15

FS_RECOVER (Continued)
record requirements, 5-8
security, 5-8
segment requirements for, 5-9
stopping, 5-13
tape drive checking, 5-14
tapes reading crash dump, 5-14
using, 5-11
working directory, 5-16

H

Halts
actions for non-ROAM systems, table, 3-13
cold starts, use of, 3-18
defined, 3-1
ECCU, discussion of, 3-7
forced shutdown, 3-4
discussion of, 3-5
messages, 3-4
recovery from, 3-15
unsuccessful, 3-6
handling procedure, 3-1
hangs vs., 2-7
hangs, distinguishing from, 3-2
hardware failures, 3-7
identifying, 3-2
immediate, 3-4
discussion of, 3-7
messages, 3-4
recovery from, 3-15
registers displayed, 3-7
machine checks, 3-7
messages, 3-4, 3-13
procedures for, 2-7
recovery, 6-6
under PRIMOS, 3-11
while booting, 3-8
ROAM-based products and, 3-13
robust partitions and, 6-9
symptoms of, 3-3
trapped, 3-4
discussion of, 3-6
messages, 3-4
recovery from, 3-15
types of, 3-4
table, 3-5
warm starts, use of, 3-16

Halts and hangs. *See* Halts; Hangs; System crashes
Hangs
defined, 3-1
halts vs., 2-7
halts, distinguishing from, 3-2
handling procedure, 3-1
identifying, 3-2
PRIMOS, recovery from, 3-9
procedures for, 2-7
recovery
procedure, 3-10, 3-12
under PRIMOS, 3-9
while booting, 3-8
symptoms of, 3-3

I

ICOP+ disk controller mode, SPIN_DOWN support, 6-4
Immediate halts
discussion of, 3-7
recovery procedure, 3-15
warm starts, use of, 3-16
INIT_RECOVER.CPL, 5-9
–AUTO_ANALYSIS option, 5-22
–PAUSE option, 5-9
defined, 1-3
Installation, FS_RECOVER, 5-4, 5-7, 5-8

L

Locate buffers
defined, 1-4
flushing, 5-3
Log book, 5-17
Logical file type, 6-6

M

Maintenance Processor
entering, 5-2
microcode, 2-2
Maintenance Processor microcode. *See* Microcode
MAKE command
–AC and –IC options, 4-6
crash dump disk, 4-6
MAPS directory, 4-2, 5-8
MASTER CLEAR button, 3-2, 3-10
Memory, determining size of, 5-10
Messages
forced shutdown halts, 3-5
halts, displayed at, 3-13
immediate halts, 3-7
trapped halt, 3-6
Microcode, 2-2
Microdiagnostics, 6-13
MIRROR_OFF command, 5-21
Mirroring
configuration directives, 6-3
directives, configuration, 6-3
paging partitions, 6-3
partial disk, caution on, 6-3
partitions
maximum number of, 6-3
primary and secondary, 6-2
performance of, 6-3
purpose of, 6-1
requirements for, 6-2
Mirrors, breaking, 5-21
Modes, DUAL/UNI, cold start, 3-18
MP
actions with ASR, 2-1
VCP–V, 6-11
MP commands
BOOTQ, 6-11
CPBOOT, 6-11
LOADTM, 6-11
RUN 600, 2-1
RUNTM, 6-11

P

Paging, partition, mirroring, 6-3
Paging partitions
SPIN_DOWN restriction, 6-4
using as crash dump disk, 4-6
PARTIAL_TAPEDUMP, VCP command
halt recovery, 3-11
partial tape dump defined, 4-1
Partitions
clean, 1-3, 5-23
damaged, 5-23
errors on, 6-6
mirroring, maximum number for, 6-3
paging, mirroring of, 6-3
primary, 6-2
robust. *See* Robust partitions

Partitions (Continued)
secondary, 6-2
Performance considerations, crash dumps, 4-1
Phantoms
FIX_DISK manager, 5-22
FIX_DISK monitor, 5-6
for automated FIX_DISK, 5-6, 5-21
for FS_RECOVER, 5-7
PRIMOS
booting, halt and hang recovery, 3-8
halt recovery, 3-11
procedure for, 3-14
halts and hangs, identifying, 3-2
hang recovery, 3-9
procedure for, 3-9
mirroring actions, 6-1
PRIMOS revision, FS_RECOVER support, 5-4
PRIMOS.COMI file
execution, 5-12
FS_RECOVER, 5-9
INIT_RECOVER -PAUSE, 5-9
pausing, 5-9, 5-12
SYS_RECOVER.CPL including, 2-3

Q

QBOOT command, 6-11
Quick Boot mode, 1-4, 6-11
defined, 1-4
hang while booting, 3-8

R

RAS, defined, 1-1
Records
FS_RECOVER requirements, 5-8
required for full crash dump, 5-10
required for partial crash dump, 5-10
Recoverability, understanding the concept, 6-8
Recovery, file system. *See* File system
Recovery, file system, 1-5
Registers, DSW, displayed at halt, 3-7
Resident Forced Shutdown. *See* RFS
RFS, 5-2
defined, 1-2
initial disk state, 5-2
invoking, 5-2
messages, 5-3
use of, 3-15, 3-18
warm starts and, 3-16
ROAM-based products
cold starts, use of, 3-18
halt recovery, 3-13
warm starts and, 3-16
Robust partitions
access to, 6-8
adding, 3-19
advantages of, 6-5
boot procedure, 6-7
defined, 6-4
directory structure on, 6-8
file organization, 6-5
halts and fast FIX_DISK, 6-9
logical file typing, 6-5
restrictions on use of, 6-7
sectoring on, 6-8
space needed for files on, 6-7
RUN 660, VCP command, 2-1
RUN 661, VCP command, 4-7
RUN 662, VCP command, 5-2

S

SAM files, operation of FIX_DISK on, 6-10
SCSI disks
in 75500-6PK device module, 6-4
malfunctioning, 6-4
spin down, 6-4
Search rules
AUTOPSY, 5-8
COMMAND$, 5-8
ENTRY$, 5-8
FS_RECOVER changes for, 5-8
MAPS, 5-8
Security considerations, FS_RECOVER, 5-8
Segment directories (segdir), 6-9
Segments, FS_RECOVER requirements, 5-9
Sense switch settings, 6-12
Shutdown
See also RFS
fast, 1-2
SPIN_DOWN command, 6-4
Splitting disks, for crash dump disk, 4-2
STATUS SYSTEM command, 5-10
STOP command, 5-2
Switch settings, changing with Boot commands, 6-12
SYS_RECOVER.CPL file, 2-3
System
availability, 6-1
halts and hangs, identifying, 3-2
non-ROAM, halt actions, table, 3-13
System Administrator
maintaining log book, 5-17
segment requirements, 5-9
System crashes
See also Halts; Hangs
analyzing data integrity, 5-17
crash dump to disk, 4-1, 4-7
determining machine state, 5-17
forced shutdown, 5-12
recovery recommendations, 5-11
recovery tools, 5-1
SYSTEM users, 5-9
SYSTEM_DEBUG* directory
ACL requirements, 5-8
CRASH, 4-2, 5-8, 5-10, 5-15, 5-16
in COMMAND$ search rules, 5-8
INIT_RECOVER.CPL, 5-9
installing FS_RECOVER, 5-8
SYSTEM_RECOVER command
–AA option, 5-12
default mode, 2-3
defined, 1-2
non–default mode, 2-11
options, 2-11

T

Tape drives
assigning for FS_RECOVER, 5-14
error messages, 5-15
Tape dumps
halts, during, 3-11
types of, 4-1
TAPEDUMP, VCP command, full tape dump, 4-1
Tapes
crash dump using, 5-14
FS_RECOVER installation, 5-8
multi-reel crash dump, 5-15
Trapped halts, 3-4
discussion of, 3-6

Trapped halts (Continued)
recovery procedure, 3-15
warm starts, use of, 3-16

U

Users, SYSTEM, 5-9

V

VCP commands
crash dump to disk, 4-7
DSW, displaying registers, 3-7
PARTIAL_TAPEDUMP
halt recovery, 3-11
partial dump, 4-1
QBOOT, 6-11
Resident Forced Shutdown, 5-2
RUN 660, 2-1
RUN 661, 4-7
RUN 662, 5-2
STOP, 5-2
hang recovery procedure, 3-9
TAPEDUMP
full dump, 4-1
halt recovery, 3-11

W

Warm starts
cold starts, use of, 3-18
halts, use of, 3-16
procedures for, 3-16, 3-17
RFS, use with, 3-16
risk due to, after halt, 3-2
ROAM-based products and, 3-16