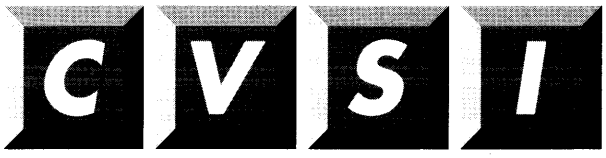

CVSI, Inc.
4G Crosby Drive
Bedford, MA 01730

# System Administrator's Guide, Volume I: System Configuration

**Revision 23.0**

**DOC10131-3LA**

# System Administrator's Guide, Volume I: System Configuration

*Third Edition*

**Roberta C. King**

*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 23.0 (Rev. 23.0).*

Prime Computer, Inc., Prime Park, Natick, MA 01760

## Printing History

First Edition (DOC10131-1LA) July 1987 for Revision 21.0
Second Edition (DOC10131-2LA) September 1988 for Revision 22.0
Release Notes (RLN10131-21A) June 1989 for Revision 22.1
Third Edition (DOC10131-3LA) June 1990 for Revision 23.0

## Credits

Project Support: Steve Calapai, George Gove
Editorial: Judith Goodman
Illustration: Carol Smith, Elizabeth Wahle
Production: Judy Gordon

**WARNING**

This equipment generates and uses radio frequency energy and if not installed and used properly, i.e., in strict accordance with the instructions manual, may cause harmful interference to radio communications. It has been tested and found to comply with the limits for a Class A computing device pursuant to Subpart J of Part 15 of FCC rules, which are designed to provide reasonable protection against such interference when operated in a commercial environment.

Operation of this equipment in a residential area is likely to cause interference, in which case the user at his own expense will be required to take whatever measures may be required to correct the interference.

This equipment has been tested and found to comply with the limits for VDE 0871 Class A computing devices. Any modifications or substitutions to this equipment (particularly the cables) specified in this manual may invalidate this compliance.

This product meets all the requirements of CSA standard C 108.8-M 1983 for a Class A digital apparatus.

Ce produit est conforme aux exigences de la norme CSA - C108.8M, 1983 pour les appareils digitaux de classe A.

If there are any questions, please contact your Prime Field Office.

# CONTENTS

# ABOUT THIS BOOK

The *System Administrator's Guide, Volume I: System Configuration*, is the first volume of the System Administrator's Guide set. It helps you plan your configuration, and also mentions installation issues and the responsibilities of the System Administrator. If you have administrative responsibility for a Prime system, this book is intended for you. It contains a master dictionary of all configuration directives, an overview of all the PRIMOS® directories and files, and a template for the PRIMOS.COMI file. Other administrative functions are described in the remaining books in the set:

- *System Administrator's Guide, Volume II: Communication Lines and Controllers* (DOC10132-2LA) provides definitions, instructions, and examples of the directives and commands necessary to configure communication lines.
- *System Administrator's Guide, Volume III: System Access and Security* (DOC10133-3LA) documents all the security features available on the operating system, including ACLs, EDIT_PROFILE, and the Security Audit facility. It also describes environmental factors and orderly procedures necessary to maintain the security of terminals, peripherals, and storage media.

You are expected to have some familiarity with Prime systems before reading the volumes of the System Administrator's Guide. If you are not familiar with the PRIMOS operating system, read the *PRIMOS User's Guide* (DOC4130-5LA), which explains Prime's file management system and describes essential commands and utilities.

## ORGANIZATION OF THIS BOOK

This book contains nine chapters and one appendix, as described below.

**Chapter 1, Overview:** An explanation of the responsibilities of the System Administrator, lists of the the directories containing PRIMOS software, and lists of servers and other system services.

**Chapter 2, Creating and Allocating Disk Space:** An explanation of the allocation of disk space, both for file storage and for paging, on all disk systems supported by Prime equipment. This chapter also explains how to control tape drive assignment.

**Chapter 3, The PRIMOS Search Rules Facility:** Overview of the PRIMOS search rules facility and a more detailed explanation of the ENTRY$, ATTACH$, and COMMAND$ search rule lists.

**Chapter 4, Planning Your File System Structure at Rev. 23.0:** An explanation of the changes to the PRIMOS file system at Rev. 23.0 and suggestions for utilizing the new features.

**Chapter 5, Initializing System Software:** Information about ensuring the proper initialization of shared system software, including shared segments, shared static-mode libraries, library EPFs, and registered EPFs.

**Chapter 6, Adding and Modifying System Software:** Information about adding and modifying system software.

**Chapter 7, Planning the System Configuration:** An explanation of the several categories of configuration directives, and how to use them.

**Chapter 8, Configuration Directives:** A reference to all the configuration directives.

**Chapter 9, The Scheduler:** An explanation of the purpose of the scheduler and how System Administrators can tune the scheduler at Rev. 23.0.

**Appendix A, Obsolete and Rarely Used Commands and Directives:** Commands and directives that are obsolete or are best avoided, as an aid to updating your system to current usage.

## RELATED DOCUMENTATION

Here is other Prime documentation that will be of help to you:

- *PRIMOS User's Release Document* (DOC10316-1PA). This document updates the *PRIMOS User's Guide* and the *PRIMOS Commands Reference Guide* for Rev. 23.0.
- *Rev. 23.0 Software Installation Guide* (IDR10176-3XA). This book describes how to install Rev. 23.0 PRIMOS, either on a new system, or when upgrading from earlier revisions of PRIMOS. It also contains a sample startup file.
- *ICS User's Guide* (DOC10094-1LA) and its update package for Rev. 21.0 (UPD10094-11A). This book provides detailed information on Prime's Model 2 (ICS2) and Model 3 (ICS3) Intelligent Communications Subsystems.
- *Operator's System Overview* (DOC9298-3LA). This book introduces the series of operator's guides and describes computer-room operation of Prime systems.
- *Operator's Guide to System Monitoring* (DOC9299-3LA). This book describes how to monitor system activity and respond to system and user messages.
- *Operator's Guide to File System Maintenance* (DOC9300-4LA). This book describes the PRIMOS file system and explains how to format partitions with MAKE, how to

run the disk maintenance program FIX_DISK, how to determine physical device numbers, and how to interpret disk error messages.

- *Operator's Guide to Data Backup and Recovery* (DOC10129-1LA). This book describes how to save information on disk or tape and how to restore that information later.
- *Operator's Guide to System Commands* (DOC9304-5LA). This book provides an alphabetical reference of the commands used by system operators.
- *Operator's Guide to the Spooler Subsystem* (DOC9303-3LA). This book describes how to set up, monitor, and control the Spooler subsystem.
- *PRIMOS Commands Reference Guide* (DOC3108-7LA). This book is a detailed reference of user commands.
- *Programmer's Guide to BIND and EPFs* (DOC8691-1LA) and its update package for Rev. 22.0, (UPD8691-11A). Administrators should read this book's information about EPF Libraries.
- *Site Preparation Guide* (DOC5029-3LA) its update packages (UPD5029-31A and UPD5029-32A), and the release notes (RLN5029-3LA). This book provides information for preparing and maintaining a system site.
- *DSM User's Guide* (DOC10061-2LA). This book describes Prime's Distributed Systems Management, and tells how to use DSM to log events, to monitor status, and to control operations, either on a single system or on a network of systems.
- *Remote Job Entry Phase II Guide* (DOC6053-4LA) and its update (UPD6053-41A). This book contains information on setting up the directories required for RJE.
- *Translator Family Software Release Document* (DOC10217-1PA). This book contains information on installation of compilers, assemblers, and editors. It also explains how to change the default values of compiler options.
- *System Architecture Reference Guide* (DOC9473-2LA). This book describes the Prime 50 Series™ architecture. The chapter on process exchange should be helpful when trying to understand the scheduler.
- *Advanced Programmer's Guide I: BIND and EPFs* (DOC10055-2LA). This book provides comprehensive information for programmers regarding BIND and EPFs. System Administrators should find the information on registered EPFs helpful.
- *Advanced Programmer's Guide II: File System* (DOC10056-3LA). This book thoroughly explains the PRIMOS file system structure.
- *Subroutines Reference II: File System* (DOC10081-2LA) This book describes the PRIMOS subroutines that relate to the file system.

## NETWORK DOCUMENTATION AVAILABLE AT REV. 23.0

The following network documentation may be of use:

- *Rev. 23.0 Prime Networks Release Notes* (RLN10252-1LA).
- *PRIMENET Planning and Configuration Guide* (DOC7532-4LA)
- *Programmer's Guide to Prime Networks* (DOC10113-1LA and UPD10113-11A)
- *Operator's Guide to Prime Networks* (DOC10114-1LA and UPD10114-11A)
- *User's Guide to Prime Network Services* (DOC10115-1LA)
- *NTS Planning and Configuration Guide* (DOC10159-1LA and UPD10159-11A)
- *NTS User's Guide* (DOC10117-1LA)
- *WSI300 User's Guide* (DOC10155-1LA and UPD10155-11A)

## NEW FEATURES AT REV. 23.0

Following are the new features at Rev. 23.0 that pertain to material in this guide.

- At Rev. 23.0 the file system has become singly-rooted, meaning that there is one root directory instead of many disk partitions at the top level of the file system. (See Chapter 4.)
- Partitions can now be added to the system with directory names containing as many as 32 characters and they may be added at any level in the file system hierarchy. (See Chapter 4.)
- A new server process, Name Server, is available to replicate the root directory across the file system name space. (See Chapter 4.)
- The ability to tune the Scheduler is expanded and refined. The SET_SCHEDULER_ATTRIBUTES and LIST_SCHEDULER_ATTRIBUTES commands, as well as a new option to the USAGE command, -SCHED, are available to help with scheduler tuning. (See Chapter 9.)
- PRIMOS has a new function that monitors the amount of paging space available on a system and generates warning messages if it is getting low. (See Chapter 2.)
- EPFs can be registered by the System Administrator for greater efficiency. (See Chapter 5.)

# 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* | In command formats, words in lowercase bold italic indicate variables for which you must substitute a suitable value. In text and in messages, variables are in non-bold lowercase italic. | **LOGIN *user-id***<br><br>Supply a value for *x* between 1 and 10. |
| Brackets [] | Brackets enclose a list of one or more optional items. Choose none, one, or several of these items. | **LD** [**-BRIEF**<br>**-SIZE** ] |
| Underscore in examples | In examples, user input is underscored but system prompts and output are not. | OK, RESUME MY_PROG<br>This is the output<br>of MY_PROG.CPL<br>OK, |
| Ellipsis ... | An ellipsis indicates that you have the option of entering several items of the same kind on the command line. | **SHUTDN *pdev-1***<br>[***...pdev-n***] |
| Hyphen - | Wherever a hyphen appears as the first character of an option, it is a required part of that option. | **SPOOL -LIST** |
| Subscript | A subscript after a number indicates that the number is not in base 10. For example, a subscript 8 is used for octal numbers. | $200_8$ |

# 1

# OVERVIEW

This chapter describes

- The responsibilities of the System Administrator
- Directories containing PRIMOS software
- Server processes and other system services

## ROLE OF THE SYSTEM ADMINISTRATOR

A System Administrator organizes and manages computer systems. Therefore, a System Administrator must

- Plan and set up each system, including the environments and attributes of the system's users
- Allocate system resources
- Set the policy for the use of the systems
- Make the systems secure

The System Administrator is the person to whom users and operators turn when anything goes wrong or when problems arise unexpectedly. Although this book frequently refers to a single System Administrator, at your installation several people may share the job of administering the system. Alternatively, an operator or a user may double as the System Administrator.

The three volumes of the System Administrator's Guide cover many of the tasks that you, in your role as administrator, may be expected to perform. This volume shows you how to allocate disk space and software resources, including creation or revision of a system configuration file and a system startup file.

The *System Administrator's Guide, Volume II: Communication Lines and Controllers* explains the configuring of asynchronous lines, as well as I/O buffer allocation.

The *System Administrator's Guide, Volume III: System Access and Security* explains how to

- Determine the attributes of individual users and the user profiles they must have before they can log in to the system
- Set access rights on MFDs, system directories, and user directories to make the system secure
- Set up the Spooler and the Batch subsystems
- Set schedules for and perform backups
- Monitor system usage

## OVERVIEW OF PRIMOS DIRECTORIES AND FILES

You should be familiar with the directories that comprise PRIMOS. Some directories and their associated files are delivered with all versions of PRIMOS. Others are available as separately chargeable products.

The next sections describe those directories that are nonchargeable and are required to run PRIMOS, those directories that are not required and are nonchargeable, and those directories that contain chargeable software and are not required to run PRIMOS.

Following are files delivered with all versions of PRIMOS.

**BOOT**
This file contains part of the bootstrapping procedure for cold starting the system.

**BOOT_RUN_FILE_TREENAME**
This file contains the pathname of the PRIMOS runfile used in the last successful boot of PRIMOS, in machine readable format. Do not attempt to edit this file.

**Disk Record Availability Table**
This file, sometimes called the DSKRAT, has as its filename the name of the partition on which it resides. It contains a table of the available records on the partition. This table is dynamic; that is, as disk records are used or freed, the table is constantly updated.

The MAKE command creates a new DSKRAT when making a partition. The PRIMOS file system uses the DSKRAT to keep track of available records and the FIX_DISK command uses it in checking and repairing a partition's file structure.

**BADSPT**
A disk surface can have physical defects such as scratches or areas with little or no coating. The BADSPT file contains a list of all records that contain physical defects, or badspots. If a partition has no badspots, this file may be absent. When allocating disk records, PRIMOS scans the BADSPT file to ensure that no information is copied onto unusable records.

**DYNBSP**

Disks on intelligent controllers can be created with a dynamic badspot file, which holds a constantly-updated list of badspots. This file, DYNBSP, controls access to the dynamic badspot file.

## Required Directories

The following top-level directories are required to run PRIMOS.

**CMDNC0**

The directory CMDNC0 contains external PRIMOS commands. External commands are those that are not internally embedded in the operating system; examples of external commands are ED and FIX_DISK. Frequently, this directory contains special commands that have been customized for your particular system. There are internal PRIMOS commands, such as ATTACH, RDY, and LOGOUT, that do not appear in CMDNC0.

CMDNC0 also contains the configuration file (typically named CONFIG) and the system startup file, normally named PRIMOS.COMI.

**DOWN_LINE_LOAD***

DOWN_LINE_LOAD* contains the files that are loaded into the various controller subsystems when the system is booted.

**DSM***

DSM* contains the files and directories needed for Distributed Systems Management. These are described in detail in the *DSM User's Guide*. The directory DSM*>LOGS is required for system and event logging to take place.

**LIB**

The directory LIB contains static-mode libraries.

**LIBRARIES***

The directory LIBRARIES* contains library EPFs.

**PRIRUN**

The directory PRIRUN contains load maps and the PRIMOS runfiles, which are the files used to start up PRIMOS. This directory also contains the PRIMOS.COMI.TEMPLATE file.

**SAD**

The directory SAD (System Administration Directory) contains all user profile and project information. You can boot a system without a SAD, but users could not log in. You use CONFIG_USERS or EDIT_PROFILE to create the SAD, as described in *System Administrator's Guide, Volume III: System Access and Security*. This directory is created when you first invoke CONFIG_USERS or EDIT_PROFILE.

**SEARCH_RULES***

The directory SEARCH_RULES*, created automatically when you run the installation program SYSTEM>ENTRY$.INSTALL.CPL, holds the default system entrypoint search rules file, ENTRY$.SR, and the ADMIN$.ENTRY$.SR file. It also holds several other search rules files, including ATTACH$.SR, BINARY$.SR, COMMAND$.SR, and INCLUDE$.SR.

**SERVERS***

The directory SERVERS* contains the runfiles for the system servers such as the Login Server, Name Server, Copy Server and the Auditor if your system is using C2. This directory must be present.

**SYSTEM**

The directory SYSTEM contains all shared subsystem software, such as FORMS, compilers such as CBL, and utilities such as ED. SYSTEM contains the LOGIN_SERVER.COMI, which is run by the Login Server, and SET_LSR_ACLS.CPL, which sets ACLs on each directory in the System Administration Directory to grant LUR access to the Login Server. This directory also contains the command files INSTALL.STD.COMI, which installs PRIMOS, CREATE.ALL.COMI, which creates the directories required by chargeable software, and INSTALL.ALL.COMI, which installs the chargeable software. Use these as templates, first deleting those products you will not be using, and then running them to install your software.

**UP_LINE_DUMP***

The LAN300 subdirectory of UP_LINE_DUMP* holds LHC dump files when they are generated and the ICS subdirectory holds the InterServer Communication Subsystem dump files.

## Other Important Directories

The following directories are important, but are not required for running PRIMOS:

**BACKUP***

Contains the files that comprise the BRMS utility, which is described in the *Data Backup and Recovery Guide.*

**BATCHQ**

Contains the files that are used whenever Batch jobs are run. These files include the Batch monitor runfile, Batch queue definition files, and job submittal files.

**BOOTRUN**

Contains the BOOT.INSTALL.COMI file, which installs BOOT from this directory into the MFD.

**CONFIG_USERS***

Contains internationalization databases, the screen library, and text files for HELP and USAGE.

**DIAG**

Contains the files that comprise diagnostic tools for use by Customer Service Representatives.

**DOS**

Contains the obsolete single-user operating system, PRIMOS II, in the file DOS.SAVE.

**INFO23.0**

Contains the files that summarize the major changes in the current Revision. The name of this directory always matches the revision number.

**HELP***
Contains HELP files for PRIMOS commands.

**RJSPLQ***
Contains the files to run the Remote Job Entry (RJE) product.

**SEG**
Contains the files that build a SEG file to run as a command.

**SEGRUN***
Contains segment directories (V-mode and I-mode runfiles).

**SIT***
Contains the system internationalization tools for DSM.

**SPOOL***
Contains ASCII files that control the environments of printer operations. This includes files that monitor the Spooler and determine user privileges.

**SPOOLQ**
Contains the files L.DEST and L.TYPE on those systems running OAS. Otherwise, this directory is no longer used.

**SPOOL_DATA***
Contains copies of the files to be printed (unless they were spooled with the -NOCOPY or the -SPOOL_WHILE_OPEN option). There can be several SPOOL_DATA* directories on a system, but only one per partition. You must create this directory.

**SPOOL_QUEUE***
Contains the list of print requests that are awaiting attention from a printer and may contain a few optional files. There should be only one SPOOL_QUEUE* directory per system.

**SYSCOM**
Contains parameter insert files for compilers.

**SYSOVL**
Contains files required by CBL and the data files used by the FORTRAN 77, Pascal, PL/I, PL/I Subset G, and RPG compiler default driver programs. It also contains the EPF error table.

**T&MRUN**
Contains test and maintenance programs used by Customer Service Representatives.

**TOOLS**
Contains files and programs that can perform tasks such as updating or converting certain commands or directories. This directory also contains the driver programs for the PL/I Subset G, Pascal, and FORTRAN 77 compilers.

## Some Optional Directories

Prime offers many additional products, most of which are stored in directories with names ending with the asterisk (*) symbol. Sample directories are:

**FORMS***
Contains files needed to run the Forms Management System (FORMS). Must be installed to use FORMS. See the *FORMS Programmer's Guide.*

**FTSQ***
Contains File Transfer Service (FTS) runfiles, the configuration data base, queues of transfer requests, and copies of user files for transfer. See the *PRIMENET Planning and Configuration Guide.*

**NTS***
Contains the files necessary for Network Terminal Service. This is a chargeable product.

**PRIMENET***
Contains files needed to run PRIMENET networks, and the pre-Rev. 21.0 network event log files. See the *PRIMENET Planning and Configuration Guide.* This is a chargeable product.

## ACL Settings on System Directories

Most system directories require some access control to prevent unintentional or unauthorized use of their contents. See the *System Administrator's Guide, Volume III: System Access and Security* for a discussion of Access Control Lists and their use. Tables 1-1 and 1-2 provide recommended ACL values for system directories.

*TABLE 1-1. Access Rights for System Directories*

| *Directory* | *Minimum Access Needed* |
|---|---|
| BATCHQ | Protection is set by Batch subsystem.<br>Must not be a password directory. |
| CMDNCO | SYSTEM:ALL<br>$REST:LUR<br>*System Administrator*:ALL recommended |
| DEVICE* | SYSTEM:ALL<br>$REST:NONE |
| DSM* | .DSM$:ALL<br>SYSTEM:LUR |
| DOS | SYSTEM:LUR |
| DOWN_LINE_LOAD* | LHC_DLL_SERVER:LUR<br>LTS_DLL_SERVER:LUR<br>SYSTEM:ALL<br>*System Administrator*:ALL |
| HELP* | $REST:LUR |
| INFO23.0 | $REST:LUR recommended<br>Directory name changes to match Rev. |
| LIB | $REST:LUR<br>DALURW recommended for user modifying the libraries |
| LIBRARIES* | $REST:LUR<br>DALURW recommended for user modifying the libraries |
| MFD (on command device) | .DSM$:LUR<br>SYSTEM:ALL<br>$REST:LU |
| PRIRUN | SYSTEM:LUR |
| SAD | Protection maintained by EDIT_PROFILE. |
| SEARCH_RULES* | $REST:LUR |

*TABLE 1-1. Access Rights for System Directories (continued)*

| *Directory* | *Minimum Access Needed* |
|---|---|
| SEGRUN* | $REST:LUR |
| SERVERS* | SYSTEM:ALL<br>LOGIN_SERVER:LUR<br>COPY_SERVER:LUR<br>NAME_SERVER:LUR<br>$REST:LUR |
| SPOOL* | *System Administrator*:ALL<br>.SPOOL_ADMINISTRATOR$:ALL<br>$REST:LUR |
| SPOOL_DATA* | .SPOOL$$:ALL<br>$REST:NONE |
| SPOOL_QUEUE* | .SPOOL$$:ALL<br>.SPOOL_ADMINISTRATOR$:ALL<br>$REST:NONE<br>Must not be a password directory. |
| SPOOLQ | $REST:LUR |
| SYSCOM | $REST:LUR |
| SYSOVL | $REST:LUR |
| SYSTEM | SYSTEM:LUR<br>$REST:LUR (for SYSTEM>DISCS) |
| UP_LINE_DUMP*>LAN300 | LHC_ULD_SERVER:ALL<br>LTS_DLL_SERVER:ALL |

*TABLE 1-2. Access Rights for Special Products*

| *Product* | *Directory* | *Minimum Access Needed* |
|---|---|---|
| DISCOVER | DISCOVER* | Should be a password directory |
| FED | FED* | $REST:RU<br>*Installer*:ALL |
| FORMS | FORMS* | $REST:ALL |
| FTS | FTS | *System Administrator*:ALL |
| | FTSQ* | SYSTEM:ALL<br>YTSMAN:ALL<br>FTP:ALL<br>RT_FTP:ALL<br>*FTS Servers*:ALL<br>$REST:DALURW |
| Networks | NETWORK_MGT* | $REST:LUR |
| NTS | NTS* | SYSTEM:ALL<br>$REST:LUR |
| PRIMENET | PRIMENET* | NETMAN:UR<br>RT_SERVER:UR<br>*Network Administrator*:ALL<br>SYSTEM:ALURWX<br>$REST:NONE |
| | MFD containing PRIMENET* | NETMAN:U<br>RT_SERVER:U |
| PRIME/SNA | PRIME/SNA* | See the *PRIME/SNA Administrator's Guide.* |
| PRIMIX | PRIMIX* | See *Using PRIMIX on the Prime 50 Series.* |
| RJE | RJSPLQ* | See the *Remote Job Entry Phase II Guide.* |
| ROAM | ROAM* | .ROAM_ADMIN:ALL<br>SYSTEM:ALL<br>$REST:LUR |

# SYSTEM SERVERS AND PHANTOMS

In normal operation, PRIMOS runs several server and phantom processes, each of which provides a service for all users. Certain ones must be running for the system to operate, while others are optional, depending on the software you have installed. The servers shown in Table 1-3 must always be able to run.

For the System Administrator, there is one important difference between phantoms and servers: you must configure, by using the NPUSR directive, the maximum number of phantoms allowed. On the other hand, no configuration of the number of servers is necessary, or even possible. Tables 1-3, 1-4, 1-5, and 1-6 show many servers and phantoms. You can always distinguish servers from phantoms by using the LIST_PROCESS -TYPE SERVER command.

Remember, when configuring phantoms you must allow additional phantoms for your users who may be using the PHANTOM or JOB commands.

*TABLE 1-3. System Servers*

| *Server* | *Required By* | *Handles* |
|---|---|---|
| LOGIN_SERVER | PRIMOS | LOGIN and validation |
| LOGOUT_SERVER | PRIMOS | Cleanup after LOGOUT |
| DSMSR | DSM | Message Collection and request distribution |
| DSM_LOGGER | DSM | DSM logging process |
| SYSTEM_MANAGER | DSM | Event logging |
| TIMER_PROCESS | PRIMOS | Timed events |

If your system is part of a network, you will also need some or all of the servers shown in Table 1-4. NM_SERVER is a phantom in spite of its name. All the others are servers.

*TABLE 1-4. Network Servers and Phantoms*

| *Process* | *Required By* | *Handles* |
|---|---|---|
| NETMAN | PRIMENET | PRIMENET support for Ring, Sync, PSDN, LAN300 |
| ISC_NETWORK_SERVER | PRIMENET | Interserver Communications |
| NM_SERVER | NTS | Network Management |
| NTS_SERVER | NTS | LAN terminal server |
| NAME_SERVER | PRIMOS | Replication of root directory |

There are other phantoms and servers that may be running on your system depending on other software packages you may be using. Table 1-5 shows a few examples, all of which are phantoms, not servers.

**Note**

The names of the two FTS servers in Table 1-5 are the choice of the person who administers FTS. FTP and YTSMAN, shown here, are frequently used.

*TABLE 1-5. Additional Phantoms*

| *Process* | *Required By* | *Handles* |
|---|---|---|
| RT_SERVER | PRIMENET | Route through |
| FTP | FTS | File Transfer |
| YTSMAN | FTS | File Transfer |
| BATCH_SERVICE | BATCH | Batch jobs |
| Spooler phantoms | SPOOL | Printing of files |

Additionally there are various temporary servers and phantoms started by the software to perform various tasks. Some of them are listed in Table 1-6. Only the COPY_SERVER is actually a server. The others, in spite of their names, are phantoms.

*TABLE 1-6. Temporary Servers and Phantoms*

| *Process* | *Required By* | *Handles* |
|---|---|---|
| COPY_SERVER | PRIMOS | Disk Mirroring |
| DSMASR | DSM | Applications running under DSM |
| LHC_DLL_SERVER | COMM_CONTROLLER | LHC downline load files |
| LHC_ULD_SERVER | COMM_CONTROLLER | LHC recovery |
| LTS_DLL_SERVER | COMM_CONTROLLER | Loading LTS300 controllers |

# 2

# CREATING AND ALLOCATING DISK SPACE

Among the more important tasks of the System Administrator is the creation and allocation of disk space for system use and for users. Before your disks can be used for reading, writing, and updating information, the disks must conform to your system's requirements and your users' needs.

Here is what you have to know in order to set up your disk space for optimum efficiency and security:

- You must know the type and storage capacity of your disks.
- You must plan for dividing your total disk space into subdivisions (called partitions) and distributing the partitions on your system. Some partitions will hold files; others will be for paging.
- When you allocate paging space, you must make such decisions as the number of paging partitions to use, and the amount of space to allocate for paging.
- When you allocate user space you may set quotas (limits) on the number of records allocated to each top-level directory.

In addition to making decisions about disks, you must decide how to set up your magnetic tape drives. These disk and tape concerns are covered in this chapter.

Other related topics are covered elsewhere:

- The *Operator's Guide to File System Maintenance* describes how to format disks (with the MAKE utility) and how to repair the file system (with the FIX_DISK utility).
- The *System Administrator's Guide, Volume III: System Access and Security* describes how to monitor your system's disk space.

# DISK TYPES AND STORAGE CAPACITIES

Prime supports two types of disk drives:

- Storage Module Disks (SMDs)
- Fixed-Media Disks (FMDs)

Each type is available in several storage capacities. For more information about all the disks that Prime supports, see the *Operator's Guide to File System Maintenance.*

**Storage Module Disks** (SMDs) are platters assembled into removable disk packs. The disk pack is inserted into and removed from its storage module drive. Prime supports two storage capacities for storage module disks: 80 and 300 megabytes, which have disk packs with 5 and 19 usable surfaces, respectively.

**Fixed-Media Disks** (FMDs), also called Winchester disks, are permanently enclosed dust-free drives. Prime supports several storage capacities for Winchesters, ranging from 60 to 817 megabytes. Some are available only on specific CPUs.

# DIVIDING DISK PACKS

Before you format your disks, you must make three decisions:

- How to divide your total disk space into partitions
- The size of the partitions
- How to distribute the partitions on your disk controllers and disk drives

In making these decisions, you have two goals:

- To allocate space equitably among your users and allow for the system's needs for space (including reserving space for future expansion)
- To distribute the I/O workload evenly among your disk drives and controllers

## Dividing Total Disk Area Into Partitions

To create partitions, you need to know the following information:

- The number of users in your various user groups. For example, how many users are in payroll, in manufacturing, or in inventory control?
- The nature of each group's work. How much storage space will each group require for its type of work?
- The workload of each group. Will the workload in each group be light or heavy six months or a year from now? How much storage space will each group require in the future?

- The software products that will be in use. Consult each product's documentation for suggestions on arranging disk storage.
- The amount of security and data reliability required by each group. For example, is there a reference database that must always remain unmodified, and is best kept on a read-only partition? Is there a constantly changing critical database that should be kept simultaneously on two separate disk drives, by use of the process called *mirroring*?
- The frequency of backups. For example, is there a small database requiring daily backups, and a large one for which weekly backups would suffice? Putting the two on separate partitions might substantially reduce the time required to perform daily backups.
- The number of disk drives and their storage capacities, as well as the number of controllers to handle the disk drives.

After you have collected this information and any other information that is important to your installation, you can decide how to partition your total disk space according to your users' needs.

## Size of Partitions

Following are some guidelines for deciding whether to use large or small partitions.

**Advantages of Using Large Partitions:** Failing to grant enough disk space to a user partition at the time of the partition's creation is a common problem. Plan ahead when creating new user partitions, especially if your system is new. Allocate enough partition space so that you reduce the number of times the partition has to be moved, enlarged, or remade. It might be appropriate to set up all, or nearly all, of the space on each disk drive as a single partition.

Ideally, before creating the partitions, you should know which user groups are likely to have substantial increases in their workloads. You can thus allocate more space to those partitions. You can also use quotas to restrict space on large partitions. Other advantages of large partitions are:

- Holding large databases
- Being more efficient in storage
- Being more efficient in access time, due to reduced seek time
- Making it easier to reallocate space among directories

**Advantages of Using Small Partitions:** Smaller partitions provide the ability to write-protect a database, by making a partition read-only.

Other advantages of small partitions are:

- Less data is lost if the partition is erased. That is, if most or all of the data on a small partition is somehow ruined or deleted, less data is lost than on a large partition.
- You gain flexibility in deciding how many directories to have online at any given time.
- You can isolate crucial data requiring mirroring or very frequent backup.
- Administration of short-term directories may be easier. For example the entire partition of student directories might be removed after every school term.

Some systems use small partitions to control the allocation of disk space among users. However, a more effective way of controlling the use of disk space is by setting quotas on top-level directories (as explained later in this chapter).

**Note**

At Rev. 23.0, the System Administrator has the capability of using logical mounts, that is, mounting disk partitions anywhere in the file system hierarchy. This capability may be helpful if you are trying to limit partition size. See Chapter 4 of this guide for more information on logical mounts.

**Backup Considerations:** Disk-to-disk backups with COPY_DISK require source and target partitions to be of equal size. Therefore, you might want to standardize the sizes of your partitions as much as possible. If you use PSR to do disk-to-disk backups, however, the source and target partitions do not have to be of equal size. (See the *Operator's Guide to System Commands* for more information on PSR.)

For more information on backups, see the *Operator's Guide to Data Backup and Recovery* and the *System Administrator's Guide, Volume III: System Access and Security.*

## Distributing Partitions to Drives and Controllers

When adding new partitions to your system or adjusting existing partitions, follow this rule of thumb: Distribute the use of your partitions evenly among your disk drives and distribute your drives evenly among your controllers. An even distribution makes read/write operations faster and the system more efficient. In particular, if two (or more) partitions are often accessed simultaneously, placing them on separate controllers may greatly improve performance. This rule applies equally to file storage and to paging. See the section Allocating Paging Space for further guidelines.

For example, if you have five partitions, two drives, and two controllers, you might place the three smaller partitions on one drive and the two larger partitions on the other, thereby balancing the data distribution as much as possible. Then, you would place one drive on each controller, so that read/write operations on both drives could occur simultaneously.

If you subsequently discover that read and write activity to the smaller partitions is far more frequent than to the larger ones, you might want to divide the smaller partitions

among the two drives, thereby balancing the I/O activity as much as possible. Each drive would then hold one large partition and one or two smaller ones, or one lightly used partition and one or two heavily used partitions.

**Monitoring the Distribution:** Keeping your partitions and drives evenly distributed is an ongoing process and requires that you do the following:

- Monitor the data distribution regularly.
- Watch the trends and patterns in the way your users manipulate their storage space. For example, if one partition's workload increases, more data is added to the partition and the read/write operations on that partition increase substantially.
- Be prepared to adjust the data distribution so that the increase in read/write operations does not hamper system efficiency.

Five PRIMOS commands that monitor system operations and data storage information are AVAIL, LIST_QUOTA, PRIMON, STATUS, and USAGE. These commands, along with other commands and information on monitoring the system, are discussed in the *Operator's Guide to System Monitoring*, and in the *System Administrator's Guide, Volume III: System Access and Security*.

Two of the System Information and Metering (SIM) commands, described in the *DSM User's Guide*, are particularly helpful for monitoring disk and file usage. LIST_DISKS displays the status of the local disk partitions (or all disk partitions if your system is not running the Name Server), and LIST_UNITS displays the names of each user's active files.

## Pre-Rev. 22.0 Partitions

Although you can use Rev. 18, Rev. 19, and Rev. 20 partitions on a Rev. 22.0 or Rev. 23.0 system, to gain the advantages of the file structure introduced at Rev. 21.0, you should convert pre-Rev. 21.0 partitions to Rev. 22.0 or Rev. 22.1 format. Hashed directories, and the Date/Time Created and Date/Time Accessed attributes for files and directories were introduced at Rev. 20.0; quotas and ACLs were introduced at Rev. 19.2. To convert pre-Rev. 21.0 partitions to Rev. 22.0 or Rev. 22.1 format partitions, use the MAKE utility. (Although it is not likely that you will need to change Rev. 21.0 format partitions to Rev. 18 format or Rev. 19 format, you can perform such a conversion with the -DISK_REVISION option of MAKE.) Before attempting any conversions, read the *Operator's Guide to File System Maintenance* carefully.

A Rev. 22.1, 22.0, 21.0, or 20.0 partition cannot be added locally on a pre-Rev. 20 system, but it can be added remotely. Users logged in on a pre-Rev. 20.0 system and attached to a Rev. 20.0 or later partition cannot display the Rev. 20.0 Date/Time Created and Date/Time Accessed file and directory attributes with the LD command.

# ALLOCATING PAGING SPACE

As System Administrator, you must make sure that your system has enough paging space, not only when you first configure it, but later, when you have added more users. If there is not enough paging space, you may find that users cannot log in when the system is being heavily used.

To do this, you must make two decisions: how much paging space to allocate, and the number of paging partitions to use. To create the partitions for paging, use the MAKE utility, described in the *Operator's Guide to File System Maintenance.* For additional information concerning conversion of pre-Rev. 20 paging partitions, see the handbook for your CPU model.

## An Overview of Paging

Programs execute in a computer's main, high speed, physical memory. Because the amount of this memory is limited, the PRIMOS operating system uses **paging space** (disk space on **paging partitions**) as secondary, or virtual, memory. The main memory, the secondary memory, and the programs are divided into **pages** of 2048 bytes. For a program to execute, only those pages of it containing the current instruction and the data used by that instruction need be in main memory. When an instruction refers to data that is not in main memory (or when the next instruction is not in main memory), the appropriate pages are brought in as needed. This method is called **demand paging.** To make space for the pages being brought in, other pages, less recently used, may need to be written out to the paging space.

## EPFs and Paging

EPFs have filenames ending in .RUN; you can find EPF programs in CMDNC0, and libraries for EPF programs in LIBRARIES*. When paging occurs for an EPF program, pages of per-user data are read from (or written to) the paging partitions as required. The pages of sharable data (such as executable code) are never written out at all, because they never change. Instead, they are read directly from the .RUN files as required.

Paging of an EPF program often requires simultaneous disk activity both on the COMDEV (the partition where CMDNC0 and LIBRARIES* reside) and on the paging partitions. If there is a paging partition on the same physical disk that holds the COMDEV, the heads of that disk may spend much time seeking between the COMDEV and paging partitions. It is usually a good idea to have paging partitions and the COMDEV on separate disks, or even separate disk controllers.

The executing user program cannot detect that it is being paged. It sees no distinction between main memory and secondary memory; indeed, it has available to it an addressing space much larger than the main memory. This feature, along with the memory management scheme for implementing it, is sometimes called **virtual memory.**

For a more accurate, and much more detailed, explanation of memory management in Prime computers, see the *System Architecture Reference Guide.*

## Running Out of Virtual Memory

A system can run out of virtual memory for one of three reasons. Each requires you to take a different action.

- If a user runs out of segments (indicated by a `Not enough segments` message), you may need to use EDIT_PROFILE or CONFIG_USERS to increase the number of static or dynamic segments for that user.
- If a system runs out of system segments (indicated by the user error condition NO_AVAILABLE_SEGS$), you may need to increase the NSEG or NVMFS configuration directives.
- If the paging partition becomes full (indicated by the user condition PAGING_DEVICE_FULL$), you must add more paging partitions or increase the size of the existing partitions.

**Note**

At Rev. 23.0, it is very unlikely that your system will run out of paging space. A function has been added in PRIMOS that monitors the amount of paging space available and generates warnings at the supervisor terminal so that there is time to correct the problem before PRIMOS halts. This enhancement to PRIMOS is explained later in this chapter in the section called Paging Alarms and the SET_PGALARM Command. The SET_PGALARM command is documented in the *Operator's Guide to System Commands.*

## Establishing Paging Partitions

A system can use up to eight partitions for paging. Use the PAGING directive, discussed in Chapter 8, Configuration Directives, to tell PRIMOS which partitions are for paging.

**Notes**

The PAGING directive replaces the now-obsolete PAGDEV and ALTDEV directives. See Appendix A, Obsolete and Rarely Used Commands and Directives.

Also, the PRATIO operator command has replaced the obsolete PRATIO directive. See the *Operator's Guide to System Commands* for more information.

The paging partitions should, if possible, be on disk drives that are not used heavily for other purposes. For instance, there should not be a paging partition on the drive that holds the command device partition (the COMDEV). Multiple paging partitions may help divide the loads among the disk drives and controllers. There is, of course, no advantage in putting more than one paging partition on any disk drive. For help in making these decisions, consult your Prime Customer Service Representative.

## Paging Space Requirements

Paging space is allocated in units of 16 kilobytes. This means that when the first eight pages of a segment are accessed, only 16 kilobytes of paging space are used by the segment. Therefore, a given amount of paging space can accommodate a varying number of segments, depending on the number of pages used in each.

Because PRIMOS cannot determine whether the amount of paging space is adequate for the number of available system segments (set by the NSEG configuration directive), paging space may be exhausted while the system is running. If paging space is exhausted, the user requesting the additional memory receives the error condition PAGING_DEVICE_FULL$.

## Determining the Amount of Paging Space

There are two methods for determining the amount of paging space:

- Calculate the maximum and minimum amounts of paging space your system could require, using the formulas given in the sections that follow. Your optimal paging space will fall somewhere between the two. For small systems and lightly loaded systems the paging space should probably be closer to the minimum figure, and for large or heavily loaded systems, closer to the maximum.
- A good rule of thumb for determining the amount of space you need for paging is to allocate 7800 records (or 16 megabytes, that is, one disk surface of a 300 megabyte disk) for paging for every 6 to 8 users. (In other words, allocate approximately 1000 paging records per user.) The number of users is the sum of the NTUSR, NRUSR, NTSUSR, NPUSR, and NSLUSR configuration directives, plus the number of server processes. See Chapter 1 of this guide for information on server processes.

**Calculating Maximum Paging Space:** The formula for calculating the maximum amount of paging space needed on your system is:

MAX_SPACE = NSEG * 64

where

MAX_SPACE is the maximum paging space needed (in records).

NSEG is the total virtual address space for the system, as set by the NSEG configuration directive.

64 is the number of pages per segment.

**Calculating Minimum Paging Space:** The formula for calculating the minimum amount of paging space needed is:

MIN_SPACE = PRIMOS + SHARED_PRODUCTS + (NUSR * 304)

where

MIN_SPACE is the minimum amount of paging space (in records) your system requires.

PRIMOS is the number of pages used by PRIMOS. This is 2048 at Rev. 23.0.

SHARED_PRODUCTS is the total number of pages used by the shared products on your system. Table 2-1 lists the number of pages per product and pages per user needed for each shared product. Add the per-product figures for each shared product in use on your system. For each product, multiply the per-user pages by the expected number of simultaneous users for that product (not by the total number of users on the system). Use the total of these figures as the number for SHARED_PRODUCTS.

NUSR is the sum of the configuration directives NTUSR + NPUSR + NRUSR + NTSUSR + NSLUSR, plus the number of servers.

304 is 38 segments per user * 8 pages per segment.

*TABLE 2-1. Space Required by Shared Products*

| *Product* | *Per-Product Pages* | *Per-User Pages* |
|---|---|---|
| BASIC | 0 | 48 |
| BASICV | 56 | 24 |
| CBL | 376 | 184 |
| DBG | 224 | 64 |
| DBMS | 194 | 178 |
| DPTX | 53 | (see Notes) |
| ED | 24 | 8 |
| EDB | 0 | 8 |
| EMACS | 448 | 52 |
| FED/FORMS | 160 | 280 |
| FTN | 0 | 40 |
| FTS | 152 | 272 |
| LOAD | 0 | 16 |
| MIDASPLUS | 320 | 200 |
| PMA | 0 | 16 |
| PSD | 0 | 8 |
| RJE | 0 | (see Notes) |
| ROAM | 288 | 48 |
| RPG | 0 | 32 |
| RUNOFF | 0 | 48 |
| SEG | 0 | 40 |
| SORT | 0 | 24 |
| VPSD | 0 | 16 |

*Notes for Table 2-1.*

DPTX The per-system value assumes a maximum configuration of 7 emulators running and 1 line for support use. The per-user value depends on the type of terminal in use. Values are as follows: for the PT45™, 64; for the PT46, 56; for the OWL, 53; for the PST 100™ and PT200™, 60.

RJE To calculate per-user paging space for RJE, allow 208 pages for the common runfiles, plus 168 pages for each emulator you use (1004, 200UT, 7020, GRTS, HASP, X80, XBM).

## Split Paging Disks

A paging partition is called a **split disk.** It is split between paging space and file system storage, so that there is room for badspot files. See the *Operator's Guide to File System Maintenance* for several important instructions on constructing paging partitions, including the use of the -SPLIT option of the MAKE command.

## Paging Alarms and the SET_PGALARM Command

Prior to Rev. 23.0, PRIMOS did not perform any checks against the amount of available paging space and the number of users. The System Administrator was given no warning when paging space was getting low. As a result, paging space would continue to be depleted until there was no longer enough to handle the system's needs. When that point was reached, PRIMOS would suddenly halt without warning, making it necessary to run FIX_DISK.

At Rev. 23.0, PRIMOS has an automatic monitoring function that generates warning messages, inhibits logins, forcibly logs out users, and finally shuts down the system at various stages of paging space depletion. All warning messages generated by PRIMOS are logged to DSM. The SET_PGALARM command is available as well (specifically the -DISABLE option), so that the Administrator can inhibit warning messages from being displayed if he or she is aware of the situation and intends to continue operating. Once beyond the warning message stage, however, this command cannot be used to prevent PRIMOS from inhibiting logins, forcibly logging out users, or shutting down the system.

There are five paging thresholds at which warning messages and/or other events (for example, inhibited logins) are generated by PRIMOS. The five thresholds, the messages generated at the supervisor terminal, the messages received by users, the events that occur after the warning message stages, and the actions you should take are explained in the following sections. As System Administrator, if you take the appropriate action at each threshold you greatly reduce the likelihood that your system will run out of paging space.

**Note**

The method for calculating the amount of paging space you should configure, and the recommended amount of paging space per user have not changed at Rev. 23.0. (See the section Allocating Paging Space earlier in this chapter.)

**Initial Paging Allocation Check:** The first check on paging space that PRIMOS performs (before you reach any of the five paging thresholds) occurs when you boot your system. As explained earlier in this chapter, you should allocate approximately 1000 paging records per configured user. If, when you boot PRIMOS, you have allocated less than the recommended amount, you see the following message at the supervisor terminal:

```
System Paging Allocation Below Recommended Amount
(1000 Pages Per Configured User)
System Configured With xxx Paging Records
Recommend yyy Records For nnn Users
```

This is an informational message and does not require you to take any action. You should, however, configure your system using the recommended amount of paging space or you could have problems later.

**The First Paging Threshold:** When the first paging threshold is crossed, you receive the following message at the supervisor terminal:

```
WARNING  n% Paging Device Utilization day, date/time
Paging Records Configured xxx
Paging Records Used       yyy
Paging Records Available  zzz
```

Users do not receive a warning message when this threshold is crossed.

*xxx* is the number of paging records you allocated at system configuration.

*yyy* is the number of records used by the system for paging at the time of the violation.

*zzz* is the number of records remaining for system use minus the number of records reserved for the Locate Buffers.

**The Second Threshold:** When the second paging threshold is crossed, you receive the same message at the supervisor terminal as at the first threshold. The percentage of paging device utilization, of course, will be higher. In addition, the user causing the page fault receives the following message:

```
WARNING, xx.x% Paging Device Utilization, Notify System Administrator
```

**Note**

It is possible to inhibit warning messages from being generated at the first two thresholds by using the -DISABLE option to the SET_PGALARM command, once the System Administrator has become aware of the fact that paging space is being depleted and has decided to continue operating under those conditions.

Only warning messages generated at the first two thresholds can be inhibited, however. Once you reach the third threshold (described next), you can no longer inhibit messages or events.

**The Third Threshold – Logins Inhibited:** When the third paging threshold is crossed, any further user logins are inhibited. You receive the following message at the supervisor terminal:

```
Login Failed, Insufficient Paging Space Available
Paging Records Configured xxx
Paging Records Used       yyy
Paging Records Available  zzz
```

In addition, the user who is prevented from logging in receives the following message:

```
***Login Failed***
Insufficient Paging Space Available, Notify System Administrator
```

**The Fourth Threshold – Forced User Logout:** The fourth threshold is the forced logout threshold. At this point, logins are still inhibited. Users that are not privileged users are forcibly logged out. One error message is sent to the supervisor terminal regarding ordinary users and a different message is sent regarding privileged users. The following error message is sent regarding an ordinary user:

```
(User id) Logged Out Due To Insufficient Paging Space
day, date/time
Paging Records Configured xxx
Paging Records Used       yyy
Paging Records Available  zzz
```

The following error message is sent regarding a privileged user:

```
Insufficient Paging Space Detected
Paging Records Configured xxx
Paging Records Used       yyy
Paging Records Available  zzz
```

Privileged users are not logged out.

The logged out user receives the following message:

```
(User nnn) Logged Out Due To Insufficient Paging Space
day, date/time
```

**The Fifth Threshold – Orderly System Shutdown:** When the fifth paging threshold is crossed, PRIMOS automatically proceeds with an orderly system shutdown before all paging space needed for such a shutdown is depleted. This reduces the need to run FIX_DISK.

You receive the following message at the supervisor terminal:

```
SHUTDOWN Due to Insufficient Paging Space
```

Users receive the standard PRIMOS inactive message:

```
WAIT,
PRIMOS NOT IN OPERATION
```

# ALLOCATING USER SPACE WITH QUOTAS

Ensuring equitable sharing of disk storage among users is a primary function of the System Administrator. You can provide that equity by setting limits (called **quotas**) on the amount of storage space that directories occupy on a partition.

The quotas, which are measured and allocated by the number of disk records, can be set by both the System Administrator and the user with the SET_QUOTA command. As the

System Administrator, you are responsible for setting and modifying the quotas on top-level directories.

**Note**

A quota cannot be placed on an MFD.

After you have set quotas on your system's top-level directories, users can set or modify quotas on subdirectories only if they have Protect rights (in ACL directories) or Owner rights (in passworded directories) to the next higher directory. That is, the user must have the appropriate rights to the directory that contains the subdirectory whose quota is to be set.

Users can find instructions and guidelines for setting and modifying quotas in the *PRIMOS User's Guide.*

## Four Strategies for Setting Quotas

The amount of disk space on a partition that is reserved for users is the number of records remaining after you allocate space to paging and to mandatory PRIMOS files and directories. After you have determined this space, you can use one of four strategies discussed below to distribute and manipulate user disk space. The strategies all include setting quotas on top-level user directories.

Set quotas on top-level directories according to how structured you want your user space to be. That is, decide whether to set strict limits on each user (or user group), or whether to set looser limits within which users compete for the disk space.

You can use any of four major strategies for setting quotas on top-level directories:

- The **Exact** strategy, in which the sum of the quotas on the top-level directories equals the capacity of the partition.
- The **Undercommitted** strategy, in which the sum of the top-level quotas is less than the capacity of the partition.
- The **Overcommitted** strategy, in which the sum of the top-level quotas is greater than the capacity of the partition.
- The **Unregulated** strategy, in which one or more directories on the partition has no quota.

**The Exact Strategy:** The Exact strategy distributes all the disk's space precisely among users. Each user is guaranteed his or her entire quota. No user's usage of disk space affects other users. No user encounters "Disk Full" errors, because the quota limit is always reached first.

For example, suppose your partition (MFD) has a capacity of 100,000 records that are reserved for users' work space, and that you plan to install four directories, as shown in Figure 2-1. Taking a strict approach, you could ensure that your users never use up more than 100,000 records by setting quotas that total the capacity of the partition. You might

give one directory 40,000 records and the other three directories 20,000 records each. After setting the quotas, you would monitor which top-level directories were using their space and modify the quotas accordingly.

**The Undercommitted Strategy:** The Undercommitted strategy is as strict as the Exact strategy, providing the same guarantee of available disk space to the users. Additionally, it keeps space in reserve for expansion or emergency use.

The Exact and Undercommitted strategies both create an incentive for the users to be more efficient, reserving their space for essential data and deleting unneeded data.

Using the 100,000-record partition of the previous example, you could set a quota of 15,000 records on each of the four directories, thus ensuring that you would always have 40,000 records in reserve.

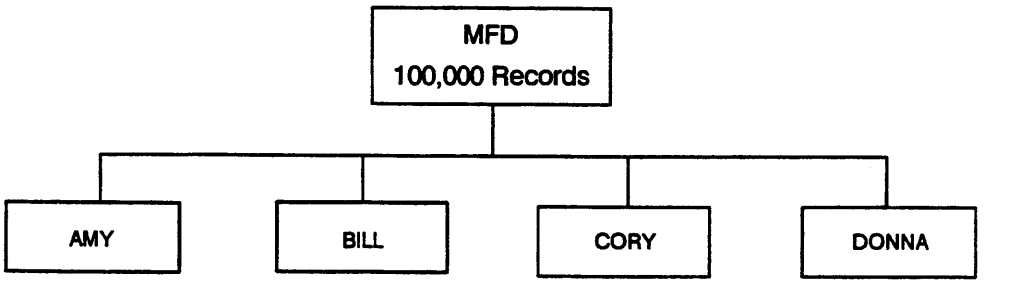


*FIGURE 2-1. Directory Structure for Disk Quota Examples*

**The Overcommitted Strategy:** The Overcommitted strategy is less rigid than the Exact or the Undercommitted. This strategy, in which the sum of the directory quotas is greater than the capacity of the disk, causes users to compete for disk space. The Overcommitted strategy may result in more complete utilization of the partition's space than does the Exact strategy.

Some disadvantages to the Overcommitted strategy are:

- The partition can become completely filled, resulting in "Disk Full" errors, before any users have exceeded their quotas.
- Users may waste space by keeping unnecessary files around, because they perceive their usage to be far below the available quota.
- A quota does not guarantee space to a directory. Instead, a directory is guaranteed space only to the extent that the size of the partition exceeds the sum of the quotas for the *other* directories. Here is how guaranteed space works out if we apply quotas to the partition in Figure 2-1 using the Overcommitted strategy. Remember, although the sum of all the quotas in this example is 145,000 records, the size of the partition is only 100,000.

| *Directory:* | AMY | BILL | CORY | DONNA |
|---|---|---|---|---|
| *Quota:* | 20,000 | 65,000 | 10,000 | 50,000 |
| *Sum of other quotas:* | 125,000 | 80,000 | 135,000 | 95,000 |
| *Guaranteed space:* | none | 20,000 | none | 5,000 |

**The Unregulated Strategy:** The least rigid strategy is the Unregulated strategy, where one directory (or more) has no quota at all, or has a quota larger than the size of the partition. A directory has no quota in either of two circumstances:

- The directory has never been assigned a quota.
- The quota size has been set to zero.

You can improve system performance when using the Unregulated strategy by setting an extremely large quota instead of no quota on each unregulated directory. The large quota should exceed the partition size. See the section below, Speeding Up the LD and LIST_QUOTA Commands.

The storage capacity of a nonquota directory is limited only by the physical capacity of the partition. Setting no quotas on directories gives users the impression that their allotment of disk space is unlimited. If any directory on a partition has no quota, then no other directory on that partition has guaranteed space.

You might use the Unregulated strategy if you have a special user group whose members, by the nature of their work, must be trusted with a seemingly unlimited amount of disk space. With a 100,000-record partition, you might set two of your directories to 20,000 records each, and give the two others no quota. None of the four directories would have any guaranteed amount of disk space.

## Speeding Up the LD and LIST_QUOTA Commands

You can improve the performance of the LIST_QUOTA and LD commands significantly by placing a quota on top-level directories. A quota causes PRIMOS to maintain up-to-the-minute quota information at no performance cost. Quota information is therefore readily available and does not have to be collected each time a user issues the LIST_QUOTA and LD commands. Performance is particularly improved for very large directory structures.

To improve performance without restricting space, use a very high quota (such as 1,000,000), which in essence removes any quota restriction on the directory.

## Monitoring Quotas

After you set quotas on top-level directories, you should monitor the directories to determine how many records are being stored in them. If necessary, you may have to modify the quotas. To monitor the use of space in directories, use the LIST_QUOTA, LD, and SIZE commands.

**Note**

You may need to modify quotas when disks are logically mounted on your system because quota restrictions do not cross logical mount boundaries. Records originally used by a mount-point directory become available when contents of the directory are copied to a newly added disk and then deleted from the mount-point directory. These records can be reallocated among remaining directories in the tree. (See Chapter 4 of this guide for more information on logical mounts and quota restrictions.)

The LIST_QUOTA command lists the maximum quota on a directory, the total number of records used by the entire subtree (beginning with and including the designated directory), and the number of records used by this particular directory. For details on LIST_QUOTA, see the *PRIMOS User's Guide* and the *System Administrator's Guide, Volume III: System Access and Security.*

The LD command also supplies information on quotas and record usage. The SIZE command lists the size of directories and files. For more information on these commands, see the *PRIMOS User's Guide* and the *PRIMOS Commands Reference Guide.*

To modify a quota, use the SET_QUOTA command.

**Calculating Storage Availability:** To determine how much storage space is left in a directory, you must consider all quotas set on the entire directory tree and also the total current storage used by the entire directory tree.

See the *PRIMOS User's Guide* for explanations and illustrations of how to calculate storage availability.

**Recovering From Quota Overloads:** If a user tries to store data that will cause a quota to be exceeded, PRIMOS returns the message `Maximum quota exceeded` and does not allow storing the material.

For information on how to recover from quota overloads (including those that occur during an editing session), see the *PRIMOS User's Guide.*

# MAGNETIC TAPE DRIVES

The SETMOD command controls assignment of tape drives. Use this command to allow users to assign tape drives from their terminals, to require them to ask the operator, or to prevent use of tape drives.

The SETMOD command has three formats:

- SETMOD -USER (which is the default state for the system) allows users to perform their own tape operations. Users can issue the ASSIGN command either to assign tape drives to themselves, or to request the operator to perform the tape operation (which includes assigning the tape drive and setting its characteristics). The latter choice allows phantom jobs and batch jobs to run under operator control, while interactive jobs can run under operator or user control. Either the user or the operator can use the UNASSIGN command to unassign a tape drive that a user has assigned.

- SETMOD -OPERATOR causes the ASSIGN command to channel all requests for tape drives to the supervisor terminal. The operator must approve or disapprove each request. Either user or operator can UNASSIGN a tape drive after it is assigned to a user. This mode is appropriate if you do not permit users to enter the computer room.

  To set up your system to function in this mode as a matter of course, add the SETMOD -OPERATOR command to your PRIMOS.COMI file, so that the command is invoked when the system is cold started.

- SETMOD -NOASSIGN prohibits all tape drive assignments. When the system is in -NOASSIGN mode, an attempt to ASSIGN a tape drive produces a message stating that tape drives cannot be assigned at the present time. To make tape drives available again, use the SETMOD command with either the -USER or -OPERATOR option.

  Use the -NOASSIGN mode when the operator is not available to handle tape requests or when you want no tape operations conducted.

If you use -USER mode, you must still decide whether to allow users to enter the computer room to load and unload tapes (and perhaps to keep them in your tape storage facility), or whether to allow only operators to perform these tasks.

See the *Operator's Guide to System Commands* for the complete instructions on using the SETMOD command, and the *Operator's System Overview* for examples.

# 3

# THE PRIMOS SEARCH RULES FACILITY

## OVERVIEW OF SEARCH RULES

The PRIMOS search rules facility is a general-purpose mechanism for specifying a search sequence. It enables you to prespecify locations for PRIMOS to use when conducting a search. Each prespecified location is known as a **search rule.** A search rule names a location that may contain the object of the search. For example, a directory name would be a search rule when the object of the search is a file.

Search rules are grouped into sequences known as **search lists.** A search list is an area in memory that contains search rules, listed in sequential order. You initially write the sequence of search rules into a text file known as a **search rules file.** Before these search rules can be used, they must be copied from the search rules file into a search list. The process of copying search rules into a search list is known as setting the search list.

When using a search list, PRIMOS searches the first search rule in the search list, then the second search rule in the list, and so forth until PRIMOS either finds the object of the search or encounters the end of the search list.

One common use of search rules is to locate file system objects without requiring the user to enter the fully qualified pathname. You can create different search lists for different kinds of search operations. For example, you can establish a search list to search multiple disk partitions for a top-level directory or establish a search list to search multiple directories for a file.

You can invoke such a search by using a PRIMOS command, a CPL function, or a subroutine call. The EXPAND_SEARCH_RULES (ESR) command, for example, takes a filename as input and uses the search rules facility to determine the absolute pathname of that file. The search rules facility is invoked automatically by system software, such as the PRIMOS command processor and the BIND program linker.

PRIMOS maintains a separate group of search lists for each process. This means that users can customize their search lists to meet individual requirements. Because a group of search

lists is specific to a process, a program uses the search lists of the user (or phantom) currently executing the program.

PRIMOS uses a basic set of active search rules lists. As System Administrator, you can accept or modify the systemwide search rule sequences that apply to all users. You can control two different categories of systemwide search rules:

- System default search rules. These are the default for a user who does not create a list of personal search rules. If users create a personal search rules list, they have the option of including or excluding System Default search rules.
- Administrator search rules. These are always first, before both system default rules and user-defined search rules. Users cannot modify or exclude Administrator search rules.

The search rules for both categories reside in the directory SEARCH_RULES*. Each process at initialization reads the presently active set of search lists, located in the directory SEARCH_RULES*. The process copies the rules and follows them until process completion, even if the lists in SEARCH_RULES* receive changes in the interim.

The following section describes the contents of SEARCH_RULES*. It also summarizes how to modify and maintain both the System Default search rules and the Administrator search rules. For a more detailed description of search rules, see the *Advanced Programmer's Guide II: File System.*

## The SEARCH_RULES* Directory

Both System Default and Administrator search rules must be located in the directory SEARCH_RULES* on the command partition. Set the ACLs for this directory to provide all users with LUR access, but reserve all other access for the System Administrator. SEARCH_RULES* holds a separate file for each type of search list. It should contain the following System Default search rules files:

- ATTACH$.SR: searches directories to locate specified directory
- BINARY$.SR: searches directories to locate binary object code files
- COMMAND$.SR: searches directories to locate executable code files
- ENTRY$.SR: searches EPF library files to locate entrypoints
- INCLUDE$.SR: searches directories to locate source code files

SEARCH_RULES* should also include the following Administrator search rules list:

ADMIN$.ENTRY$.SR

You can add other search lists within the directory SEARCH_RULES*. Create them as you would any other file, following the same naming conventions, but note that the $ before the .SR is reserved for PRIMOS search rules. Use the form *list_name*.SR for additions to System Default search rules. Use the form ADMIN$*list_name*.SR for Administrator search rules.

Users should have LUR access to all search lists. You can also add search rules to existing System Default and Administrator search rules lists.

## Search Rule Keywords

A search rules file can contain keywords that perform specific operations. Keywords that begin with a hyphen are directions to the search rules facility. These directions are carried out either when you set the search list or when you perform a search operation on that search list. Keywords enclosed in square brackets are variables for which the appropriate literal is supplied when the search list is used. The following are the available search rule keywords:

```
-insert
-system
-optional
-added_disks
-static_mode_libraries
-primos_direct_entries
[origin_dir]
[home_dir]
[referencing_dir]
-public
```

You should place each keyword on its own line in a search rules file. Keywords and search rules can be intermixed in any sequence within a search rules file. Keywords can be written in either uppercase or lowercase. See the *Advanced Programmer's Guide II: File System* for definitions of and more information about these keywords.

## System Default Search Rules

The System Default search rules automatically become the active search rules for each user at login. User search rules revert to system defaults following a logout or ICE operation, unless an invoked login file uses SET_SEARCH_RULES (SSR) to set rules otherwise. Unless a user specifies otherwise, the System Default search rules precede personalized user search rules.

Users can issue the SET_SEARCH_RULES (SSR) command with its -NO_SYSTEM option to totally exclude System Default search rules in favor of user search rules. Users who want only to defer access to these defaults can do so by including the -SYSTEM search rule at the desired point in the sequence of their user search rules.

The System Default search rules replace PRIMOS-level search support. These search rules provide search support identical to that provided by PRIMOS.

However, if a user either deletes the System Default search rules or excludes them from a user search list, that user can lose PRIMOS functionality. A user who accidentally does this may repair the problem by adding -SYSTEM as the final rule in the user search list.

Finally, if there are no search lists, PRIMOS in many cases performs the same location operations as the System Default search rules. (See the section ACLs and the ATTACH Command in Chapter 5 of the *System Administrator's Guide, Volume III: System Access and Security*.) For example, without an ATTACH$ search list, PRIMOS searches the disk table or the Global Mount Table when an MFD or pathname is requested. Without a COMMAND$ search list, PRIMOS searches CMDNC0 only.

## System Administrator Search Rules

As System Administrator, you can define sets of search lists that always precede system default or user-defined search rules for all users. Administrator search rules apply systemwide to all processes, whether governed by System Default or by user-specified search rules. Users cannot modify, exclude, or resequence Administrator rules in their search lists. This requirement ensures centralized control of search operations.

Administrator search lists are defined in SEARCH_RULES*, using the naming convention ADMIN$.*list_name*.SR. A single, default Administrator search list is provided (ADMIN$.ENTRY$.SR), but the System Administrator may add to that list or create additional Administrator search lists. The Administrator search lists must have the same ACL protection as their parent directory:

*system_administrator*:ALL
$REST:LUR.

The setup of ADMIN$.ENTRY$.SR establishes identical initial search rules for all system users. Because these rules affect all user processes, the System Administrator should limit search rules to essentials. Each user incurs a slight processing delay to receive the advantages of the Administrator search rules. If an Administrator search rule helps only a few users, it should be changed into a private search rule to minimize processing overhead.

The System Administrator can preface an Administrator search rule with the search rule keyword -OPTIONAL. If you specify an Administrator search rule as optional, each user can either enable that search rule using the SR$ENABL subroutine or leave it disabled. Users can thus enable and disable optional Administrator rules for their own processes.

To maintain security, however, the System Administrator must apply one Administrator search rule to all users. ADMIN$.ENTRY$.SR must contain the keyword -PRIMOS_DIRECT_ENTRIES. Failure to include this Administrator search rule results in a breach of system security. The ADMIN$.ATTACH$.SR search list (if one exists) should *not* contain the entry -ADDED_DISKS, because if that entry is included it invalidates all individual user-defined search rules.

Users can modify search rules and search lists by the use of certain PRIMOS subroutines, but there are special restrictions on the application of search rule subroutines to Administrator search rules.

- SR$ADDB and SR$ADDE cannot add a search rule before an Administrator rule.
- SR$REM cannot remove an Administrator search rule from a search list.
- SR$SETL cannot be used to modify the locator pointer for an Administrator rule.

Note that another subroutine, SR$DEL, does indeed delete the complete search rules list, including the Administrator and the System Default search rules. However, recreating the search list by any means (the SSR command, the SR$CREAT, SR$SSR, SR$INIT subroutines, or initializing the system) reestablishes the search list and its Administrator rules. For a full description of these subroutines, refer to the *Subroutines Reference II: File System.*

## Other Administrative Aspects

The Search Rules facility uses three PRIMOS commands. The first of these, EXPAND_SEARCH_RULES (ESR), is in CMDNC0. However, the other two are internal commands. These commands, SET_SEARCH_RULES (SSR) and LIST_SEARCH_RULES (LSR), may therefore be invoked even when CMDNC0 is inaccessible. Search rules are therefore independent of CMDNC0.

Search rules initialize to defaults (Administrator rules and System Default rules) at each system initialization. A program that executes successfully with the user's search rules may either fail or give different results when rerun after an Initialization of Command Environment (ICE), unless the user's LOGIN command file specifies the user's search rules.

Search rules also initialize to defaults at each process initialization. Two successive initializations of the same process may give different results, if the Administrator rules or System Default rules received changes between the two initializations.

System Administrators and Operators should establish procedures for ensuring that a job is not run unless the required search rules have been set.

An error in a search rules list can prevent the initialization of search rules. If there is an error in search rules initialization at cold start, the following message is displayed at the supervisor terminal:

```
Error initializing search rules.  Please check template files
in SEARCH_RULES*.
```

The System Administrator or Operator can then isolate the problem as follows. At the supervisor terminal, type LSR to identify the lists that have been set correctly. Examine the remaining files in SEARCH_RULES* to uncover the error. If the error is not obvious, use SSR to try to set rules again. If SSR cannot set search rules after a cold start, it outputs other error messages to assist you.

The following sections explain the three types of search rule lists that are of particular concern to the System Administrator: ENTRY$, ATTACH$, and COMMAND$. BINARY$ and INCLUDE$ are explained further in the *Advanced Programmer's Guide II: File System.*

# ENTRY$ SEARCH RULES

The system entrypoint search list (ENTRY$) determines the order in which PRIMOS searches libraries to find a match to a subroutine entrypoint in a program. The template file for the default entrypoint search list, ENTRY$.SR, is kept in the directory named SEARCH_RULES*. The installation program SYSTEM>ENTRY$.INSTALL.CPL automatically creates the SEARCH_RULES* directory, if it does not already exist, and copies SYSTEM>ENTRY$.SR into it. You can delete the copy of ENTRY$.SR file in the SYSTEM directory after you check that ENTRY$.SR is in the SEARCH_RULES* directory.

In addition to containing ENTRY$.SR, the SEARCH_RULES* directory must also contain a file called ADMIN$.ENTRY$.SR. This file, which the installation program also puts in place, contains the following single rule:

```
-PRIMOS_DIRECT_ENTRIES
```

The Administrator can add to this list if necessary but must not delete the -PRIMOS_DIRECT_ENTRIES search rule.

## Format of the Search List

The system entrypoint search list template file is a text file that contains a list of search rules (one search rule per line). A search rule has one of the following formats:

- The pathname of a systemwide library EPF. For example:

  LIBRARIES*>FTN_LIBRARY.RUN

- A keyword that begins with a hyphen. For example:

  -STATIC_MODE_LIBRARIES

  or

  -PUBLIC *epf_name*

The -PUBLIC search rule refers to registered EPFs. See the section Registered EPFs in Chapter 5 of this guide for more information on registered EPFs.

Following is an example of an ENTRY$.SR file.

```
OK, lsr entry$

List: ENTRY$
Pathname of template: <SYSS62>SEARCH_RULES*>ENTRY$.SR

    -PRIMOS_DIRECT_ENTRIES
    -static_mode_libraries
    -public
    LIBRARIES*>CC_LIBRARY.RUN
    LIBRARIES*>SYSTEM_LIB$PRC.RUN
    LIBRARIES*>SYSTEM_LIB$PRG.RUN
    LIBRARIES*>MAIL_LIBRARY.RUN
    LIBRARIES*>SIT_LIBRARY.RUN
    LIBRARIES*>APPLICATION_LIBRARY.RUN
    LIBRARIES*>EMACS_STATE.RUN
    LIBRARIES*>X409LIB.RUN
    LIBRARIES*>DSMLIB.RUN
    LIBRARIES*>SP$LIB.RUN
    LIBRARIES*>ECL$LIB.RUN
    LIBRARIES*>SOCKET.RUN
    LIBRARIES*>DSMSITLIB.RUN
    LIBRARIES*>PRIFORMA_EX_LIBRARY.RUN
    LIBRARIES*>TRANS_LIB$PRC.RUN
    LIBRARIES*>CONFIG_USERS_LIB.RUN
    TPTOOLS>TOOLS_LIBRARY>TZCDMMP2.RUN
    LIBRARIES*>INFO_LIBRARY.RUN
    EMACS*>LIBRARIES*>COMMON.PP.RUN
    EMACS*>LIBRARIES*>MODULA_LM.RUN
    EMACS*>LIB>LIBRARIES*>SHI_LIBRARY.RUN
OK,
```

## Search Order

The order in which the search rules are listed in the ENTRY$ search list is the order in which PRIMOS searches the libraries to find a match to a subroutine entrypoint. Typically, the order indicates that systemwide library EPFs (in the LIBRARIES* directory) are to be searched first (after internal PRIMOS entrypoints, which are always searched before any libraries listed in the entrypoint search list).

At some point, the search list usually contains the search rule -STATIC_MODE_LIBRARIES, which directs that the static-mode libraries are to be searched. Although Prime supplies several individual static-mode libraries, these libraries are treated as one entity.

Because the order of the search rules determines the order in which the libraries are searched, a proper ordering improves the speed at which the subroutine is found. The library for a frequently called subroutine should be listed so that it requires the shortest search time possible.

The search order is also important when naming conflicts occur between libraries. The order in which the conflicting entrypoints appear determines which copy of a subroutine is actually invoked. It is best, of course, to avoid such naming conflicts altogether.

At Rev. 22.1, a new command became available to System Administrators to help in determining whether your existing search rules are optimum or even appropriate for your system. The command is MONITOR_SEARCH_RULES (MSR) and is documented fully in the *Operator's Guide to System Commands*. It's a good idea to use this command to monitor your search rules before you modify them.

## Access Rights on the Search List

Set access rights on the system entrypoint search list SEARCH_RULES*>ENTRY$.SR so that only you (or someone designated by you) can modify the file. You might use the following access rights:

```
SYSTEM: ALL
$REST:LUR
```

## User Entrypoint Search Lists

Users can create their own entrypoint search lists and enable them with the SET_SEARCH_RULES command. A user's entrypoint search list automatically includes SEARCH_RULES*>ENTRY$.SR unless the user's command line contains the -NO_SYSTEM option and the user's list does not include -SYSTEM.

Users can display their current entrypoint search list by using the LIST_SEARCH_RULES (LSR) command.

If there are a number of libraries, each of which is used by only a few people, it may be better for those people to have their own entrypoint search lists, rather than for those libraries to be in the default search list. The latter situation would require all users to have enough dynamic segments to map in everything on the entrypoint search list, or risk getting the error condition `Not enough segments.` (The EDIT_PROFILE or CONFIG_USERS command sets the number of dynamic segments per user.) A short entrypoint search list usually results in better performance.

**Caution**

You should usually encourage users to use the system copy of ENTRY$.SR (which is obtained automatically) rather than maintaining and using a private copy. If a user needs a private copy, the user should do one of the following:

- To include system rules at the beginning of the list, use the SET_SEARCH_RULES command without the -NO_SYSTEM option.
- To put the system rules other than at the beginning, put the keyword -SYSTEM in the list if the list does not contain any of the rules in the system copy of ENTRY$.SR.

Some users may desire to have search rules that are unrelated to the system copy of ENTRY$.SR. (They would do this by using the -NO_SYSTEM option to SET_SEARCH_RULES, and by omitting the -SYSTEM rule from their ENTRY$.SR files.) You should make them aware that it is their own responsibility to keep their ENTRY$.SR files up to date if the system copy is changed.

## Linkage Faults

If the end of the search list is reached without the target subroutine having been found, or if the ENTRY$ list has been altered or improperly installed, the dynamic linking mechanism signals the condition LINKAGE_FAULT$. The linkage fault normally produces an error message such as the following:

```
Error: condition "LINKAGE_FAULT$" raised at 4243(3)/1031.
Entry name "GET_LINE" not found while attempting to resolve
dynamic link from procedure "FIND_NUM".
ER!
```

The following steps should remedy the condition:

1. The user should enter the following command to reinitialize the search rules to system default:

   ```
   SET_SEARCH_RULES -DEFAULT ENTRY$
   ```

   If the user can now perform the operation that caused the linkage fault without generating an error message, the user's private entrypoint search list contains an error. If the user repeats the operation and the linkage fault occurs again, perform Step 2.

2. You should enter the following command to display the entrypoint search list:

   ```
   LIST_SEARCH_RULES ENTRY$
   ```

   Check the displayed list. The rule -PRIMOS_DIRECT_ENTRIES should be at the top. If it is not, check that a file named SEARCH_RULES*>ADMIN$.ENTRY$.SR exists, and that it contains only the rule -PRIMOS_DIRECT_ENTRIES. If not, create that file, or correct it, and try the operation again.

**Note**

The LIST_SEARCH_RULES command does not display the SEARCH_RULES*>ENTRY$.SR file, but displays a list stored in memory. The rule -PRIMOS_DIRECT_ENTRIES should appear in the displayed list and in the SEARCH_RULES*>ADMIN$.ENTRY$.SR file, but not in the SEARCH_RULES*>ENTRY$.SR file.

If the linkage fault persists, a library name may be missing from the SEARCH_RULES*>ENTRY$.SR file. Check that all the libraries necessary to execute the program causing the linkage fault are listed in the ENTRY$.SR file. Add the pathnames of any missing libraries to the end of the file. Check that the list contains no typographical errors and that all the pathnames are correct. If the pathname is for a remote file, check that the line is up, and the disk is added. (It is recommended that target libraries be stored on the local system to improve performance.) If you change the ENTRY$.SR file, perform Step 1 again.

## ATTACH$ SEARCH RULES

ATTACH$ search rules let you predetermine the locations of file system objects when you use unqualified pathnames. Before Rev. 23.0, PRIMOS searched only the directories in the MFD of each specified disk partition. At Rev. 23.0, PRIMOS can search any directory, no matter at what level in the file system hierarchy the directory resides.

An ATTACH$ search list is, in effect, a list of pathname prefixes. When encountering an unqualified pathname, PRIMOS transforms it into a fully-qualified pathname by using ATTACH$. PRIMOS adds each ATTACH$ search rule, one at a time, to the beginning of the unqualified pathname and then checks the validity of the new pathname. If the now fully-qualified pathname is an actual directory, the search is over. If the pathname is not valid, the search continues until the directory is found or the ATTACH$ search list is exhausted.

Before Rev. 23.0, the only valid ATTACH$ search rules were disk partition names and valid keywords, including the special search rule called **-added_disks** (described later in the section **-added_disks**). Disk partition names had to be enclosed in angle brackets.

```
<DISKA>
```

At Rev. 23.0, ATTACH$ search rules have been generalized so that any directory can be specified as a valid search rule. The directory can be anywhere in the file system hierarchy, from the highest level (the root directory) to the lowest level. This means that many directories on the same partition can be specified as search rules.

The following shows an example of a pre-Rev. 23.0 ATTACH$ search list:

```
<sysdsk>
<wrkdsk>
<bckdsk>
-added_disks
```

When a user specifies a directory name, PRIMOS prefixes the directory name with each disk partition name in the search list in the sequence specified. PRIMOS stops searching when the first directory with the name the user requested is found.

**Note**

The EXPAND_SEARCH_RULES (ESR) command can be used to determine the result of using the ATTACH$ search list to convert an unqualified pathname into a fully-qualified pathname. For example, issuing the following command

```
EXPAND_SEARCH_RULES MYDIR
```

might yield the fully-qualified pathname <SYSDSK>MYDIR.

At Rev. 23.0, the ATTACH$ functionality has been expanded so that a search list might appear as follows:

```
<sysdsk
<wrkdsk>myproj>mywork
<bckdsk
<newdsk>info
<
```

Each rule in the above search list is explained below.

**<sysdsk** — The disk partition (**<sysdsk>**) is searched first.

**<wrkdsk>myproj>mywork** — The lower-level directory **mywork** is searched next. Note that this search rule is a fully-qualified pathname. This type of search rule allows you to search any level in the file system hierarchy.

**<bckdsk** — The disk partition **<bckdsk>** is searched next.

**<** — Finally, the root directory symbol indicates that all disk partitions in the root directory are searched.

If users do not set their own ATTACH$ search lists, the ATTACH$ search list defined by the System Administrator is in effect. This list resides in the file **<cmdnc0>SEARCH_RULES*>ATTACH$.SR** on the command device. If this file does not exist, PRIMOS simply uses the **-added_disks** keyword, which indicates that all disk partitions are to be searched.

The ATTACH$ search list can be invoked automatically by other search lists. This use of ATTACH$ is described at the end of this section under the heading ATTACH$ Invoked by Other Search Lists.

## -added_disks

The -added_disks keyword is used only in ATTACH$ search lists. Before Rev. 23.0, the PRIMOS file system used -added_disks to search all of the added disk partitions to resolve an unqualified pathname. PRIMOS used the local disk table to determine which disk partitions to search.

At Rev. 23.0, however, the file system name space is organized as a singly-rooted tree hierarchy (described in Chapter 4), which has the ability to accept disk partitions mounted at different levels in its structure. This means that the use of -added_disks is no longer as straightforward as it once was. In addition, if the Name Server is running on your system, the number of added disk partitions in the common file system name space can grow to be quite large.

Therefore, the -added_disks keyword at Rev. 23.0 will have different meanings, depending on whether or not the Name Server is running. The different meanings of -added_disks are discussed in the sections following.

**-added_disks Without the Name Server:** If the Name Server is not running, PRIMOS uses the disk table to determine which disks should be searched. First, the local disks are searched in the order in which they appear in the disk table. Second, the remote disks are searched in the order in which they appear in the disk table.

**Note**

Disk partitions that are not mounted in the root directory are not searched with -added_disks. Their fully-qualified pathnames must be individually added to the list in order to be searched.

**-added_disks With the Name Server:** If the Name Server is running, PRIMOS uses the disk table and the Global Mount Table (GMT) to determine which disks will be searched. (Use the LIST_MOUNTS command, described in the *Operator's Guide to System Commands*, to examine the contents of the GMT on your system.) First, the local disks are searched in the order in which they appear in the disk table. Second, PRIMOS uses the GMT to determine the remote disks to be searched.

If your system is running the Name Server, you might not want to use the -added_disks keyword under certain circumstances. Following are factors to consider when you are deciding whether to use -added_disks on a system that has the Name Server running:

- The order of the GMT is determined by the Name Server's internal replication algorithm. This order may change over time as disks on local and remote systems are added and shut down. Thus, it is not possible to directly affect the order in which remote disks are searched using -added_disks. If your system is running the Name Server, and the order in which the disks are searched is important to you, then you should explicitly define those disks in an ATTACH$ search list without the -added_disks keyword. If you do not define your own ATTACH$ search list, then -added_disks is used by default.
- The disk partitions that are mounted lower than the root directory will not be searched. If you want lower-level partitions searched, then you must explicitly add their fully qualified pathnames to your ATTACH$ search list.

- The -added_disks keyword is usually specified as the last search rule in an ATTACH$ search list, and this keyword causes PRIMOS to search all disk partitions mounted in the root directory, including those that have already been searched using previous ATTACH$ search rules. At Rev. 23.0, a common file system name space (and therefore root directory) can contain as many as 1280 disks, and this can significantly affect the performance of searches using -added_disks. For this reason, it is recommended that you do not use the -added_disks keyword when the Name Server is running on your system and your system is part of a large network.

### ATTACH$ Invoked by Other Search Lists

The only search list that requires that pathnames be fully-qualified is the ATTACH$ search list. The other system search lists can contain pathnames which are unqualified. In order to resolve unqualified pathnames found in other system search lists, PRIMOS uses the ATTACH$ search list.

Adding unqualified pathnames to search lists can greatly affect their performance, especially if there are many partitions to search. Searching for a file system object with an unqualified pathname is always slower. You should carefully consider the tradeoff between the flexibility of unqualified pathnames and the better performance of fully-qualified pathnames.

## COMMAND$ SEARCH RULES

You use the COMMAND$ search list to search directories for command files. A command file is any executable code file, such as a runfile or CPL file. A COMMAND$ search list should contain the pathnames of the directories that you wish to search for executable code files. The following are typical search rules for a COMMAND$ search list:

```
cmdnc0
glenn
glenn>project
alan>project
glenn>project>tests
glenn>status
```

The default for COMMAND$ is the directory CMDNC0, which contains the executable code files for PRIMOS commands. This default permits you to execute PRIMOS commands without supplying complete pathnames.

Once you have created a COMMAND$ search list, you can execute a command file by simply typing its name, as if it were a PRIMOS command. For example, if you include the search rule *mydir>subdir* in your COMMAND$ search list, you can execute the file *mydir>subdir>mycfile*.run from any attach point by simply typing *mycfile*. You do not have to specify the RESUME command or the filename suffix. PRIMOS searches each listed directory in sequence. PRIMOS stops searching when it finds the first file with the name

you requested and (in order of preference) the suffix .RUN, .SAVE, .CPL, or a static-mode runfile with no suffix.

You can also use the EXPAND_SEARCH_RULES (ESR) command to search the COMMAND$ search list. If you instruct ESR to use the COMMAND$ search list, you do not have to specify the .RUN, .SAVE, or .CPL filename suffix. If you instruct ESR to find a filename with a .RUN, .SAVE, or .CPL suffix, you do not have to specify use of the COMMAND$ search list. ESR returns the absolute pathname of the command file, including its suffix.

# 4

# PLANNING YOUR FILE SYSTEM STRUCTURE AT REV. 23.0

At Rev. 23.0, you can create a logical file structure uniquely tailored to the needs of your particular site. This chapter explains the changes to the file system at Rev. 23.0 (also discussed from slightly different perspectives in the *PRIMOS User's Release Document* and the *Advanced Programmer's Guide II: File System*) and shows how you can use the new file system to create a more flexible and easy-to-use environment for users.

## WHAT IS DIFFERENT ABOUT THE REV. 23.0 FILE SYSTEM?

The primary changes to the file system at Rev. 23.0 are

- The switch from a multi-rooted to a singly-rooted file system name space
- The independence of pathnames and disk names
- The ability to logically mount disk partitions

Each of these changes is discussed in the following sections.

### Singly-rooted File System Name Space

At Rev. 23.0, the PRIMOS file system has been transformed from a multi-rooted file system name space to a singly-rooted file system name space.

A **file system name space** is a collection of unique names (known as pathnames) of all file system objects that can be referenced by a given system.

**Multi-rooted Name Space:** Before Rev. 23.0, the uppermost level of the file system hierarchy was comprised of many disk partitions. As shown in Figure 4-1, each partition was the root of a separate file system tree.

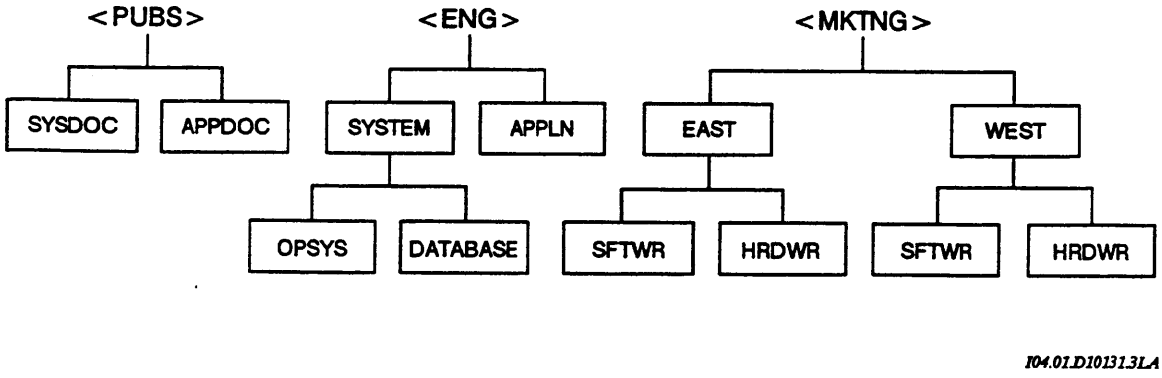


*FIGURE 4-1. Pre-Rev. 23.0 Multi-rooted Name Space*

Although, for the sake of simplicity, only three partitions are shown in the above diagram, a system usually contains many more than three partitions. Each partition in a pre-Rev. 23.0 system is at the highest level of the file system hierarchy. Fully-qualified pathnames begin with the partition name. In effect, each partition forms the root of its own file system name space because all pathnames beginning with a given partition are unique. The term multi-rooted is therefore applied to a file system name space where there are many partitions (roots) at the top level of the hierarchy.

**Singly-rooted Name Space:** Figure 4-2 shows that, at Rev. 23.0, there is only one root directory at the uppermost level of the file system hierarchy instead of many disk partitions. The root directory (designated by <) contains all the disk partitions in a hierarchical arrangement. This structure is called a singly-rooted file system name space because all file system objects, no matter where in the structure they are located, stem from this single root directory instead of any one of a number of partitions. Fully qualified pathnames start not with the disk partition, as in the multi-rooted name space, but with the root directory. Each pathname beginning with the root directory is unique.

The root directory contains only other directories (also known as root entries) that represent the MFDs of partitions. Note that in Figure 4-2 below, all of the entries in the root directory (PUBS1, ENG1, and MKTNG1) appear as directories (indicated by the rectangles), not as disk partitions (indicated by the angle brackets). (Contrast this with Figure 4-1, in which all top-level entries are shown as disk partitions.) Although the directories represent disk partitions, they are known to the logical file system as directories. Such disk partitions are known as logical mounts. More information on logical mounts is given later in this chapter.

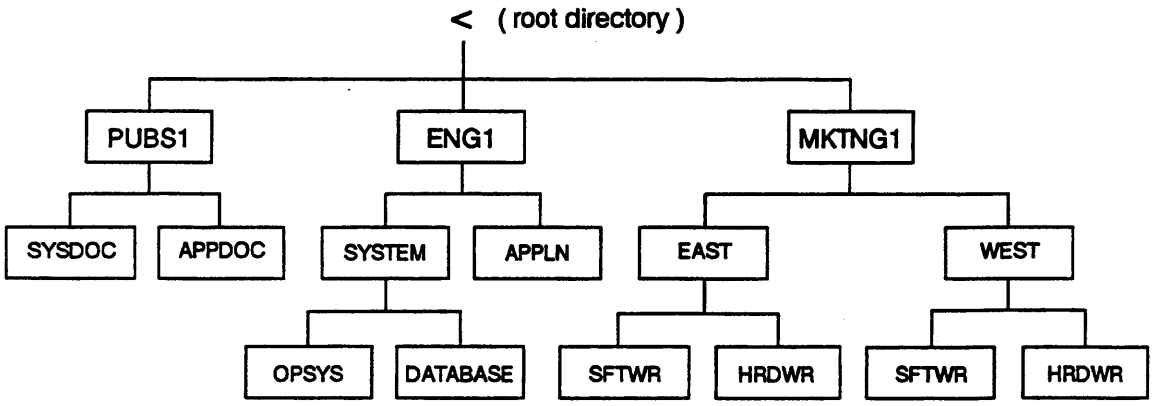


*FIGURE 4-2. Rev. 23.0 Singly-rooted Name Space*

## Independence of Pathnames and Disk Names

The root directory consists of directories that represent disk partitions. The directory names are, by default, the same as the disk names. (Disk names are the six-character names assigned to the disk when formatted with MAKE.) You can, however, add disks to your system with **mount-point pathnames** that use a separate directory name (with as many as 32 characters) to identify the disk partition to the logical file system. Refer to the ADDISK command in the *Operator's Guide to System Commands* for instructions on adding disks with mount-point pathnames. Mount-point pathnames are further explained later in this chapter.

Adding disks with mount-point pathnames enables you to assign more meaningful names to the disks than is possible with the short disk names. The disk names then become irrelevant to users and applications (although they still appear in the local disk table for compatibility with pre-Rev. 23.0 systems).

This is beneficial in situations where the disk name that might be understandable for Operators and Administrators may mean nothing to users. In such a case, the independence of disk names and pathnames names at Rev. 23.0 means that the shorter disk name can still be specified in a way that is convenient for Operators and Administrators, for example, <SYSA01>, but the disk can be added to the system with a directory name that clearly identifies its contents for users, for example, <MARKETING_EAST.

**Note**

When a disk is added to the system with a mount-point pathname, users and applications *must* reference the disk using the mount-point pathname instead of the disk name, despite the fact that the disk name still appears in the local disk table.

## Logical Mount

At Rev. 23.0, you can use the -MOUNT_PATH option to ADDISK to mount a disk partition either in the root directory with a mount-point pathname or over any local, existing directory below the root (so long as you do not mount over the MFD of a partition). Mounting a partition in either of these two ways is known as a **logical mount.** The ability to mount disk partitions logically allows the Administrator to nest partitions and to organize the file system so that it is easier to use, maintain, and expand. An example of how this works is given in the following section, How to Use the Rev. 23.0 File System. (Refer to the ADDISK command in the *Operator's Guide to System Commands* for details on using the -MOUNT_PATH option.)

**Mounting in the Root:** When you add a disk partition to the root with a mount-point pathname, a directory with the name you select for that partition is created in the root. For example,

```
ADDISK 4160 -MOUNT_PATH <ACCOUNTING
```

In this example, pdev 4160 is added to the root directory with the mount-point pathname <ACCOUNTING. Typing an LD at the root directory shows a new directory named ACCOUNTING. This directory represents the MFD of pdev 4160 and if you attach to <ACCOUNTING, you are at that MFD.

At Rev. 23.0, if you do not specify a mount-point pathname when adding a partition, the partition is automatically added in the root and the name of the directory in the root is, by default, the same as the name of the partition. When users attach to the directory, they are attached to the MFD of the newly added disk partition.

---

**Caution**

If you have pre-Rev. 23.0 systems in your file system name space, do not specify a mount-point pathname when adding a partition to the root directory. If you do, pre-Rev. 23.0 systems cannot access those partitions, as they only recognize the partition name.

---

**Mounting Lower in the Tree Structure:** You can mount disks below the root using the following format.

```
ADDISK 4062 -MOUNT_PATH <ORCHESTRA>INSTRUMENTS>VIOLIN
```

In this case, the mount-point pathname of pdev 4062 is <ORCHESTRA>INSTRUMENTS>VIOLIN. That is, pdev 4062 is added over the previously existing directory VIOLIN. Users who are attached to the original directory or its parent directory (in this case, VIOLIN or INSTRUMENTS respectively) at the time the logical mount is done continue to see the contents of the original directory until they leave that attach point. The original contents of the directory VIOLIN must be moved to the new partition. The last part of this chapter shows you the steps to follow when mounting a disk partition over an existing directory.

# HOW TO USE THE REV. 23.0 FILE SYSTEM

The new capabilities of the file system at Rev. 23.0 give an Administrator more flexibility to create a file structure that is

- more intuitive for users
- easier to maintain
- easier to expand

This section presents an example of how one corporation might use logical mounts to make the most of the new file structure. The situation at each site is different, of course, and Administrators must use their own judgement and experience in determining how much, how little, and in what ways to use the new capabilities. This example is meant simply to clarify points made earlier and to present the concepts in a concrete way.

---

**Caution**

Generally, you should use the new capabilities to expand on the file structure you have already established and not to totally rearrange the existing structure. Making radical changes in your existing structure could cause confusion for users who are used to accessing files a certain way. Changes to the file system also have implications for all applications that use pathnames. Carefully think through all changes you intend to make because every change can have far-reaching effects or subtle effects that you may not anticipate.

---

## Using the Rev. 23.0 File System Capabilities – An Example

At Rev. 23.0, you can expand your physical file system capacity without changing the logical structure; that is, the fully-qualified pathnames of file system objects remain constant. The ability to mount disk partitions logically allows you to graft entire partitions anywhere in the tree structure where more space is needed. Instead of being restricted by the physical capacity of a single partition, you can arrange disks so that users can store or access large amounts of data using a consistent pathname regardless of how much physical space is required.

For example, suppose that the Phoenix Corporation, a young and growing company, has a pre-Rev. 23.0 file system like the one shown in Figure 4-3. This illustration shows that the Phoenix Corporation has one disk partition named PHNX. This partition is divided into directories and files that contain all the corporation's data, organized much the way the Phoenix Corporation itself is organized.

The company has expanded a great deal, however, and there is no longer enough room on one disk partition for all of their data. The Administrator needs to add another disk partition, but would like the logical structure (and therefore, fully-qualified pathnames) to remain the same.

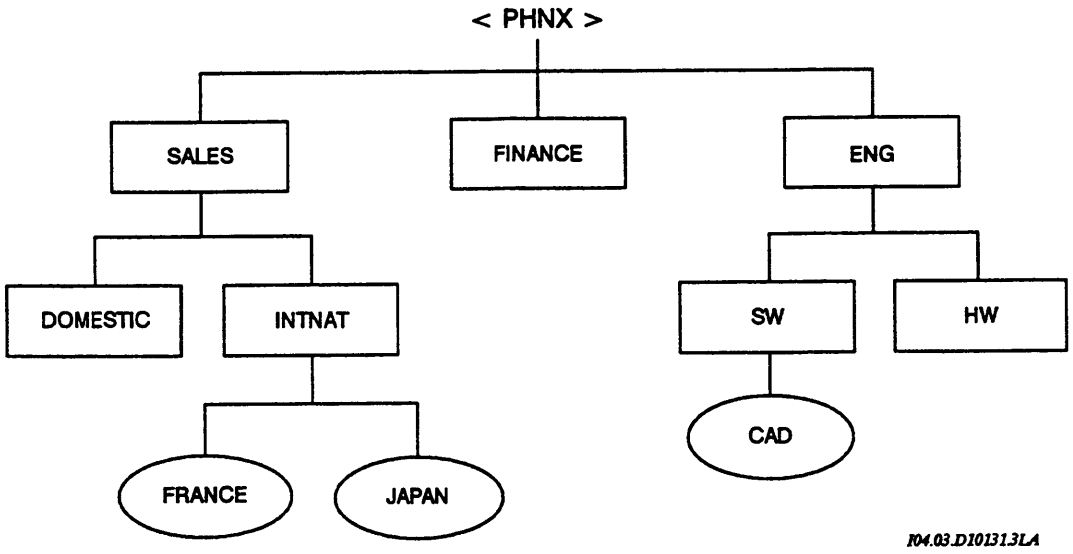


*FIGURE 4-3. Pre-Rev. 23.0 Single Partition File Structure*

On a pre-Rev. 23.0 system, the only choice the System Administrator has is to add another disk partition <PHNX2> at the top level of the hierarchy, as shown in Figure 4-4.

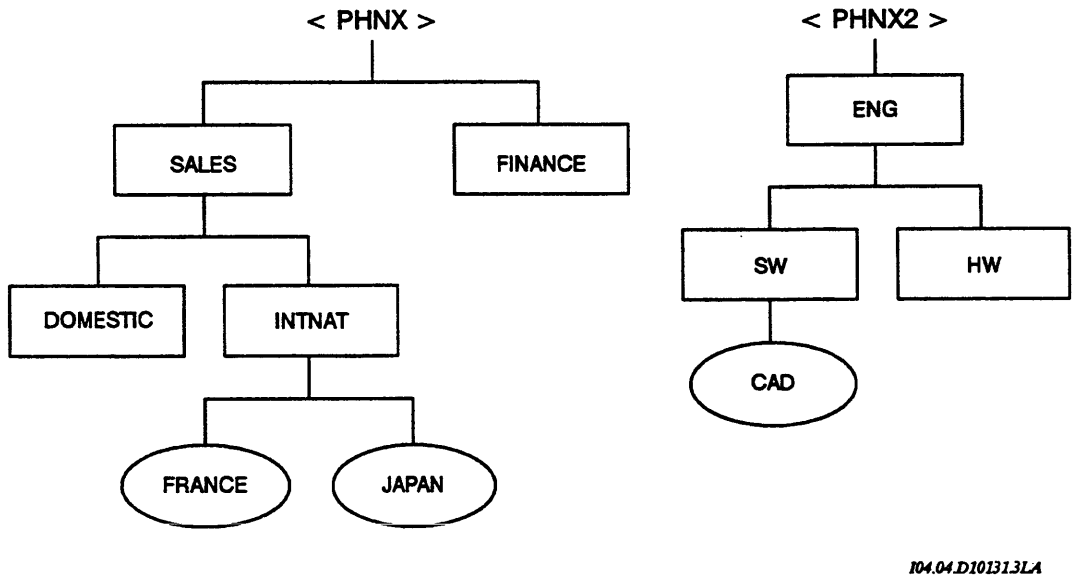


*FIGURE 4-4. Disk Partition PHNX2 Added at Top Level of Hierarchy*

With the addition of the new disk, users and applications must specify a different pathname in order to reach the file CAD. The original pathname was

```
<PHNX>ENG>SW>CAD
```

The new pathname is

```
<PHNX2>ENG>SW>CAD
```

**Note**

Trying to get around this situation by using unqualified pathnames (for example, ENG>SW>CAD) is not a good solution to the dilemma. The reason for this is explained later in this chapter in the section entitled Using Fully Qualified vs. Unqualified Pathnames.

With the upgrade to Rev. 23.0, logical mounts allow the Administrator to logically graft the new partition <PHNX2> over the ENG directory using the following command line.

```
ADDISK 4061 -MOUNT_PATH <PHNX>ENG
```

**Note**

Refer to the last part of this chapter for detailed steps for expanding storage space using a logically mounted disk partition.

Adding the disk partition at the pathname <PHNX>ENG, as shown in Figure 4-5, creates much more storage space at that pathname location.

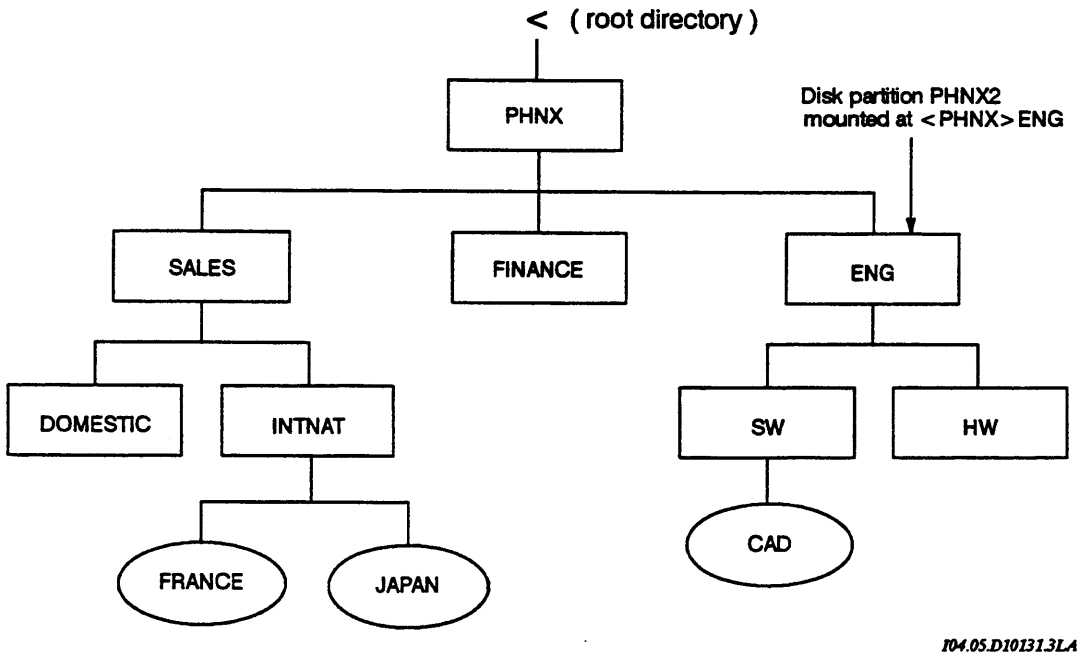


*FIGURE 4-5. Disk Partition PHNX2 Grafted Over Existing Directory*

Figure 4-5 shows that you can change the physical location of data without changing the logical location. Although the contents of ENG are now physically located on the new disk PHNX2 instead of on PHNX, the logical structure of the file system has not changed and users can continue to specify the original pathname to access much more data. (Compare Figure 4-3 with Figure 4-5; except for the presence of the root directory in Figure 4-5, the structure is the same.)

As the organization grows, the same can be done with other directories, for example, the SALES directory. Figure 4-6 shows the structure after the Administrator adds the new disk partition PHNX3 over the directory SALES.

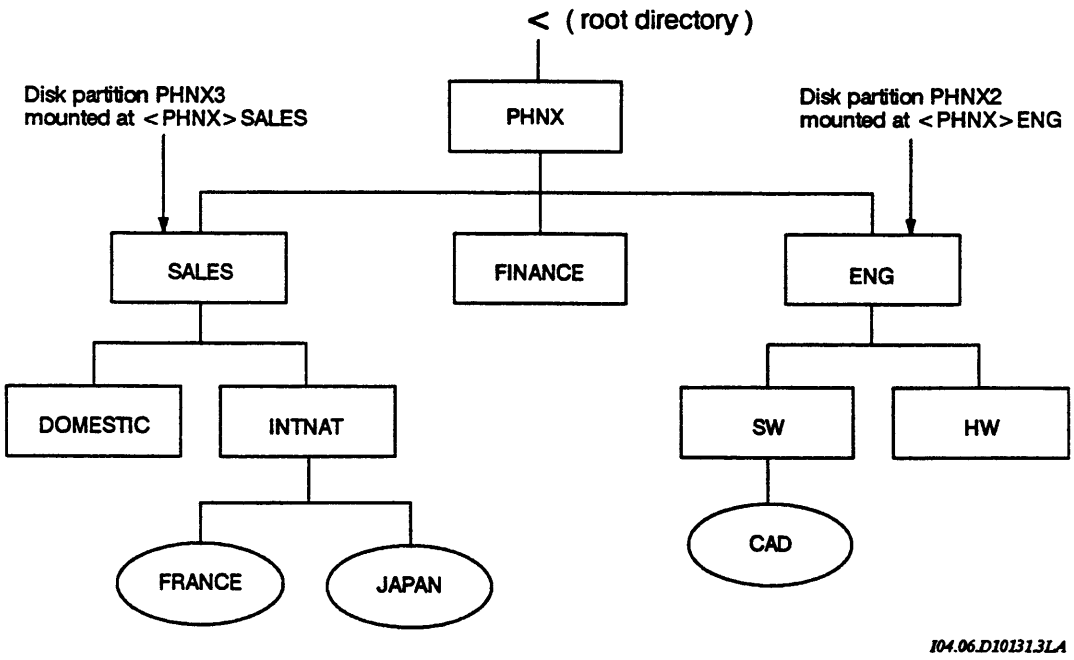


*FIGURE 4-6. Disk Partition PHNX3 Grafted Over Existing Directory*

For users and applications, nothing changes. The original pathname to get to the SALES directory was

```
<PHNX>SALES
```

The new pathname to get to the SALES directory is still

```
<PHNX>SALES
```

even though SALES is now located on disk PHNX3 instead of disk PHNX. Much more sales data can now be stored and accessed using the original pathname.

## FILE SYSTEM PLANNING CONSIDERATIONS

The new file system capabilities make long-range planning of your file structure much more important. In fact, you can only truly take advantage of the new features if you plan carefully. You should consider the following points before you make any changes or additions to your file structure.

## Using Fully Qualified vs. Unqualified Pathnames

It has always been necessary when expanding storage space, to change fully qualified pathnames to reflect the new physical location of the data. To get around this, some sites use unqualified pathnames in their applications so that, when the disk partition changes, the pathname does not. Using the example of the Phoenix Corporation, the site might use the pathname

```
ENG>SW>CAD
```

to reach the file CAD instead of the fully qualified

```
<PHNX>ENG>SW>CAD
```

There are two problems with this approach:

- The attach-scan PRIMOS goes through to resolve unqualified pathnames can contribute to poor system performance for sites with many disk partitions. Use of fully qualified pathnames is much quicker and more efficient because PRIMOS does not have to scan many disk partitions before finding the correct one.
- Unqualified pathnames may not be unique and thus can cause programs to fail. If there are directories or files with the same names on more than one partition, you may be directed to the wrong attach point. Such misdirection can occur for a variety of reasons, including disk errors during an ADDISK, operator errors in creating the disk table or ATTACH$.SR, remote systems that are down, or intermittent network errors. All of these situations could cause a process to find a different disk than was intended and attach to that point rather than the desired directory or file.

The new capability of adding partitions anywhere in the tree structure makes it easier to use fully qualified pathnames without changing them every time you require more space. Even if you need to rearrange existing disks, changing the pathnames is a one-time occurrence at Rev. 23.0.

## Specifying Mount Points in the Root Directory

When you mount a disk partition in the root directory using the -MOUNT_PATH option

- A new directory is created in the root with the name you specify, for example, <PAYROLL. (If you do not use the -MOUNT_PATH option to ADDISK, the disk is mounted in the root with its disk name by default.)
- The directory name of a partition must be unique, that is, it cannot already exist in the root.
- Users and applications cannot specify the disk name (the name given to the disk when formatted with MAKE) in pathnames when that disk has been added with a directory name. The directory name must be used instead.
- The directory name you choose may contain as many as 32 characters.

## Specifying a Local Directory Below the Root as the Mount-point Directory

When you mount a disk partition over an existing directory, the existing directory is known as the **mount-point directory.** The following rules apply in such a circumstance.

- The mount-point directory must be an existing directory on the local system.
- The mount-point directory can be at any level in the hierarchy but it cannot be the MFD.
- The supervisor terminal must have a minimum of Use (U) access rights to the mount-point directory.
- Attaching to the mount-point directory puts users in the MFD of the newly added disk partition.
- The ACLs governing access to directories on the newly added disk are determined by the ACLs protecting the mount-point directory and then by the ACLs on the MFD of the newly added disk. Remember to check the existing ACLs on the mount-point directory and adjust them if necessary.
- Quota settings should be modified when a disk partition is logically grafted into an existing tree structure. This is because the records originally allocated to the mount-point directory become available when the contents of that directory are copied to the newly added disk partition and then deleted from the mount-point directory. The records previously used by the mount-point directory should be redistributed among remaining directories on the original disk partition.

## Additional Considerations When Mounting Disk Partitions

Keep these points in mind when adding a disk partition in the root directory or anywhere else in the tree structure:

- You cannot add disk partitions with the -MOUNT_PATH option if programs already exist on your system that contain fully-qualified pathnames for the given disk. The existing pathnames would no longer be accurate. If you still want to add such partitions using the -MOUNT_PATH option, you must remember that all affected fully-qualified pathnames used in applications or by users must be changed accordingly.
- If existing disk partitions contain user initial attach points, they can be added to the file system hierarchy with mount-point pathnames only if those initial attach points are changed with EDIT_PROFILE or CONFIG_USERS to reflect the new pathnames.
- The command device can be added only in the root directory. It can be added to the root with a directory name (instead of the shorter disk name) only if you add the directory name to the COMDEV configuration directive. (See Chapter 8 of this guide for more information on the COMDEV configuration directive.)
- Operators must use a different procedure to do disk backups when disks are logically mounted. MAGSAV cannot cross logical mount points; that is, disk partitions grafted into the tree structure will not be backed up unless the new procedure is followed. (Refer to the *Operator's Guide to Data Backup and Recovery* for more information.)

## Procedure for Expanding Storage Space Using Logical Mounts

When you expand storage space using logical mounts, follow the steps outlined below. In this example, disk PHNX2 is physical device number 4061. (This is the same example as the one used to explain Figure 4-4.)

1. Add disk PHNX2 to the root directory on your system.

   ```
   ADDISK 4061
   ```

2. Copy the contents of the ENG directory to the newly added disk PHNX2 using the options shown on the command line below. (The options -RPT and -NQ are not necessary but they can be helpful.)

   ```
   copy <PHNX>ENG <PHNX2>MFD -merge -ca -rpt -nq -no_check
   ```

3. Delete the contents of the ENG directory from the PHNX disk.

   ```
   delete <PHNX>ENG>@@ -nq -rpt
   ```

   (The -NQ option is important here; otherwise you are queried about everything on the disk before it is deleted. The -RPT option is not necessary but may be helpful.)

4. Shut down disk partition PHNX2.

   ```
   SHUTDN 4061
   ```

5. Add disk PHNX2 with the correct mount-point pathname.

   ```
   ADDISK 4061 -MOUNT_PATH <PHNX>ENG
   ```

# 5

# INITIALIZING SYSTEM SOFTWARE

This chapter contains information on ensuring the proper initialization of shared system software. It discusses the following topics:

- Shared segments, including a table of the segments to which Prime has assigned products, the segments reserved for Prime, and the segments specifically reserved for customer use
- Shared static-mode libraries, including a table of the shared static-mode library package numbers
- Library EPFs
- Registered EPFs

Programmers and analysts who need to know the details of shared segments, EPFs, registered EPFs, and library EPFs should read the *Advanced Programmer's Guide I: BIND and EPFs.*

## SHARED SEGMENTS

Shared subsystems normally are initialized at cold start by the PRIMOS.COMI file. After cold start, you can initialize individual shared subsystems supplied by Prime by using this command at the supervisor terminal:

**CO SYSTEM>*name*.SHARE.COMI**

Software supplied by other vendors may use the segments reserved for customers. See Table 5-1. As System Administrator, you are responsible for assigning and coordinating the use of these segments. For shared subsystems not supplied by Prime, use this command from the supervisor terminal:

**SHARE *pathname segment-number [access-rights]***

where *pathname* is the pathname of the file whose contents are to be loaded into segment *segment-number*, and *segment-number* is the octal number of the segment to be shared. See Table 5-1 for a list of segments specifically reserved for customer-shared subsystems.

*access-rights* is a number that specifies user access to the segment. The valid values for *access-rights* are as follows:

| | |
|---|---|
| 0 | No access |
| 200 | Read access only |
| 600 | Read and execute access (default) |
| 700 | Read, write, and execute access |

See the *Operator's Guide to System Commands* for details on the SHARE command.

---

**WARNING**

If you use the SHARE command incorrectly, the result may be that user programs can overwrite the operating system and the shared utilities. Do not share into segments 0 - $1777_8$, which are reserved for PRIMOS. Other segments that may contain system utilities are listed in Tables 5-1 through 5-4.

---

*TABLE 5-1. Contents of Shared Segments at Rev. 23.0*

| *Segment* | *Product* |
|---|---|
| 2000 | ED |
| 2001-2003 | DBMS |
| 2004-2011 | SPSS |
| 2012 | DBMS |
| 2013 | BASIC/VM |
| 2014 | Reserved for Prime |
| 2015 | DPTX |
| 2016 | Reserved for Prime |
| 2017 | BASIC/VM |
| 2020 | Reserved for Prime |
| 2021 | FORMS library |
| 2022-2023 | Reserved for Prime |
| 2024-2025 | PRIME/POWERPLUS |
| 2026-2027 | FTS |
| 2030-2037 | **Reserved for customers** |
| 2040-2042 | DBG |
| 2043 | SPSS |
| 2044-2056 | Reserved for Prime |
| 2057-2065 | OAS |
| 2067 | Reserved for Prime |
| 2070 | DBMS |
| 2071 | OAS |
| 2072 | SPSS |
| 2073-2077 | DISCOVER |
| 2100 | EDMS |
| 2101 | OAS |
| 2102-2114 | EDMS |
| 2115 | DBG |
| 2116-2121 | Reserved for Prime |
| 2122-2125 | MIDASPLUS |
| 2126-2127 | FTS |
| 2130-2137 | PRIME MEDUSA |
| 2140 | EDMS, BP99 |
| 2141-2150 | Reserved for Prime |
| 2151-2153 | FED |
| 2154-2161 | CBL |
| 2162-2163 | EDMS, BP99 |
| 2164-2166 | Reserved for Prime |
| 2167 | SPOOL |
| 2170-2177 | **Reserved for customers** |
| 2200-2203 | ROAM |
| 2204-2207 | PRISAM |
| 2210-2215 | ESCAPE34 |
| 2216 | Reserved for Prime |
| 2217-2220 | ROAM |
| 2221 | Reserved for Prime |
| 2223-2224 | ROAM |
| 2225 | Reserved for Prime |
| 2226 | ESCAPE34 |
| 2227 | PRISAM |
| 2230-2267 | PRIMEWAY |
| 2270-2276 | Prime INFORMATION |
| 2277 | DISCOVER |
| 2300-2317 | **Reserved for customers** |
| 2320-2321 | MIDASPLUS |
| 2322 | Reserved for Prime |
| 2323 | PRIMEWAY |
| 2324-2327 | C |
| 2330-2337 | Prime INFORMATION |
| 2340-2347 | EMACS |
| 2350-2367 | PDGS |
| 2370-2376 | PRIME MEDUSA |
| 2377 | PRIME/SNA |
| 2400-2427 | PDMS |
| 2430-2442 | THEMIS |
| 2443 | EDMS |
| 2444-2447 | Reserved for Prime |
| 2450-2467 | PRIMEWAY |
| 2470-2475 | Prime INFORMATION CONNECTION |
| 2476 | PRIME/SNA RJE |
| 2477 | Reserved for Prime |
| 2500-2521 | Prime ORACLE |
| 2522-2534 | Reserved for Prime |
| 2535 | CBL |
| 2536-2547 | Reserved for Prime |
| 2550-2556 | C |
| 2557-2564 | PDGS |
| 2565-2567 | Reserved for Prime |
| 2570-2573 | ESCAPE |
| 2574-2575 | Reserved for Prime |
| 2576 | DBG |
| 2577 | Reserved for Prime |
| 2600-2601 | ROAM/DDM |
| 2602-2665 | Reserved for Prime |
| 2666-2765 | Reserved for EPFs |
| 6001 | (Per-user linkage |
| 6006 | segments. See |
| 6007 | Tables 5-2, 5-3, 5-4.) |
| 6010 | ORACLE, EMACS, PRIMEWAY |
| 6011 | ROAM |
| 6012 | Reserved for Prime |
| 6013 | Prime INFORMATION |
| 6014 | Reserved for Prime |
| 6015 | Reserved for Prime |
| 6016 | (See Table 5-5.) |
| 6017 | ORACLE |

*TABLE 5-2. Segment 6001*

| *Allocated* | *Product* |
|---|---|
| 0-32777 | FORMS |
| 33000-66777 | Reserved for Prime |
| 67000-67767 | SPOOL |
| 67770-67777 | BATCH |
| 70000-105777 | FORMS |
| 106000-112777 | ED |
| 113000-117777 | NPX |
| 120000-131777 | ABBREV |
| 132000-177777 | FORMS |

*TABLE 5-3. Segment 6006*

| *Allocated* | *Product* |
|---|---|
| 0-37777 | FTS |
| 40000-70000 | MIDASPLUS |
| 70001-77777 | Reserved for Prime |
| 100000-177777 | ROAM/DDM |

*TABLE 5-4. Segment 6007*

| *Allocated* | *Product* |
|---|---|
| 0-47777 | ROAM |
| 50000-122777 | PRISAM |
| 123000-137777 | Reserved for Prime |
| 140000-177777 | MAGLIB |

*TABLE 5-5. Segment 6016*

| *Allocated* | *Product* |
|---|---|
| 0-37777 | Reserved for Prime |
| 40000-177777 | ORACLE |

# SHARED STATIC-MODE LIBRARIES

A system can have a maximum of 32 shared static-mode libraries. See Table 5-6 for a list of shared static-mode libraries. Each library is supplied if the customer has purchased that particular software product. The octal package numbers are used only by the initialization routines that run at cold start.

## Features of Shared Static-mode Libraries

Each user of shared static-mode library routines uses space in private segments 6001, 6006, 6007, 6010, and 6011 in addition to the segments otherwise required by programs. These segments are used for the per-user portion of the shared static-mode libraries and represent a reduction in the size of the user's load file but not in the size of the single user working set at run time. These additional segments may be compensated for by a corresponding reduction in the number of segments in the runfile.

*TABLE 5-6. Shared Library Package Numbers*

| *Package (octal)* | *Shared Library* |
|---|---|
| 1 | Reserved for Prime |
| 2 | VKDALB |
| 3 | Reserved for Prime |
| 4 | VFORMS |
| 5 | DBMSLB |
| 6 | OAS |
| 7 | EMACS |
| 10 | Reserved for Prime |
| 11 | FTS |
| 12 | Reserved for Prime |
| 13 | PDGS |
| 14 | ROAM offline |
| 15 | ROAM online |
| 17 | Prime INFORMATION |
| 20 | PRISAM |
| 21 | ESCAPE34 |
| 22 | OAS |
| 23-24 | Prime INFORMATION CONNECTION |
| 25 | PRIMEWAY |
| 26 | Prime ORACLE |
| 27-37 | Reserved for Prime |

## Initialization of Shared Libraries

Shared static-mode libraries must be initialized (with the SHARE command) each time the system is cold started. The runfiles containing the libraries to be shared are in the directory SYSTEM.

For each piece of software having static-mode libraries to be initialized, there should be a COMINPUT file in SYSTEM that contains the appropriate SHARE commands. The file PRIMOS.COMI in CMDNC0 should contain the COMINPUT commands to execute the COMINPUT files.

The PRIMOS.COMI file shares memory image files into the proper segments (see Table 5-1) and runs the programs required to inform PRIMOS that shared libraries are activated. After the libraries are initialized, users of programs linked with the special shared library object files may run V-mode and I-mode programs that access these shared libraries. If the shared libraries are not initialized, programs that expect the shared libraries to be resident receive an error message from PRIMOS whenever they attempt to access a shared library routine.

There must be no active users of a static-mode library when that library is being reshared. To ensure this when initializing a shared library, shut down PRIMOS and then reboot it.

### Rebuilding and Reinstallation

Each shared static-mode library has a set of runfiles and a command file to install the program. If only one library must be replaced, it is necessary to rebuild that library only. The library command files put all the necessary files into the directory SYSTEM so that installation is easily accomplished by running the appropriate command file.

---

**Caution**

Do not reshare a shared static-mode library while it is in use. As programs using the shared libraries execute, links are made to the appropriate shared library routines in such a way that altering the memory image in use by the program can cause random and unpredictable behavior. Changing a shared library has the effect of making such an alteration to the user's memory image. Share new static-mode libraries only when cold starting the system.

It is safe to install the memory image files into SYSTEM at any time because these files are loaded into memory only when the explicit SHARE commands are given (such as during system startup).

---

## LIBRARY EPFS

A library EPF is a set of subroutines that are bound together (with the BIND linker) into one file. Subroutines within the file that are entrypoints are available to the PRIMOS dynamic linking mechanism.

### Features of Library EPFs

Library EPFs share the following features with shared static-mode libraries:

- User runfiles are smaller, thus reducing the time required for invocation. User interaction with the program begins sooner.
- System load is reduced with respect to private segments and private memory image sizes, and paging may also be reduced. System load reduction is important for large V-mode and I-mode programs that make extensive use of system library routines.
- Installation of a new revision of the library does not require program reloading. Installation of a rebuilt library is all that is required to make the modified library available to all users of the library.

In addition, library EPFs provide the following advantages over shared static-mode libraries:

- Library EPFs are not shared into static segments, but instead are brought into memory automatically when the dynamic linking mechanism must link to an entrypoint in the library. Therefore, the System Administrator does not have to coordinate the use of shared segments among library EPFs.

- Library EPFs do not require that the system be shut down and restarted to install a new version of the program reliably. (This procedure is recommended when installing a shared static-mode library because unrecoverable errors usually result if a user is executing the old version of the program when the new version is installed.)
- Library EPFs are not loaded into the segments at cold start.
- Users can create their own library EPFs and use ACLs to restrict their use as desired.

For more information on library EPFs, see the *Programmer's Guide to BIND and EPFs* and the *Advanced Programmer's Guide I: BIND and EPFs.*

### Installation of Supplied Library EPFs

The directory LIBRARIES* contains the library EPFs supplied by Prime. Unlike static-mode libraries, these library EPFs do not need to be shared with the SHARE command.

To install a new library EPF in the LIBRARIES* directory, use the following procedure:

1. Use the COPY command to copy the EPF into LIBRARIES*.
2. Use a text editor (such as ED or EMACS) and add the pathname of the library to the system entrypoint search list. (The entrypoint search list, named ENTRY$.SR in the SEARCH_RULES* directory, is described in Chapter 3 of this guide.)

To replace an existing library EPF in the LIBRARIES* directory, use the COPY command. See the section Adding Dynamic EPF Programs in Chapter 6 for details and examples.

When you are replacing an existing library EPF, some users may be using it; that is, the existing library may be mapped into those users' address spaces. In this case, those users continue to use the existing library, and are unaware of the change. (Users invoking the library after the change get the new library.) Moreover, the existing library is not deleted, but becomes a **replace file.** Its filename suffix is changed from .RUN to .RP*n*, where *n* is a digit from 0 to 9.

If libraries are replaced frequently, you should periodically clean up the LIBRARIES* directory by deleting the old replace files (those files with the .RP*n* suffix) that are no longer in use.

## REGISTERED EPFS

At Rev. 23.0, EPFs may be either **dynamic** or **registered.** Dynamic EPFs, which may be either program or library EPFs, are stored in the file system. Dynamic EPFs are ready to execute as soon as they have been linked with BIND. PRIMOS automatically maps a dynamic EPF into special dynamic segments set aside in the private address space of the user who invokes the EPF. The only installation needed is to copy the EPF to a useful location in the file system and possibly alter users' command search rules.

Registered EPFs, which may also be either program or library EPFs, are maintained in shared address space in memory and listed in a special registered EPF database. An EPF is registered by the System Administrator. (Details on registering EPFs are given a little later in this chapter.) When an EPF is registered, PRIMOS automatically allocates space for it from available shared dynamic segments. PRIMOS also carries out many dynamic linking and initialization tasks at registration time which can make registered EPFs more efficient than non-registered EPFs for many applications.

Registered EPFs can be either Prime-supplied or customer-specific. Prime-supplied EPFs are registered automatically at cold start. The System Administrator must specifically register customer-specific applications.

---

**Caution**

Whenever you are asked to register an EPF, check first to be sure it is a legitimate EPF and that the program does not contain a virus.

---

For detailed explanation of EPFs, see the *Programmer's Guide to BIND and EPFs* for an intermediate level discussion and the *Advanced Programmer's Guide I: BIND and EPFs* for a thorough discussion.

## Benefits of Using Registered EPFs

Registered EPFs provide an efficient means to implement programs and libraries that are shared among all users on a system. Registered EPFs offer the following advantages over dynamic EPFs:

- Registered EPFs share linkage, reducing the system working set.
- Dynamic links in shared linkage are pre-snapped, reducing execution time.
- In registered EPFs, per user data is initialized faster, reducing startup time.

How much a given program can benefit from registration depends on how the program is coded. Basically, the more linkage an EPF uses, the more it can benefit from sharing linkage. However, even programs that do not generate a large amount of shareable linkage may benefit from registration. PRIMOS creates and stores an initialized copy of per user data and linkage at registration time. When a user invokes a registered EPF, this copy can be quickly mapped to the user's address space so the EPF starts up faster. In a dynamic EPF, per user linkage/data segments must be expanded from templates each time a user invokes the EPF.

Keep in mind that a registered EPF continues to occupy system resources until it is unregistered. A registered EPF remains mapped to shared segments, and PRIMOS must store information about it even if no one invokes it. Therefore, frequently and widely used programs are better candidates for registration than programs rarely used or run by only a few users. In general, linkage intensive programs tend to benefit the most from registration.

**Note**

If you maintain shared static-mode programs or libraries on your system, you should consider converting them to registered EPFs. Registered EPFs perform as well as or better than static mode versions of the same applications, and they are far easier to build, install, and maintain. See the *Advanced Programmer's Guide, Volume I: Bind and EPFs* for information on how to convert static mode code to registered EPFs.

## Registering Prime-supplied EPFs

A number of nonchargeable products and one chargeable product, MIDASPLUS, are automatically registered at cold start. For each EPF that is registered during cold start, the supervisor terminal displays the following message:

```
[REGISTER_EPF Rev. 23.0 Copyright (c) 1989, Prime Computer, Inc.]
***From PRIMOS:  EPF epf_name has been registered***
```

No action is required on the part of the System Administrator with respect to Prime-supplied registered EPFs.

## Customer-specific Registered EPFs

Users at your site must ask the System Administrator to register or unregister any customer-specific EPFs. The commands to do this are restricted to use by the System Administrator or at the supervisor terminal. The commands to use are REGISTER_EPF and UNREGISTER_EPF, and they are documented in the *Operator's Guide to System Commands*.

## Including Registered EPFs in the ENTRY$ and COMMAND$ Search Rules

In order to execute a registered EPF, you must include the -PUBLIC keyword followed by the name of the EPF in the appropriate search lists. Remember that users must include this search rule in their individual search lists as well.

- To execute registered program EPFs, a COMMAND$ search list must include the -PUBLIC keyword followed by the name of the EPF. This tells PRIMOS to search the registered EPF database for command names.
- To execute code that dynamically links to registered EPF libraries, an ENTRY$ search list must include the -PUBLIC keyword followed by the name of the EPF. This tells PRIMOS to search registered EPFs for entrynames during dynamic linking.

To make your registered EPFs available to users, be sure that the -PUBLIC keyword followed by the name of the EPF is included either in the appropriate system default search list or in the search lists of any users who must have access to an EPF.

**Note**

If a dynamic version of a registered EPF is also available on your system, PRIMOS may find and execute it before finding the registered version, depending on search rule order. If you want to be sure that registered versions of EPFs are executed in preference to dynamic versions, be sure that the -PUBLIC search rules precede any rules that lead to dynamic versions. For example, suppose you keep your own libraries in the directory MY_LIBS and you have placed a dynamic EPF version of LIB_A.RUN in MY_LIBS. If you later register LIB_A.RUN, you can be sure that programs link to the registered version by placing the -PUBLIC LIB_A.RUN search rule before the MY_LIBS>LIB_A.RUN search rule in your COMMANDS search list.

More information on entrypoint search lists is provided in Chapter 3 of this guide.

## Multiple EPF Registration

The System Administrator can register more than one EPF with the same name. This allows the System Administrator to supply a new version of a registered EPF without unregistering the old version, thus enabling the System Administrator to update a registered EPF without the possibility of corrupting a user's executing environment. As each EPF is registered, it is given a registration number. The first version of each EPF has registration number 1. Subsequent versions have higher numbers. The LIST_REGISTERED_EPF command shows the registration number of each EPF. A user who invokes a registered EPF executes the highest numbered version at the time of invocation.

## Registered EPF Access

Unlike dynamic EPFs, registered EPFs are not ACL protected since they do not reside in the file system. You should be aware that all users have access to registered EPFs on your system.

- All users can display the names of the registered EPFs on your system using the LIST_REGISTERED_EPF and LIST_EPF -REG commands.
- All users can execute registered program EPFs.
- All users can link to the entrypoints of registered library EPFs.

To restrict access to a registered EPF, the EPF should be coded so that it must be invoked by a dynamic EPF interlude. You can then restrict access by setting ACL protection on the interlude.

## Registered EPFs and DTAR3 Segments

The System Administrator must be aware that with registered EPFs there is more of a chance of exhausting the supply of DTAR3 segments. The reason for this is that registered EPFs use DTAR1 segments for procedure and DTAR3 segments for procedure links. The limit on DTAR3 segments is 64. If you register many EPFs on your system, you can use up all of the available DTAR3 segments. In this case, you will not be able to register any new EPFs. The solution to this is to unregister other EPFs that may not be as important in order to make room for the EPFs you need to register for good system performance.

# 6

# ADDING AND MODIFYING SYSTEM SOFTWARE

This chapter contains information on adding and modifying system software:

- Adding your commands to the command directory (CMDNC0)
- Using customer-defined file suffixes
- Adding and removing libraries
- Changing defaults for compilers
- Adding files to the HELP database

## ADDING COMMANDS TO CMDNC0

You can add new commands to the command directory CMDNC0. The commands must be either runfiles (that is, compiled programs linked by BIND or by LOAD) or CPL programs, but they cannot be segment directories. (You can, though, place into CMDNC0 a CPL program that uses SEG to execute a segment directory program.)

Use the COPY command to add the runfile or CPL program to CMDNC0. (You must have at least Add and Use rights to CMDNC0.)

After you install the command, users can invoke it as they would a normal PRIMOS command. For example, if you add a runfile named COMP.RUN to CMDNC0, users entering COMP at the PRIMOS prompt (OK, or ER!) run COMP.RUN, just as entering LD runs CMDNC0>LD.RUN.

For every command you add to CMDNC0, you should create a corresponding HELP file and add it to the HELP* directory. For details, see the section below, Adding HELP Files.

**Caution**

When installing a new version of a CMDNC0 command, it is recommended that you save a copy of the old version in a convenient directory. You can delete the old version after the new version is thoroughly tested and you determine that the old version is no longer needed.

## Command Suffixes

Use the following suffixes for your programs in CMDNC0:

- EPFs (V-mode and I-mode runtime programs created with BIND) must end with the suffix .RUN. This suffix is added automatically when an EPF is created with BIND.
- R-mode runtime programs should end with .SAVE. The R-mode loader, LOAD, automatically adds the .SAVE suffix. If you have R-mode programs whose names do not have the .SAVE suffix, you should add the suffix.
- CPL program names must end with the .CPL suffix.

Users do not have to type these suffixes when invoking the commands. If a user types a command, the command processor checks CMDNC0 for files in the following order:

*command-name*.RUN
*command-name*.SAVE
*command-name*.CPL
*command-name*

## Adding CPL Programs

Use COPY to put the CPL program into CMDNC0, as shown below.

```
OK, COPY NEW_PROG.CPL CMDNC0>==
```

A `File in use` or `File open on delete` error message indicates that the current copy of the CPL program in CMDNC0 is being used. Close the file from the supervisor terminal and try again.

After you add the CPL program, any user can invoke NEW_PROG.CPL by typing NEW_PROG at the PRIMOS command prompt.

## Adding Dynamic EPF Programs

You need not be concerned with whether a previously existing version of an EPF is in use before putting it in CMDNC0. Compile the program and use the BIND command to link it. (For details on BIND, see the *Programmer's Guide to BIND and EPFs.*)

Copy the EPF runfile into the directory CMDNC0 with the COPY command. COPY notes the existence of the file in CMDNC0, and asks whether you want it replaced. If the file in CMDNC0 is in use, COPY changes the name of the old version to *program*.RP*n* (where *program* is the name of the old version and *n* is a number from 0 to 9).

Any user already running the old version of the EPF is not aware of the change and continues to execute the old version. Any user who now invokes the program gets the new version. You can delete the old version when it is no longer in use.

The following example shows how to replace an in-use EPF:

```
OK, COPY NEW.RUN CMDNCO>==
EPF file "CMDNCO>NEW.RUN" already exists, do you wish to replace it? YES
New version of EPF file CMDNCO>NEW.RUN now in place.
Old version of active EPF file now named CMDNCO>NEW.RP0.
OK,
```

If there is already a file named NEW.RP0, the old version is renamed NEW.RP1. A subsequent version would create NEW.RP2, and so on, up through NEW.RP9. If all 10 old versions exist and you try to copy an eleventh version into CMDNCO, COPY queries you, as shown in the following example:

```
OK, COPY NEW.RUN CMDNCO>==
EPF file "CMDNCO>NEW.RUN" already exists, do you wish to replace it? YES
ok to delete EPF file CMDNCO>NEW.RP0? YES
New version of EPF file CMDNCO>NEW.RUN now in place.
Old version of active EPF file now named CMDNCO>NEW.RP0.
OK,
```

If all 10 old versions are in use, the replace operation is not completed, as shown in the following example:

```
OK, COPY NEW.RUN CMDNCO>==
EPF file "CMDNCO>NEW.RUN" already exists, do you wish to replace it? YES
EPF replace files are all in use.
Unable to replace file. "CMDNCO>NEW.RUN" (qry$del)
OK,
```

If you or your users frequently modify versions of runfiles in CMDNCO, you should delete unused versions from time to time to save space. Use the DELETE command as follows:

```
OK, DELETE CMDNCO>**.RP(0 1 2 3 4 5 6 7 8 9) -NO_VERIFY -REPORT -FORCE
```

## Adding R-mode Programs

R-mode programs can be written only with the FTN compiler and the PMA assembler. To install an R-mode program into CMDNCO, use the COPY command to copy the loaded runfile.

For example, if you have written a utility program called FARLEY.FTN and have compiled and loaded it, copy the program into CMDNCO as follows:

```
OK, COPY FARLEY.SAVE CMDNCO>==
```

A `File in use` or `File open on delete` error message indicates that the current copy of the program in CMDNCO is being used. Close the file from the supervisor terminal and try again.

After you add the program, any user can invoke the program by typing FARLEY at the PRIMOS command line.

## Disabling Command Line Processing

You can create commands for which PRIMOS does not process one or more of the usual command line features: wildcard options (such as -AFTER), treewalking options (such as -WALK_FROM), iteration, and special characters on the command line (such as %, @, and =). For details on these features, see the *PRIMOS Commands Reference Guide.*

Because such commands appear to the user to be the same as standard PRIMOS commands, you must inform users which of your commands perform a nonstandard processing of the command line. The next three sections describe how to change the standard command line processing for a command.

**EPF Commands:** BIND has built-in subcommands that allow the user to create EPFs that tell PRIMOS whether to process wildcarding, treewalking, iteration, and name generation on the command line. See the *Programmer's Guide to BIND and EPFs* for details.

**CPL Commands:** PRIMOS processes only iteration for CPL commands. (Iteration is the expansion of lists contained in parentheses.) Wildcards and name generation must be processed explicitly by the CPL program itself. CPL commands are thus processed like the NX$ R-mode commands described in the next section.

**R-mode Commands:** PRIMOS processes R-mode commands in CMDNC0 in one of three ways:

- If a command name does not begin with either NX$ or NW$, full command processing is done.
- If a command name begins with NW$, iteration and treewalking patterns are processed, but wildcards and name generation patterns are not.
- If a command name begins with NX$, only iteration is processed.

You may want to create or modify commands for which you do not want to use one or more of the command line features. To prevent PRIMOS from performing the standard command line processing for a command, use the following procedure:

1. Rename the command so that it begins with NX$ or NW$.
2. Write a CPL interlude program that accepts the original invocation name and runs the renamed command.

For example, suppose you want PRIMOS to process only iteration for the command EXEC.SAVE. First, rename the command NX$EXEC.SAVE. Then add to CMDNC0 a CPL program named EXEC.CPL, which consists of the following lines:

```
&ARGS ARGS: REST
NX$EXEC.SAVE %ARGS%
&RETURN
```

Users of the old EXEC command can continue to invoke EXEC, which now invokes EXEC.CPL instead. EXEC.CPL passes the unexpanded arguments on to NX$EXEC.SAVE.

Some commands for which you might want nonstandard command line processing are the following.

- Commands added before Rev. 19 that use option names or special characters that conflict with command line features.
- Commands added before Rev. 19 for which you do not want to use one or more of the command line features.
- Commands that perform their own command line processing.
- Commands that you are adding now for which you do not want to use a command line feature. The CPL interlude is not mandatory.

## USER FILE SUFFIXES

All file suffixes beginning with the letter U are reserved for customer use. Use this suffix to create classes of user-defined files that are processed by user-written programs and commands.

All filenames must conform to Prime standards. See the *PRIMOS User's Guide* for details on filenames. Following are examples of filenames with user file suffixes:

SALES.UDATA
UPDATE.UTRANS
PROBLEMS.UXB

## ADDING LIBRARIES

To learn how to create new EPF libraries and add them to your system, read the *Programmer's Guide to BIND and EPFs*. Pay particular attention to the distinction between *program class* and *process class* EPF libraries.

## REMOVING UNUSED LIBRARIES

If you plan to remove an EPF library from your system, remember also to remove it from the system's entrypoint search list, SEARCH_RULES*>ENTRY$.SR, and to remove it from any customized entrypoint search lists under your control. Also, suggest to users that they remove it from their customized entrypoint search lists, if any.

# CHANGING COMPILER DEFAULTS

You can change the default compiler option settings for F77, Pascal, PL1, PL1G, and VRPG. For a list of the settings and a description of how to change the settings, see the current *Translator Family Software Release Document.*

# ADDING HELP FILES

At Rev. 23.0, PRIMOS HELP has been replaced with a new product which is documented in detail online. To review this documentation, type HELP HELP at PRIMOS command level. System Administrators can add HELP files (on any subject) to the HELP database supplied by Prime. After these site-created HELP files are installed, the PRIMOS HELP command can display them. This section presents a brief description of the HELP system and some instructions for adding your own HELP files.

## The HELP Database

The HELP database contains a collection of files called HELP files. HELP files are text files that contain information about a system facility, a command, or a subsystem. These files are invoked by the HELP command to provide online information about these subjects.

The name of each HELP file consists of two parts: the name of the facility, command, or subsystem, and the suffix .HELP. For example, the HELP file BIND.HELP contains information about the BIND command and subsystem.

## The HELP* Directory

HELP files are kept in the PRIMOS.TEXT subdirectory of the HELP* directory (that is, HELP*>PRIMOS.TEXT). The HELP* directory also contains the HELP_SEARCH_LIST file. The HELP_SEARCH_LIST file is a list of system-defined abbreviations for commands. This file allows a user to use a standard abbreviation as an argument for the HELP command to view the HELP file for the desired command. For example, typing either `HELP CHANGE_PASSWORD` or `HELP CPW` displays the file CHANGE_PASSWORD.HELP.

Also, there is a new feature at Rev. 23.0 called XREFS. XREFS is the cross reference capability available with PRIMOS HELP. This capability allows users to call up a list of references related to the topic at hand. Users can do this by means of the X'refs menu option which is available from any text panel in the HELP facility.

## Creating HELP Files

HELP files are standard ASCII files. To create (or modify) HELP files, use a PRIMOS text editor such as ED or EMACS.

Observe the following two rules when creating HELP files:

- The first three lines of the file are not displayed. You may leave these lines blank, or make them comment lines that indicate the date and author of the file.
- The filename must have the .HELP suffix. (That is, save the file as *command*.HELP.)

See a Prime-supplied HELP file for an example of correct format.

## Adding Files to the HELP* Directory

Use the following procedure to add HELP files to the HELP* directory:

1. Create the new file with a text editor as explained above.
2. File it in HELP*>PRIMOS.TEXT>*command*.HELP (where *command* is the name of the new command).

For detailed information about the HELP database and adding HELP files, type HELP HELP at PRIMOS command level.

## Protecting the HELP Database

When your system is first installed, HELP* is accessible to anyone. You should limit Write (W) access to this directory so that only authorized persons can alter the directory. Set the ACL for the directory to give ALL access (either by name or as a group) to users authorized to alter the database and LUR access to $REST.

**Restricted Commands:** Beginning at Rev. 23.0, more online HELP information will become available about restricted commands for which little information was given online previously. The System Administrator can set ACLS on this information by putting it in one directory and then setting appropriate ACL protection on the directory. See the *System Administrator's Guide, Volume III: System Access and Security* for more information relating to the protection of restricted HELP files.

# 7

# PLANNING THE SYSTEM CONFIGURATION

The PRIMOS operating system contains code to manage

- Access for up to 960 user processes
- Segmented virtual address space for programs up to 64 megabytes per user
- Input/output control
- The file system
- Interactive terminal users and phantom user noninteractive jobs
- Communications systems

PRIMOS is delivered in a single version that configures itself at every cold start. PRIMOS takes its configuration information from a system configuration file that defines system parameters, such as the number of users the system can support and the amount of available physical memory to be used.

Because the details of configuration vary from site to site, you must decide how you want your own system configured. This chapter discusses configuration directives and is intended to help you plan the configuration of your system.

**Note**

Before creating your system, you should establish a system log book into which you enter the parameters of your system and of your User Profile Data Base. For details on the system log book, see the *System Administrator's Guide, Volume III: System Access and Security*, the *Operator's Guide to System Monitoring*, and your CPU handbook.

## THE SYSTEM CONFIGURATION FILE

The system configuration file is usually named CONFIG, and must be in the CMDNC0 directory. It contains a series of configuration directives, one per line.

To establish or change the configuration of your system, you create or modify the system configuration file, and reboot PRIMOS. When the system comes up, the new configuration is in effect.

Read the sections that follow to learn which configuration directives you need. Then see Chapter 8, Configuration Directives, for the details of constructing the configuration file, for directions on starting up PRIMOS without a configuration file, and for a dictionary of the directives.

## TYPES OF CONFIGURATION DIRECTIVES

There are five general categories of configuration directives. The first four are discussed in this chapter. The fifth is discussed in Appendix A.

- Necessary directives, which must be set for the system to function.
- Useful directives, which need not be set, but which, when set correctly, make the system function better.
- Default-changing directives, which do not concern the system but may interest the System Administrator.
- Equipment-specific directives, which are needed if certain equipment is attached to the computer.
- Rarely used directives, which are used for system debugging or which are functionally obsolete. Avoid using these directives. For details on these directives, see Appendix A, Obsolete and Rarely Used Commands and Directives.

All numerical arguments to configuration directives must be octal numbers. Decimal equivalents are provided for ease of calculation.

## NECESSARY DIRECTIVES

Six directives must be in the configuration file:

- COMDEV specifies the command device partition.
- PAGING specifies the paging device partitions.
- SYSNAM specifies the name of the system.

- NTUSR specifies the number of directly-connected terminal users.
- NPUSR specifies the maximum number of phantoms.
- GO marks the end of the configuration file.

Four other directives may be required for networked systems:

- NRUSR specifies the number of remote users via PRIMENET.
- NSLUSR specifies the number of slave processes via PRIMENET.
- NTSUSR specifies the number of Network Terminal Service users.
- NTSASL specifies the number of Network Terminal Service assignable lines.

## Command Device

The COMDEV directive specifies which partition is logical device zero, the command device. (This partition is listed first in the output from the STATUS DISKS command.) The CMDNC0 directory on this partition is the one searched when a user invokes an external PRIMOS command.

The argument to COMDEV is the physical device number of the command device partition. See the *Operator's Guide to File System Maintenance* for details on constructing physical device numbers. In addition, at Rev. 23.0 you can specify an **entryname** (directory name containing as many as 32 characters) for the COMDEV partition using the *entryname* argument of the COMDEV directive. See the explanation of the COMDEV configuration directive in Chapter 8 of this guide.

## Paging Partitions

Each system must have at least one partition reserved for paging. (The paging partition is also referred to as the **paging device** or the **paging disk.**) The system may have up to eight such partitions.

The PAGING directive specifies the paging partitions. This directive must be included in the configuration file.

A paging partition is normally a split disk (that is, it also contains storage space for files). See Chapter 2, Creating and Allocating Disk Space, for important details on paging partitions, split disks, and methods for determining the size of paging partitions.

## System Name

The SYSNAM directive specifies the name of the system. The system uses this name to identify itself on any networks to which it may be connected. If your system is already running PRIMENET, you can give it the same name you formerly specified with the -NODE option of START_NET. Otherwise, you should choose a name appropriate for future PRIMENET or NTS operation. See the SYSNAM directive in Chapter 8 for the rules for valid system names.

If you omit the SYSNAM directive from the configuration file, PRIMOS will print an error message, and prompt you for a name. You should enter the name interactively; then remember to add the SYSNAM directive to the configuration file after PRIMOS comes up.

Likewise, if the name in the SYSNAM directive is too long, or contains invalid characters, PRIMOS will print an error message, and prompt you for a correct name.

If the SYSNAM directive is present, but supplies no name, PRIMOS will prompt you for a name, but without printing an error message. You might need this feature in either of two cases:

1. You expect to boot the system with a different name from time to time.
2. You expect to move the disk pack containing the command partition from one machine to another.

## Number of Users

The following categories of users and corresponding directives determine how many users your system can support:

- Phantom users, NPUSR
- Terminal users, NTUSR
- Network Terminal Service users, NTSUSR
- Remote users, NRUSR
- Slave users, NSLUSR

You can issue a LIST_PROCESS command to display information about each user currently on your system. Remote and slave users are for PRIMENET only. The total number of configured users of all types, NPUSR+NSLUSR+NRUSR+NTUSR+NTSUSR, must be less than or equal to 960 ($1700_8$) on the 2850™, 2950™, and the 4000 series and 6000 series systems.

On all other systems this limit is 600 ($1130_8$).

**Phantom Users:** The NPUSR directive sets the number of phantom users. Phantom users can be thought of as users at imaginary terminals because they take their commands from a file rather than from a terminal. You must set the value of NPUSR to at least 4, which is its default value. If you set NPUSR to 3 or less, PRIMOS displays the following message:

```
Warning:  Specified value of NPUSR too small.
Default value (4) will be used.
```

The processes for several services count as phantoms, for the purpose of configuration. When you calculate the value for NPUSR, you should allow phantoms for these services, as appropriate:

- In addition to some servers, which you do not need to configure, DSM uses phantoms (DSMASR) to run applications such as RESUS and SIM. You should allow at least one additional phantom for each application to be used, although the number of extra phantoms DSM actually requires will depend upon which DSM applications you use concurrently.
- Printers require one phantom per active despooler environment.
- Batch service, if used, requires one phantom for the Batch monitor, plus one or more for jobs, up to one for each queue.
- LAN300 network management requires one phantom, and the LHC downline-load and upline-dump share one more.

**Note**

It is not necessary to configure phantoms for these processes, because they are servers:

| | |
|---|---|
| Login server | DSM logger |
| Logout server | Network manager |
| Timer process | ISC network server |
| DSM system manager | NTS connection manager |
| DSM server | Security auditor |
| Name server | |

- Other communication products, including DPTX, RJE, FTS, and PRIME/SNA™, may also require phantoms.

You may want to configure some phantoms to be available for terminal users. Start with about one phantom for each five terminal users. If your terminal users complain that phantoms are not available, you can increase the number configured. If there are no complaints, you may want to decrease the number configured until there are complaints and then increase it slightly.

If you have PRIMIX on your system, see *Using PRIMIX on the Prime 50 Series* for the number of phantoms you should allot.

**Terminal Users:** The NTUSR directive sets the number of directly-connected terminal users. You can configure up to 512 ($1000_8$) terminal users. (This number includes the supervisor terminal).

The NTUSR directive, which has no default value, must be included in the configuration file. You must set the directive's value to at least the number of terminals connected to the computer, plus one for the supervisor terminal. Setting the value higher than the number of connected terminals may make it easier to add terminals in the future.

**Network Terminal Service Users:** The NTSUSR directive sets the number of simultaneous NTS terminal users. Up to 512 ($1000_8$) are allowed.

**Remote Users:** Remote users are terminal users on other systems who can log in to your system through their computer, which is networked to yours. The NRUSR directive, which has a default value of 0, sets the number of remote users. If you set the value to 0 or omit the directive from the file, no one can log in remotely to your system, regardless of any network connections. You can allow as many as 255 ($377_8$) remote users on your system.

**Slave Users:** The NSLUSR directive sets the number of slave users. Slave users are processes on your system that handle requests (made by users on other systems) for file access, attaching, and so forth.

The default value of NSLUSR is 0. If you set the value of NSLUSR to 0 or omit the directive from the configuration file, no one can access files on your system from other systems networked to yours.

You can configure as many as 255 ($377_8$) slave users.

You may wish to consult with your Prime System Analyst when setting initial values for remote and slave users. The required number depends upon your specific network and computers and upon the type of work your users are doing. Because only 255 virtual circuits are available, it usually does not make sense for NRUSR+NSLUSR to exceed 255.

### End of File

The GO directive marks the end of the configuration file. This directive must be the last noncomment line in the configuration file. Any subsequent directives will not be acted upon.

## USEFUL DIRECTIVES

Useful directives set parameters for utilization of memory, assignable asynchronous lines, and buffers. (Event logging, no longer controlled by directives, is described in the *DSM User's Guide.*)

### Assignable Asynchronous Lines

The NAMLC directive sets the maximum number of directly connected assignable asynchronous lines that can be used simultaneously. Similarly, the NTSASL directive sets the maximum number of NTS assignable asynchronous lines. A value of NAMLC or NTSASL that is too low may not become apparent immediately upon cold start. Assignable asynchronous lines are used by user programs or the spooler to communicate with serial devices such as serial printers. The default value for both NAMLC and NTSASL is 0. The number of directly connected assignable asynchronous lines (NAMLC) plus the number of directly connected terminal users (NTUSR) cannot exceed 512 ($1000_8$).

To define a line as assignable, use the SET_ASYNC command with the -ASGN YES option. To change the buffer sizes of an assigned asynchronous line use the CAB command. These two commands are described in the *System Administrator's Guide, Volume II: Communication Lines and Controllers.*

## Size of Wired Memory at Cold Start

The WIRMEM directive displays, at the supervisor terminal, the amount of wired memory (in kilobytes) at cold start. Although this value changes during system operation, it provides an indication of the wired memory required to support a particular system configuration.

## Number of Locate Buffers

PRIMOS incorporates a memory-to-disk cache that stores the most recently and most frequently accessed disk records, thus reducing disk I/O. This cache is made up of a number of main memory buffers called **locate buffers** (also called **associative buffers**). Each locate buffer is two kilobytes in size. The default number of locate buffers is 64 ($100_8$).

By using the NLBUF directive, you can configure from 8 ($10_8$) to 5000 ($11610_8$) locate buffers. Configuring more locate buffers can decrease disk I/O. However, additional locate buffers can use up more memory, and if not enough memory is available, paging I/O may increase to the point where it cancels the advantage gained by increasing the number of locate buffers. For example, 256 buffers require one-half megabyte of memory. The maximum value, 5000, uses about ten megabytes, which may be more than the total amount of memory on some systems.

The optimal number of locate buffers depends upon the applications running on the system. These buffers are most useful when applications access the same file records repeatedly. It may be appropriate to configure more buffers if the USAGE command reports a high locate miss rate (the %Miss field) and a low or normal page fault rate (PF/S). An increase in NLBUF should result in a decrease in %Miss without significantly changing the page fault rate. Also, lowering NLBUF can help where the page fault rate is high, but %Miss is normal or low.

At Rev. 22.1 the value of NLBUF is checked more extensively at cold start. If the value of NLBUF is invalid at cold start, one of the messages described below is displayed on the supervisor terminal.

If the value supplied for NLBUF is below the minimum allowed (8 buffers), this message is displayed at the supervisor terminal:

```
NLBUF LESS THAN MINIMUM
```

After the message appears, NLBUF is set to 64.

If the value supplied for NLBUF uses more memory than is physically available, this message appears on the supervisor terminal:

```
NLBUF VALUE TOO LARGE FOR AVAILABLE MEMORY
```

After the message appears, NLBUF is set to a value that uses one half (1/2) the size of the available memory in physical pages, or to the maximum value allowed for NLBUF, whichever is smaller.

If the value supplied for NLBUF exceeds the maximum value allowed, this message appears on the supervisor terminal:

```
NLBUF VALUE EXCEEDS MAXIMUM ALLOWED
```

After the message appears, NLBUF is set to a value that uses one half (1/2) the size of the available memory in physical pages, or the maximum value allowed for NLBUF, whichever is smaller.

A new background process, BUFFER_SERVER, monitors the number and availability of Locate buffers, and flushes them for your system. You can display the process by using the STAT USER and LIST_MEMORY commands.

Priority and timeslices for BUFFER_SERVER cannot be changed.

If you see this message displayed on your terminal, no action is required. The system will attempt to restart the process.

```
BUFFER_SERVER aborted by raising of xxx condition at yyy.
(flush$proc)
Attempting to restart BUFFER_SERVER.
```

If BUFFER_SERVER fails to spawn, you will see this message:

```
CANNOT SPAWN BUFFER SERVER PROCESS
(startfl)
Activated User 1 buffer flush
System performance may be degraded; cold start required
to restart buffer server.
```

The message is logged with DSM, and a flag is set that tells User 1 to flush the buffers. You must cold start the system to stop system degradation.

If BUFFER_SERVER fails to restart after aborting, you will see this message:

```
BUFFER_SERVER aborted by raising of xxx condition at yyy.
(flush$proc)
Activated User 1 buffer flush
System performance may be degraded; coldstart required
to restart buffer server.
```

The message is logged with DSM, and a flag is set that tells User 1 to flush the buffers. You must cold start the system to stop system degradation.

### VMFA Dynamic Segments

The NVMFS directive sets the number of VMFA (Virtual Memory File Access) dynamic segments available in virtual address space for the system. VMFA segments are used by EPFs to map segments dynamically.

The default number of VMFA segments is 100 ($144_8$). You may want to increase this number if users frequently receive messages such as `Not enough segments` or `No space available from process class storage heap`, or if you expect them to use a large variety of EPF programs or libraries. You can specify a maximum of 1024 ($2000_8$) VMFA segments.

If you have PRIMIX on your system, see *Using PRIMIX on the Prime 50 Series* for the number of VMFA segments you should specify.

## DEFAULT-CHANGING DIRECTIVES

Default-changing directives change the default values of the directives that control the following: printing the directives as they are being processed, user-defined abbreviations, erase and kill characters, ECCU handling, and certain login and logout procedures.

### Displaying Configuration Directives

By default, configuration directives are not displayed at the supervisor terminal as they are processed. To display these directives at the supervisor terminal, include the TYPOUT YES directive in the file.

All directives after the TYPOUT YES directive are displayed until either the TYPOUT NO directive or the GO directive is encountered in the configuration file.

### User-defined Abbreviations

By default, users can use the ABBREV command to create abbreviations for PRIMOS commands and their arguments. The abbreviations are stored in abbreviation files. When used on the command line, the abbreviation is expanded by the system's abbreviation processor.

If you do not want users to create and use abbreviations, disable the abbreviation processor by including the ABBREV NO directive in the configuration file.

If you omit the ABBREV directive from the configuration file (or specify ABBREV YES), the abbreviation processor is enabled and users can employ command line abbreviations.

## Erase and Kill Characters

Erase and kill characters are used on the PRIMOS command line and within programs. The erase character erases the character to the immediate left of the cursor. For example, typing the word DATE and then next typing the erase character is the same as if you had typed only DAT.

The kill character nullifies all characters to the left of the cursor. For example, typing the word DATE at the OK, prompt and then next typing the kill character is the same as if you had typed nothing after the prompt.

The default systemwide erase character is the double-quote character (") and the default kill character is the question mark (?). To change these default characters, use the ERASE and KILL directives. If you change either or both of these characters, inform all of your users because Prime's documentation assumes that ERASE and KILL have the original default values. Remember that the character values for erase and kill must be different from one another.

**Note**

Whether or not the System Administrator changes the default erase and kill characters, users can change these characters for their terminal sessions by using the -ERASE and -KILL options of the PRIMOS TERM command. Details of TERM are given in the *PRIMOS User's Guide.*

## ECCU Handling

An ECCU (Error Correction Code Uncorrectable) is a two-bit memory parity error. The MEMHLT directive determines how PRIMOS handles the occurrence of an ECCU.

If the MEMHLT directive is not in the configuration file or if the default MEMHLT YES is included, the system halts when an ECCU occurs.

If MEMHLT NO is in the configuration file and certain conditions are met, PRIMOS can detect what user process encountered the ECCU. PRIMOS then logs out that user process, prints a message at the supervisor terminal listing the user ID of the process, and continues operating normally for other users. See the MEMHLT directive in Chapter 8, Configuration Directives, for the conditions that must be met for the user process to be logged out.

MEMHLT NO is appropriate only if your system is serviced regularly and if you are not running ROAM-based data management products (DBMS, DISCOVER™, and PRISAM™). If you use MEMHLT NO and your system still halts with memory parity errors, have your system serviced. Otherwise the system may experience an undetectable or falsely corrected error because it is running with faulty memory.

For a discussion of whether to use a warm start or a cold start after a system halt or hang, see the chapter on Equipment and Environment in the *System Administrator's Guide, Volume III: System Access and Security* and your CPU handbook.

**Caution**

Systems running ROAM-based data management products (DBMS, DISCOVER, PRISAM) should have MEMHLT YES in the configuration file and should be cold started after any system halt. A cold start is necessary so that rollback of incomplete transactions can occur. A warm start may cause loss of data.

## Changing the Login/Logout Procedure

Six directives modify the default login and logout procedures. These directives control the following:

- The printing of login/logout messages at the supervisor terminal (LOGMSG)
- The printing of unsuccessful login messages at the supervisor terminal (LOGBAD)
- The use of the LOGIN command for logged-in users (LOGLOG)
- The automatic logging out of disconnected users (DISLOG)
- The length of the inactivity timeout (LOUTQM)
- The time allowed for a login procedure (LOTLIM)

**Printing Login/Logout Messages:** When a user logs in or out, a message to this effect is printed by default at the supervisor terminal. These messages provide the System Administrator with a record of these transactions. (Activity of child processes is not recorded.)

If you decide that such detailed information is not necessary, you can disable these messages by using the LOGMSG NO directive in the configuration file. (Disabling such messages saves paper on hard-copy supervisor terminals.) Omitting the LOGMSG directive (or specifying LOGMSG YES) causes these messages to be displayed at the supervisor terminal.

**Printing Unsuccessful Login Messages:** The LOGBAD directive controls the printing, at the supervisor terminal, of messages about unsuccessful login attempts. If you omit the directive from the configuration file or specify LOGBAD NO, such messages are not printed.

If LOGBAD is enabled (by your having specified LOGBAD YES in the configuration file), any unsuccessful attempt to log in (due to an invalid user ID, incorrect password, or invalid project ID) causes a message to be printed at the supervisor terminal.

**Disabling the LOGIN Command:** By default, a user can issue the LOGIN command while logged in. (A logged-in user might wish to log in under a different user ID or under a different project.) The user is first logged out and then logged in again according to the arguments of the LOGIN command. External logout and login programs in CMDNC0 are run if they exist.

To allow use of the LOGIN command to logged-in users, either specify LOGLOG YES in the configuration file or omit the directive.

Specifying LOGLOG NO prevents the use of the LOGIN command for logged-in users and forces them to log out explicitly (with the LOGOUT command) before being able to log in again. Forcing users to log out explicitly prevents a user from unknowingly logging out another user who has left a terminal and not logged out.

**Logging Out Disconnected Users:** The DISLOG directive, described in Chapter 8, provides one means for selecting, for each line, whether disconnection of the line causes automatic logout. The preferred method is to use the SET_ASYNC command with the -DISLOG option, described in the *System Administrator's Guide, Volume II: Communication Lines and Controllers*. The SET_ASYNC command allows you to change the dislog status of lines dynamically.

Use the directives DTRDRP and DISLOG YES or DISLOG *line-number* if you have lines configured for Auto Speed Detect (ASD) so that each line is returned to ASD when the user disconnects.

**Inactivity Timeout:** You can set the amount of time a terminal can remain idle before its user is automatically logged out (inactivity timeout). Prime supplies a default time of 1000 ($1750_8$) minutes, which is 16 hours and 40 minutes. To retain this default value for the inactivity timeout, omit the LOUTQM directive from the configuration file.

To change the value of the inactivity timeout, use the LOUTQM directive. For example, LOUTQM 74 sets the inactivity timeout to 1 hour (1 hour equals 60 minutes, whose octal value is $74_8$).

**Length of Login Procedure:** You can use the LOTLIM directive to set the length of time allowed for a user to log in. The default value of three minutes is the recommended length because it gives a user a reasonable amount of time to type in required information without wasting system resources.

To change the time allowed for login, use the LOTLIM directive. The minimum amount of time you can allow is one minute. There is no maximum. The time should always be less than the time allowed by the LOUTQM directive.

## Amount of Memory to Use

Omit the MAXPAG directive from the configuration file. MAXPAG limits the amount of memory available to PRIMOS and is not normally useful. One reason for using MAXPAG is to test system performance with reduced memory. Another is to allow PRIMOS to run on a system that has some defective memory in higher addresses. See Appendix A for full information.

# EQUIPMENT-SPECIFIC DIRECTIVES

Several directives change parameters that control such equipment as buffers, the AMLC programmable clock, the supervisor terminal, and several types of lines.

## Changing Buffer Sizes

Although many devices operate with the default buffer sizes, it is often desirable (and, in some cases, necessary) to change these sizes. At Rev. 22.0 the CAB command, described in the *System Administrator's Guide, Volume II: Communication Lines and Controllers*, replaces the REMBUF, AMLBUF, NTSBUF, and NTSABF configuration directives.

**DMC Tumble Tables:** The AMLIBL directive sets the size of each input buffer for the AMLC controller DMC (Direct Memory Control) tumble tables. If the directive is not included in the configuration file, the default buffer size is 48 ($60_8$) halfwords.

An AMLC line attached to a high-speed input device could send data into the tumble tables faster than it could exit, resulting in the loss of the data. In this case, you can increase the buffer size with the AMLIBL directive or you could let the system calculate a value for you. The maximum size for the buffers depends upon the number of controllers and the amount of space available in the system for buffers. To let the system calculate and set this value, specify either AMLIBL 0 or AMLIBL with no argument.

For more information on the DMC tumble tables, see the *System Administrator's Guide, Volume II: Communication Lines and Controllers.*

**ICS Controllers:** The ICS INPQSZ directive changes the size of the input queues for ICS controllers from the default value of 63 ($77_8$). You may need to change the queue size on systems that have many terminals sending large amounts of data. For further details on configuring ICS lines, see the *System Administrator's Guide, Volume II: Communication Lines and Controllers.*

## AMLC Programmable Clock

The AMLC hardware contains a software programmable clock. The clock's default baud rate is 9600 ($22600_8$). To change the default, use the AMLCLK directive with a value in the range from 29 ($35_8$) to 19200 ($45400_8$) baud. You should keep the default baud rate if you are using Auto Speed Detect (ASD) on any of your lines.

To specify that an asynchronous line *n* (a decimal number) is to use the programmable clock speed, use the command SET_ASYNC -LINE *n* -DEFAULT -SPEED CLOCK. (SET_ASYNC is a command, *not* a configuration directive.) If you use the -SPEED CLOCK option, but the AMLCLK directive is not in the configuration file, the line assumes the default speed of 9600 ($22600_8$) baud. See the *System Administrator's Guide, Volume II: Communication Lines and Controllers*, for details on the SET_ASYNC command.

## Telephone Lines

The AMLTIM configuration directive sets the values for the three timers associated with dialup lines. You can use the defaults for the first two arguments, *ticks* and *disctime.* However, you may want to set the third timer, *gracetime.* The third timer is essentially the amount of time the system allows a line to remain active without a process being logged in.

A reasonable value for *gracetime* is from 1 to 3 minutes. The value for one minute (which is 600 tenths of a second) is $1130_8$ and is specified with the AMLTIM directive as follows:

```
AMLTIM 2 3410 1130
```

(The first two arguments, 2 and 3410, are the defaults for the first two timers.) To set *gracetime* to two minutes, use the value $2260_8$. To set *gracetime* to three minutes, use the value $3410_8$.

When the user logs out, the line remains active for the period specified by the *gracetime* argument. If the user was logged in on a dialup line and hangs up the telephone without logging out, whether the DTR (Data Terminal Ready) signal is dropped depends on the presence or absence of the DTRDRP directive in the system configuration file.

- If DTRDRP is not in the configuration file, the DTR signal is dropped within the time period specified by the first argument (*ticks*) to AMLTIM, causing the line to become inactive.
- If DTRDRP is in the configuration file, the DTR signal is dropped immediately, regardless of the value of *gracetime.* This prevents a user from dialing in and being connected (with full access rights) to the process of another user who has disconnected without logging out.

## Supervisor Terminal

Two directives, ASRATE and ASRBUF, change the default baud rate and buffer size of the supervisor terminal. You may have to use these directives if you have a nonstandard supervisor terminal. For details on using them, see Chapter 8.

**Changing the Baud Rate:** The ASRATE directive sets the baud rate of the supervisor terminal. Many hard-copy supervisor terminals have a baud rate of 300, which is the default of ASRATE, although some can use 1200. The rate of 110 baud is available for slower terminals.

If you have a screen terminal (such as a PT200™ or a PT250™) as your supervisor terminal, you may want to use one of the other available baud rates of 1200 or 9600.

If used, the ASRATE directive should be the first directive in the configuration file.

**Changing Buffer Sizes:** Most hard-copy supervisor terminals can use the default input buffer of 256 ($400_8$) bytes and the default output buffer of 384 ($600_8$) bytes. If your supervisor terminal runs at greater than 300 baud, or if you are using LAB or DSM SIM commands, you might need to use the ASRBUF directive to increase the output buffer size.

If you are using the RESUS remote supervisor terminal facility, your main and remote supervisor terminals may be operating at different baud rates. Because the data stream is sent to *both* terminals, the faster terminal may occasionally be held to the speed of the slower unless you increase the buffer size of the *slower* terminal.

## Synchronous Lines

Synchronous lines to other computers and devices are enabled and configured with the SYNC directives. Which directive is used and what values are assigned depend upon the specific hardware and controllers on your system. The SYNC ON directive is used for all synchronous line types, including MDLC and ICS lines.

For details on the four SYNC directives, see Chapter 8, Configuration Directives.

**Note**

At Rev. 20, SYNC became a synonym for SMLC. For example, specifying SYNC ON is the same as specifying SMLC ON. SYNC is now the preferred term.

## Uninterruptible Power Supply

An Uninterruptible Power Supply (UPS) maintains power to the CPU and memory during a power failure. On many systems it also automatically performs a warm start. If your system has UPS, the UPS directive determines what action is taken after a warm start.

The UPS directive with an argument of 0 produces a warm start followed by a halt. The operator must then intervene to bring up the system.

The UPS directive with a positive argument tells UPS to perform a warm start and then wait for a number of seconds (as specified by the argument) before bringing up the system. The delay allows the disks to reach full speed before PRIMOS attempts to access them. For example, UPS 100 tells UPS to wait 64 seconds after a warm start before it brings up PRIMOS. A value of $100_8$ is recommended for a storage module. If your system does not have an Uninterruptible Power Supply, omit the UPS directive from the configuration file.

## ICS Controllers

If your system has an ICS controller, you may use these two directives:

- ICS INTRPT, which sets the asynchronous interrupt rate
- ICS CARDS, which verifies an asynchronous Line Adapter Card configuration of an ICS2 or ICS3 controller

- ICS INPQSZ, which sets the size of the input queue buffers for ICS lines.

In addition, the ICS INPQSZ directive sets the size of the ICS input queue buffers. This directive is described in the *System Administrator's Guide, Volume II: Communication Lines and Controllers.*

**Modifying the Interrupt Rate:** To set a faster interrupt rate than the default 100 millisecond rate on lines connected to ICS controllers, use the ICS INTRPT directive. You can set the rate to a value between 100 ms and 10 ms. (To set the interrupt rate on lines connected to AMLC controllers, use the AMLC command, as explained in the *System Administrator's Guide, Volume II: Communication Lines and Controllers.*) It is usually better to change the buffer size for line for a particular device than to change the interrupt rate.

**Verification of ICS Configuration:** ICS2 and ICS3 controllers, which can have up to 16 Line Adapter Cards (LACs), are configured at each cold start. To check that the LAC configuration is as you expected, include the ICS CARDS directive in the configuration file.

If the actual LAC configuration at cold start is different from that specified by this directive, PRIMOS displays an error message explaining the discrepancy. Such a discrepancy can occur if an extra LAC was added since you configured your system, or if a LAC went bad since the system was last cold started.

If the ICS CARDS directive is omitted for an ICS2 or ICS3 controller, that controller's configuration is not checked at cold start.

For further details on the ICS CARDS directive, see Chapter 8, Configuration Directives, or the *ICS User's Guide.*

# 8

# CONFIGURATION DIRECTIVES

The PRIMOS operating system configures itself at every cold start. The configuration information that PRIMOS needs is stored in the system configuration file. The first half of this chapter discusses this file and the PRIMOS CONFIG command that processes it. The second half presents all configuration directives alphabetically, explaining each directive and its arguments.

## SYSTEM STARTUP AND CONFIGURATION

### The System Startup File

At cold start, PRIMOS runs the **system startup file**, PRIMOS.COMI, in the CMDNC0 directory. The first command in that file must be the CONFIG command, which specifies the system configuration file and begins its processing.

PRIMOS.COMI also must contain several other commands for initializing communication lines and various software packages. See the *System Administrator's Guide, Volume II: Communication Lines and Controllers* for details.

**Notes**

The name C_PRMO for the system startup file is obsolete, but still supported. If your system uses C_PRMO, you should change it to PRIMOS.COMI.

If CMDNC0 contains both PRIMOS.COMI and C_PRMO, PRIMOS uses PRIMOS.COMI.

The *Rev. 23.0 Software Installation Guide* contains a sample of a system startup file. A template of the PRIMOS.COMI system startup file is shipped with each Prime system as PRIRUN>PRIMOS.COMI.TEMPLATE. Copy it to CMDNC0 as PRIMOS.COMI, and use a text editor to change the template to suit your installation.

### The CONFIG Command

The CONFIG command functions only during system startup. The format of the CONFIG command is:

**CONFIG -DATA *config-filename***

*config-filename* is the name of the configuration file (usually CONFIG) in the CMDNC0 directory. In the following example, the current directory is CMDNC0 and the name of the configuration file is CONFIG:

```
CONFIG -DATA CONFIG
```

**Note**

If you reexecute the PRIMOS.COMI startup file while PRIMOS is running (by issuing a **COMINPUT PRIMOS.COMI** command at the supervisor terminal), the file CMDNC0>CONFIG.CPL executes, preventing the generation of an error when the system encounters the CONFIG command in the startup file. Instead, CONFIG.CPL displays the message

```
Primos already running, CONFIG command is ignored.  (CONFIG.CPL)
```

and PRIMOS continues executing the PRIMOS.COMI file. (You may occasionally need to reexecute the startup file to re-share and reinitialize system software.)

## THE SYSTEM CONFIGURATION FILE

Because PRIMOS is configured each time the system is cold started, the System Administrator can reconfigure a system as necessary to meet changing needs. Most systems, however, use a constant set of configuration directives. The directives that configure the system are stored in a text file called the **system configuration file.** The system configuration file is usually named CONFIG and must be stored in the CMDNC0 directory.

### Creating the Configuration File

Because configuration files vary greatly from system to system, a template configuration file is not shipped with the system software. Therefore, it is your responsibility to create the configuration file.

Use the following rules and guidelines when creating the configuration file:

- For a new system, boot without a configuration file (see the section Booting PRIMOS Without a Configuration File) and, once the system has booted, use the NSED text editor, under PRIMOS, to create an initial configuration file.
- Enter each configuration directive on a separate line. The directives may be entered in uppercase or lowercase.
- You may insert comment lines after a directive or on a separate line. Comments must begin with the /* character pair.

- The ASRATE command should normally be the first directive in the file
- The COMDEV, PAGING, NTUSR, NPUSR, SYSNAM, and GO directives must be in the configuration file. GO must be the last directive in the file.
- All numerical values in the configuration file must be in octal. For details on constructing physical device numbers needed as arguments for certain directives, see the *Operator's Guide to File System Maintenance.*

To change the system configuration for the next cold start, modify the configuration file with a text editor. The next time the system is brought up, the new configuration takes effect. If you cannot bring up PRIMOS because of errors in the configuration file, use the procedure described in the section Booting PRIMOS Without a Configuration File later in this chapter.

## Network Information

If your system will be part of a PRIMENET or NTS network, you must set certain configuration directives, such as NSLUSR, NRUSR, or NTSUSR. Other information dealing with the computer's interface to the network is stored in a global network configuration file.

For PRIMENET, the System Administrator creates the global network configuration file with the CONFIG_NET utility. The file is usually stored in the PRIMENET* directory.

For additional PRIMENET information, see the *User's Guide to Prime Network Services* and the *Operator's Guide to Prime Networks.* For information on configuring and administering the PRIMENET system, see the *PRIMENET Planning and Configuration Guide.*

For NTS, the System Administrator creates the global network configuration file with the CONFIG_NTS utility. The file is usually stored in the NTS* directory.

For additional NTS information, see the *NTS User's Guide.* For information on configuring and administering NTS, see the *NTS Planning and Configuration Guide.*

## Examining the System Configuration

To examine the configuration of a running PRIMOS system, use the LIST_CONFIG command. For more information about LIST_CONFIG, see the *DSM User's Guide.*

## Configuration File Errors

Certain errors in the configuration file prevent a successful cold start of the system. When an unsuccessful cold start occurs, a message is displayed at the supervisor terminal that explains which configuration directive was at fault and requests a system restart. You can then boot PRIMOS without a configuration file (as described below), correct the file, and reboot PRIMOS.

The most common causes of errors are:

- Placement of configuration directives into the system startup file (PRIMOS.COMI).
- Placement of system startup commands into the configuration file.
- Decimal numbers used by mistake instead of octal numbers.
- Out-of-range values for parameters (for example, a value of 12 in the SYNC directive, which only takes values from 00 to 07).
- Values that are correct in themselves, but conflict with other values. For example, the value for NPUSR+NSLUSR+NRUSR+NTUSR+NTSUSR on the 2850, 2950, and 4000 series and 6000 series processors must be no greater than $1700_8$ (decimal 960). Values of $760_8$ for NPUSR and NTUSR, plus $20_8$ for NRUSR and NSLUSR, each correct in isolation, produce a sum ($760_8+760_8+20_8+20_8=2000_8$) greater than $1700_8$ and prevent the system from starting up.

For a listing of PRIMOS cold start messages, including those produced by erroneous directives, see your CPU handbook.

## Booting PRIMOS Without a Configuration File

To boot PRIMOS without a configuration file, use the manual boot procedure. See the CPU handbook for your computer model for detailed information on booting. Brief instructions are given below.

These three situations may require manual booting:

- The system is new and has no configuration file.
- The configuration file or PRIMOS.COMI has been damaged or made incorrect.
- The hardware configuration has been changed, and the configuration file now is inappropriate for booting. (For example, COMDEV is now on a different disk drive or controller.)

---

**Caution**

Avoid using the obsolete single-user system PRIMOS II. In particular, do not use NSED under PRIMOS II to edit the configuration file on a Rev. 20 or later disk. PRIMOS II cannot write to Rev. 20 or later disks. Instead, use NSED under manually-booted PRIMOS, as explained below.

---

To change the configuration file using the manual boot, perform the following five steps:

1. At the CP> prompt, use the SYSCLR command to master-clear the system.
2. At the next CP> prompt, enter the BOOT command with the 100000 boot option added to the option word (for example, BOOT 14134 becomes BOOT 114134). Respond to the prompts:

   | | |
   |---|---|
   | `Enter COMmand DEVice:` | The physical device number (pdev) for the command partition. This is the argument to the COMDEV directive. |
   | `Enter PAGING device:` | The physical device number (pdev) for a paging partition. This must be a single value and *not* the multiple values permitted by the PAGING directive. |
   | `Enter Number Terminal USeRs:` | Any appropriate small number. Because the system is being brought up only to change the configuration file, there will be no users other than the supervisor terminal. This number is the argument to the NTUSR directive. |

   Coldstart now begins, and a few messages appear, followed by another prompt.

3. Respond to the prompt:

   | | |
   |---|---|
   | `Enter system name:` | The six-character name for this system. This is the argument to the SYSNAM directive. |

   A few more messages now appear as PRIMOS comes up. The PRIMOS.COMI file is *not* executed, with the result that no user software is shared, and no users other than the supervisor terminal are permitted.

4. Use NSED (the nonshared editor) to edit the configuration file.
5. Try out the new or revised configuration by shutting down PRIMOS, and rebooting it without the 100000 boot option.

**Notes**

Adding the 100000 value to the BOOT option word (at Step 2) instructs PRIMOS to ignore the PRIMOS.COMI system startup file. As a consequence, PRIMOS does not read the configuration file. Instead, PRIMOS queries the operator for the COMDEV, PAGING, NTUSR, and SYSNAM parameters, and uses default values for the other configuration parameters.

PRIMOS also does not downline-load the communication controllers or any intelligent disk controllers.

Because DSM has not started, no system logging takes place.

# DESCRIPTION OF CONFIGURATION DIRECTIVES

This section describes the directives used in configuration files. See also Chapter 7, Planning the System Configuration, for more information and guidelines on these directives.

### ▶ ABBREV [YES / NO]

Controls user abbreviations.

YES enables the abbreviation processor, allowing use of the ABBREV command. YES is the default.

NO prohibits the use of the ABBREV command.

### ▶ AMLBUF *line* [*in-buff-size* [*out-buff-size* [*dmq-size*]]]

Sets the size of an asynchronous line's I/O buffers.

The AMLBUF directive, obsolete at Rev. 22.0, is described in Appendix A. To set buffer sizes, use the CAB command, described in the *System Administrator's Guide, Volume II: Communication Lines and Controllers.*

### ▶ AMLCLK *baudrate*

Sets the software programmable clock in the AMLC hardware to a baud rate of *baudrate* bits per second.

| *Parameter* | *Minimum* | *Maximum* | *Default* | *Recommended* |
|---|---|---|---|---|
| *baudrate*, octal (decimal) | 35 (29) | 45400 (19200) | 22600 (9600) | as required |

The value specified for *baudrate* must be no less than $35_8$ (29 decimal) and no greater than $45400_8$ (19200 decimal). The default is $22600_8$ (9600 decimal).

The default speed is recommended if you are using Auto Speed Detect (ASD) on any line.

When used on a system where ICS or LHC300 asynchronous lines are present, *baudrate* must be one of the valid baud rates listed in the table under the ASYNC JUMPER directive.

### ▶ AMLIBL *buffer-size*

Sets the size of the DMC AMLC input tumble tables at cold start.

*buffer-size* is the number of halfwords allocated to each input buffer. Except for the special value of 0 described below, *buffer-size* must be greater than $20_8$. The maximum value of *buffer-size* is variable and depends on the number of AMLC controllers configured and the amount of space available in the system for buffers.

If *buffer-size* is 0 or is omitted, the size of the buffers is automatically calculated as the maximum size allowed by the available buffer space. If the AMLIBL directive is omitted from the configuration file, the default buffer size is $60_8$ (decimal 48).

Each AMLC controller has one pair of buffers and all buffers are configured to the same size. Data is stored one character per halfword.

During cold start initialization, the error message BAD AMLIBL PARAMETER (CINIT) is displayed if *buffer-size* is too small, and INPUT BUFFERS TOO LARGE (AMINIT) is displayed if *buffer-size* is too large. Modify the parameter to be a value within the permissible range as described above.

See the *System Administrator's Guide, Volume II: Communication Lines and Controllers*, for more information on configuring asynchronous lines.

### ▶ AMLTIM [*ticks* [*disctime* [*gracetime*]]]

Sets time intervals for the three variable event timers.

| *Parameter* | *Minimum* | *Maximum* | *Default* | *Recommended* |
|---|---|---|---|---|
| *ticks*, octal | 1 | - | 2 | 24 or more |
| *disctime*, octal | 0 | - | 3410 | 400 or more |
| *gracetime*, octal | 0 | - | 0 | 1130 |

The arguments have the following values and meanings:

*ticks* The interval (in tenths of a second) between carrier check operations. At the end of each tick period, PRIMOS checks each line for carrier loss. If a loss has occurred, the DTR signal is dropped for a period equal to the value of ticks. In addition, if the DISLOG directive is set and a process is connected to the the line, the process is logged out. The value of ticks must be greater than 0. The default is 2 (0.2 seconds).

*disctime* The maximum time period (in tenths of a second) allowed before DTR is dropped on lines that have no carrier. (The actual time period in which this event occurs varies from 0 to disctime.) When disctime expires, the DTR signal is dropped for a period equal to the value of ticks. Specifying a value of 0 disables this feature. Otherwise, the value must not be less than the value of ticks and is truncated to the nearest multiple of the tick value. The default is 3410 (1800 in decimal, which is 3 minutes).

*gracetime* The minimum time period (in tenths of a second) allowed before DTR is dropped for lines that have active carriers but are not connected to logged-in processes. (The actual grace period varies from gracetime to twice gracetime.) Gracetime in effect defines the amount of time for a caller to establish itself with a logged-in process. When the grace period expires, the DTR signal is dropped for a period equal to the value of ticks. The default value of 0 disables this feature. The specified value (if not 0) must be greater than ticks and is truncated to the nearest multiple of ticks. The value of gracetime should be large enough to enable PRIMOS to generate a forced logout of a previous user and enable another user to complete a login attempt.

**Notes**

The AMLTIM directive affects the operation of Auto Speed Detect (ASD). No standard settings for the AMLTIM parameters can be recommended if your installation has ASD, but the following values for *ticks*, *disctime*, and *gracetime* have been shown to give satisfactory results. Set the value for *ticks* to at least $24_8$. Set the value for *disctime* to at least twice that of *ticks*, preferably larger than $400_8$. Set a value of $1000_8$ for *gracetime*. ASD uses up a portion of *gracetime* before a user logs in.

For NTS terminal lines there is no carrier signal. The value specified for *disctime* is ignored, and *gracetime* is the time allowed for an LTS user to log in, once connected.

For those lines using DATA SET SENSE (carrier is used for flow control), ticks, gracetime, and disctime processing do not apply.

### ▶ ASRATE *rate*

Sets the baud rate of the supervisor terminal.

The only valid values for *rate* are the following four octal integers:

| *Octal code* | *Baud rate (decimal)* |
|---|---|
| 0110 | 110 |
| 1010 | 300 (default) |
| 2010 | 1200 |
| 3410 | 9600 |

**Note**

The values for *rate* are coded bit patterns, and are **not** the octal equivalents of decimal values.

If used, ASRATE should be the first directive in the configuration file because it ensures that any subsequent configuration error messages are displayed at the desired speed.

### ▶ ASRBUF 0 [*in-buff-size* [*out-buff-size*]]

Sets the sizes of the supervisor terminal I/O buffers.

| *Parameter* | *Minimum* | *Maximum* | *Default* | *Recommended* |
|---|---|---|---|---|
| *in-buff-size*, octal | 0 | 7777 | 200 | 200 |
| *out-buff-size*, octal | 0 or 100 | 7777 | 300 | 300, but see below. |

The arguments have the following values and meanings:

*in-buff-size* — The size in halfwords (two characters per halfword) of the supervisor terminal input buffer. The default is $200_8$ (128 decimal). If 0 is specified, the buffer size remains at its previously set value (which is usually the default size).

*out-buff-size* — The size in halfwords (two characters per halfword) of the supervisor terminal output buffer. The default is $300_8$ (192 decimal). The minimum value (other than 0) is $100_8$ (64 decimal). If 0 is specified, the buffer size remains at its previously set value (which is usually the default size).

If you are using screen-formatted commands such as LAB, DSM SIM, or USAGE at the supervisor terminal, you should increase *out-buff-size* to at least $2000_8$ (1024 decimal).

You might also want to enlarge *out-buff-size* if you have the following situation: a 300 or 1200 baud supervisor terminal, LOGMSG enabled (which prints login and logout messages at the supervisor terminal), and many logins and logouts. Such a situation can result in noticeable delays if frequent messages are sent to a supervisor terminal with a standard *out-buff-size* and a relatively slow baud rate.

### ▶ ASYNC JUMPER *speeda speedb speedc*

Defines the available line speeds for asynchronous lines.

*speeda*, *speedb*, and *speedc* are line speeds (specified as bits per second) in octal. These three speeds can be chosen from the list below. The speeds you can use on lines configured for Auto Speed Detect (ASD) are marked with an asterisk. ICS lines and NTS assignable lines support all of the speeds listed below.

| *Speed (bps)* | *Octal Value* | *Speed (bps)* | *Octal Value* |
|---|---|---|---|
| 50 | 62 | 1800 | 3410 |
| 75 | 113 | 2400 | *4540 |
| 110 | *156 | 3600 | 7020 |
| 150 | 226 | 4800 | *11300 |
| 200 | 310 | 7200 | 16040 |
| 300 | *454 | 9600 | *22600 |
| 600 | *1130 | 19200 | *45400 |
| 1200 | *2260 | | |

The defaults are 75 for *speeda*, 150 for *speedb*, and 1800 for *speedc*.

Use the following guidelines in determining whether to specify the ASYNC JUMPER directive:

- If you have only unmodified AMLC lines and are not using ASD, do not use the ASYNC JUMPER directive because it does not affect the lines.
- If you have an AMLC board on which hardware jumpers provide non-default line speeds on that AMLC, you must use ASYNC JUMPER for those speeds, regardless of whether you are using ASD.
- If you have AMLC lines and are using ASD, use the ASYNC JUMPER directive and specify speeds that match the speeds of the on-board hardware jumpers. Line speed is determined by both the on-board hardware jumpers and by this directive.
- If you have both AMLC lines and ICS lines, use the ASYNC JUMPER directive to set both types of lines to the same value. What speed you choose is up to you, but the speed for both types of lines must be the same, and must match the hardware jumper speeds on the AMLC boards.

### ▶ COMDEV *pdev* [*entryname*]

Specifies *pdev* as the physical device number of the command device, the partition on which the system command directory, CMDNC0, resides. See the *Operator's Guide to File System Maintenance* for details on the construction of pdevs.

The partition specified by this directive becomes logical disk zero. The COMDEV directive must be specified in the configuration file.

At Rev. 23.0, you can also specify an optional **entryname** (directory name containing as many as 32 characters) for the COMDEV partition. If you wish to have the COMDEV mounted in the root directory with a directory name, use the *entryname* argument.

**Note**

Once you have specified a directory name for the COMDEV partition, the partition must always be accessed by users and in pathnames by the directory name and not by the six-character diskname. Also, if there are any pre-Rev. 23.0 systems in your file system name space, do not add the COMDEV or any other partition with a name longer than six characters. If you do, the pre-Rev. 23.0 systems cannot access that partition.

### ▶ COMDVM *pdev*

Specifies *pdev* as the physical device number of the partition that mirrors the command device (COMDEV) and enables mirroring of file system partitions.

This directive turns on command device mirroring at system startup. If the COMDVM directive is used, it must follow the COMDEV directive in the configuration file. See the *Operator's Guide to File System Maintenance* for more information on mirroring.

### ▶ DISLOG [YES | NO | *line-number*]

Enables automatic logout when a line is disconnected. A line is defined as disconnected when the carrier detect signal goes logically low.

**Note**

To set DISLOG dynamically on individual lines, you can use the -DISLOG and -NO_DISLOG options of the SET_ASYNC command, which is described in the *System Administrator's Guide Volume II: Communication Lines and Controllers.* This method is preferable to using the DISLOG directive.

DISLOG YES enables automatic logout-on-disconnect for all users. DISLOG *line-number* enables it only for the specified line. (You can specify only one line number per each DISLOG directive, but you can specify as many DISLOG directives as you need. The value of *line number* may be up to $1000_8$, which is 512 decimal.) DISLOG NO serves as an indication that automatic logout-on-disconnect is not desired, but neither enables nor disables it for any line. NO is the default.

Automatic logout is useful for installations with port selectors or dialup modems. Use the SET_ASYNC -DISLOG command, or specify DISLOG YES or DISLOG *line-number*. In addition, specify DTRDRP if you are using Auto Speed Detect (ASD).

DISLOG YES overrides all other DISLOG directives. DISLOG YES and DISLOG *line-number* override DISLOG NO. DISLOG NO cannot override anything. The relative position of multiple DISLOG directives within the configuration file is unimportant. To disable a DISLOG directive, you must comment it out, or remove it from the file.

NTS lines automatically have logout-on-disconnect, regardless of DISLOG.

### ▶ DTRDRP

Controls the dropping of the DTR (Data Terminal Ready) signal associated with an asynchronous line.

If specified, the DTRDRP directive automatically forces the dropping of the DTR for any user when that user logs out, regardless of the period set by the *gracetime* value of the AMLTIM directive.

DTRDRP is useful only for installations using Auto Speed Detect (ASD) with port selectors or dialup modems. (Users who have logged out can also issue the PRIMOS DROPDTR command explicitly.)

### ▶ ERASE [*character* | *octal-value*]

Sets the system default erase character. *character* is any printing ASCII character. *octal-value* is the octal value of any ASCII character.

The ERASE directive accepts either a printing character or an octal number as its argument. In this example, either of the following directives sets the system default erase character to the exclamation mark (!):

```
ERASE !
ERASE 241
```

If you omit the ERASE directive, the default erase character is the double quotation mark ("), which is $242_8$. The only two characters normally recommended as the erase character are (") and backspace. Use `ERASE 210` to set the erase character to the BACKSPACE key.

The ERASE and KILL directives take effect upon being processed (rather than later in the startup), with the following results:

- If the SYSNAM directive is omitted from the configuration file, the operator may use the specified erase and kill characters to correct errors when answering the `Enter system name` prompt.
- If the SYSNAM directive is present, but PRIMOS prompts the operator because SYSNAM does not have a valid argument, the specified erase and kill characters are in effect only if the ERASE and KILL directives come before the SYSNAM directive in the configuration file.
- In the unlikely event that an erase or kill character is used in a later directive in the configuration file, it erases or kills on that line in the file. For example, the following directives would result in a system name of SY3, rather than SYS#3.

```
ERASE #
SYSNAM SYS#3
```

### ▶ GO

Marks the end of the configuration file.

Any subsequent lines are ignored after this directive. The configuration file must include the GO directive as the last noncomment line of the file.

### ▶ ICS CARDS *device-address config-word*

Checks the asynchronous Line Adapter Card (LAC) configuration of an ICS2 or ICS3 controller.

The ICS CARDS directive causes PRIMOS to check the actual configuration of an ICS2 or ICS3 controller at both cold and warm starts. An error message is displayed if the actual

configuration differs from that specified by this directive. (For these error messages, see the *ICS User's Guide* or the handbook for your particular computer model.) The differences are due to unexpected, faulty, or missing LACs.

Whether or not you use the ICS CARDS directive, ICS2 and ICS3 asynchronous configurations are maintained from cold start to shutdown (including across warm starts). However, you should use the ICS CARDS directive to determine that the ICS2 and ICS3 asynchronous configurations have not changed.

If the ICS CARDS directive is omitted for any ICS2 or ICS3 on your system, its configuration is not checked at cold start.

The arguments have the following values and meanings:

*device-address* The address of the ICS2 or ICS3 controller. Valid values are 10, 11, 15, 16, 17, 32, 35, 36, 37, 52, 53, and 54.

*config-word* The octal conversion of a 16-bit word in which each bit represents a slot in the LAC Card Cage. A bit with a value of 1 means that an asynchronous LAC is present in that slot. A bit with a value of 0 means that either a synchronous LAC is present in that slot or the slot is empty. The *System Administrator's Guide, Volume II: Communication Lines and Controllers* explains how to calculate *config-word*.

For further details on ICS CARDS, see the *ICS User's Guide*.

### ▶ ICS INPQSZ *queuesize*

Sets the size of all the ICS input queues.

| *Parameter* | *Minimum* | *Maximum* | *Default* | *Recommended* |
|---|---|---|---|---|
| *queuesize*, octal | 17 | 1777 | 77 | 77 |

The default size of the ICS input queues is 77 halfwords (63 decimal). Data is stored one character per halfword.

*queuesize*, which is the octal length of the queue, must be no greater than 1777 and must be equal to one less than a power of two. Examples of possible queue sizes are 17, 37, 77, 177, 377, 777, and 1777.

Specifying an invalid value causes cold start to fail and an appropriate error message to be displayed.

See the *System Administrator's Guide, Volume II: Communication Lines and Controllers*, for further details on configuring ICS lines.

### ▶ ICS INTRPT *rate*

Sets the async interrupt rate for ICS controllers.

| *Parameter* | *Minimum* | *Maximum* | *Default* | *Recommended* |
|---|---|---|---|---|
| *rate*, octal | 12 | 144 | 12 | See below. |

*rate* is an integer specifying the number of interrupts per second. The default and minimum values are both 12 (10 decimal), representing a 100-millisecond interrupt rate. The maximum value is 144 (100 decimal), representing a 10-millisecond interrupt rate.

**Calculating Interrupt Rates:** To set a value between 12 and 144, either use the table below or divide 100 by the desired interrupt rate in milliseconds and truncate the result. Then convert the result (which equals the number of 10-millisecond intervals between interrupts) into octal. The interrupt rate must be in multiples of 10 milliseconds. The *Equivalent Asynchronous Baud Rate* column in the table shows what the last asynchronous line would be set at for equivalent performance.

| *Number of Interrupts per Second* | | *Interrupt Period (milliseconds)* | *Equivalent Asynchronous Baud Rate* |
|---|---|---|---|
| *Octal* | *Decimal* | | |
| <12 | <10 (error) | 100 | 100 |
| 12 | 10 | 100 | 100 |
| 13 | 11 | 90 | 111 |
| 14 | 12 | 80 | 125 |
| 15-16 | 13-14 | 70 | 143 |
| 17-20 | 15-16 | 60 | 167 |
| 21-24 | 17-20 | 50 | 200 |
| 25-31 | 21-25 | 40 | 250 |
| 32-41 | 26-33 | 30 | 333 |
| 42-62 | 34-50 | 20 | 500 |
| 63-144 | 100 | 10 | 1000 |
| >144 | >100 (error) | 10 | 1000 |

**Errors:** If you specify a value for *rate* that is less than 12, the default rate of 12 is used. If you specify a value that is greater than 144, *rate* is set to 144. In either case, the error message `BAD ICS DIRECTIVE: INTRPT` is displayed and cold start continues.

### ▶ KILL [*character* / *octal-value*]

Sets the system default kill character. *character* is a printing ASCII character. *octal-value* is the octal value of any ASCII character.

The KILL directive accepts either a printing character or an octal number as its argument. In this example, either of the following directives sets the system default kill character to the question mark (?):

```
KILL ?
KILL 277
```

If you omit the KILL directive, the default kill character is the question mark (?), which is 277. This is the character usually recommended as the system default kill character. In some circumstances another character, such as DEL (also known as RUBOUT), may be appropriate. Use `KILL 377` to set the system default kill character to the DEL key.

The ERASE and KILL directives take effect upon being processed (rather than later in the startup). For more information, see the ERASE directive earlier in this chapter.

### ▶ LHC *number address*

Sets the physical address assignments for LAN Host Controllers to agree with their physical location in the backplane. The most recent LHC directive overrides any existing address assignments.

The arguments have the following values and meanings:

*number* — Indicates the logical number assigned to the LHC in the NTS configuration file. *number* ranges from 0 to 7

*address* — Specifies the LHC's physical device address in octal. Valid addresses are 10, 11, 15, 16, 17, 32, 35, 36, 37, 50, 51, 52, 53, 54, and 56, provided these locations are not occupied by another controller.

Use the LIST_COMM_CONTROLLERS command to display the current logical controller number and the octal device address. The two-digit octal device address can be obtained with the STATUS COMM command as well.

See the *System Administrator's Guide, Volume II: Communication Lines and Controllers*, for more information on the LHC directive.

### ▶ LOGBAD [YES / NO]

Enables the printing of messages about unsuccessful login attempts.

If LOGBAD is enabled with the YES argument, any login attempt that is unsuccessful (due to an unrecognized user ID, incorrect password, or incorrect project) causes a message to be displayed on the supervisor terminal.

The default is NO, which does not cause messages to be displayed at the supervisor terminal about unsuccessful login attempts.

### ▶ LOGLOG [YES / NO]

Allows the use of the LOGIN command while logged in.

YES specifies that users can use the LOGIN command while logged in. If a user logs in on a terminal that already has a logged-in user, the logged-in user is first logged out and then the new user is logged in. YES is the default.

NO specifies that the LOGIN command is inhibited for a logged-in user.

### ▶ LOGMSG [YES / NO]

Enables the display of login and logout messages.

YES specifies that a message be displayed at the supervisor terminal when a user logs in or logs out. YES is the default.

NO specifies that login and logout messages are suppressed.

If you use LOGMSG, have many users logging in and out frequently, and have your supervisor terminal running at 300 or 1200 baud, you may want to enlarge the output buffer size of the supervisor terminal to increase its efficiency. See the ASRBUF directive earlier in this chapter.

### ▶ LOTLIM *minutes*

Specifies login time limit in minutes.

| *Parameter* | *Minimum* | *Maximum* | *Default* | *Recommended* |
|---|---|---|---|---|
| *minutes*, octal | 1 | none | 3 | 3 |

*minutes* is the octal number of minutes allowed for a user to log in. The minimum value is any value greater than 0. The default is three, which is recommended for most systems, because it gives users adequate time to type, and prevents wastage of system resources. There is no maximum value for *minutes*, but the value should be less than the time allowed by LOUTQM.

### ▶ LOUTQM *minutes*

Specifies inactivity time for automatic logout.

| *Parameter* | *Minimum* | *Maximum* | *Default* | *Recommended* |
|---|---|---|---|---|
| *minutes*, octal | 2 | none | 1750 | 144 |

*minutes* is the number of minutes of inactivity (minus 1) allowed at the terminal before the user is automatically logged out. The value should be greater than 1. The default is $1750_8$ (1000 decimal minutes, which is 16 hours and 40 minutes).

### ▶ MAPOUT bad-page

The MAPOUT directive maps the page having an octal page number of *bad-page* out at cold start, so that it is not available.

Whenever an ECCC memory error occurs, PRIMOS attempts dynamically to map out the page in which it occurred. ECCC stands for Error Correction Code Correctable, and refers to correctable single-bit memory errors.

You should use the MAPOUT directive on any page that maps out dynamically due to repeated ECCCs, as shown in the DSM log. If a wired page generates an ECCC, you must use MAPOUT to map it out, because wired pages cannot be mapped out dynamically. The error messages in the DSM log or on the supervisor terminal provide the octal page addresses for *bad-page*.

The configuration file may contain as many instances of MAPOUT as are necessary (to a maximum of 256). You cannot map out the low-numbered physical pages that contain the coldstart (PRIMOS kernel) image; a service call is required.

### ▶ MEMHLT [YES / NO]

Controls the handling of an ECCU (Error Correction Code Uncorrectable), which is a two-bit memory parity error.

YES, which is the default, halts the system when an ECCU occurs.

If NO is specified and certain conditions are met, PRIMOS determines which user process encountered the ECCU, logs out that process, displays a message at the supervisor terminal listing the ID of the process, and continues normal operation. The form of the message is as follows:

```
User 48 (NETMAN) logged out due to a memory parity error.
```

The following conditions must be met for the single failing user process to be logged out.

- The process that is failing must not be User 1.
- The system must not be in process exchange (switching between processes).
- The process that is running must be executing in Ring 3.
- The page that took the ECCU must not be wired.
- If the page that took the ECCU is shared by more than one user, an up-to-date copy of the page must be on disk.

See the following caution.

---

**Caution**

Systems running ROAM-based data management products (DBMS, DISCOVER, PRISAM) should have MEMHLT YES in the configuration file and should be cold started after any system halt. A warm start may cause loss of data.

---

### ▶ MIRROR

Enables disk mirroring, but does not turn on any mirroring at system startup. See COMDVM and PAGINM. (Use the MIRROR_ON command to start the mirroring of file system partitions.) See the *Operator's Guide to File System Maintenance* for further information on mirroring.

### ▶ MTRS *size1 size2*

Specifies the maximum tape record size allowed on each tape controller.

| *Parameter* | *Minimum* | *Maximum* | *Default* | *Recommended* |
|---|---|---|---|---|
| *max-size*, octal | 14000 | 40000 | 14000 | As required. |

*size1* specifies the largest magnetic tape record size, in halfwords, that the first tape controller can handle. The smallest value, $14000_8$, is $6144_{10}$ halfwords, or 12 kilobytes. The largest value, $40000_8$, is $16384_{10}$ halfwords, or 32 kilobytes. *size2* specifies the largest magnetic tape record size, in halfwords, that the second tape controller can handle. The largest and smallest values are the same as those for the first controller. If you leave out the second size *size2*, the second tape controller is automatically assigned the same values as the first tape controller.

The increased maximum record size allows programs, via the T$MT call, to perform tape I/O with tapes made on other systems that use a greater record size than was supported by pre-Rev. 22.0 PRIMOS.

With enhanced MAGSAV/MAGRST, if you want to write tapes with 16K blocks you must set MTRS to 16K bytes or greater. That translates to $20000_8$ or greater.

If the value supplied to the MTRS directive is not within the specified range, the default value of 14000 is used and an error message appears during cold start:

```
Value of MTRS, nn, is out of range.
The default value, 14000, will be used (CINIT).
```

The value *nn* is the invalid number in the CONFIG file.

---

**Caution**

If you use MTRS, you should keep in mind that when you increase the record size, the system will use up more pages of segment 0 to support the larger tape records. (PRIMOS uses segment 0 pages to perform I/O for various devices it supports.) Each device gets exclusive use of its segment 0 pages, and these pages are allocated at cold start. It is possible that selecting a large record size may cause PRIMOS to run out of segment 0 pages. The result would be that a device that works correctly when MTRS is the default value would fail when a larger value is used, or that PRIMOS will fail at cold start.

---

### ▶ NAMLC *number-of-lines*

Specifies the maximum number of directly-connected assignable asynchronous lines.

| *Parameter* | *Minimum* | *Maximum* | *Default* | *Recommended* |
|---|---|---|---|---|
| *number-of-lines*, octal | 0 | See below. | 0 | See below. |

*number-of-lines* specifies the maximum number of directly-connected asynchronous lines that may be assigned simultaneously on your system. Assignable NTS lines are not handled by NAMLC. The default is 0. NAMLC + NTUSR + NTSUSR + NTSASL must be less than or equal to $2000_8$ (1024 decimal).

The SET_ASYNC command, described in the *System Administrator's Guide, Volume II: Communication Lines and Controllers*, specifies which lines are assignable.

### ▶ NLBUF *buffers*

Specifies the number of locate buffers to be configured.

| *Parameter* | *Minimum* | *Maximum* | *Default* | *Recommended* |
|---|---|---|---|---|
| *buffers*, octal | 10 | 11610 | 100 | See Chapter 7 |

*buffers* is the number of locate buffers for which the system is to be configured. Minimum value is $10_8$ (8 decimal) and maximum is $11610_8$ (5000 decimal). The default is $100_8$ (64 decimal). Each locate buffer occupies 2048 bytes (2KB) of memory.

At cold start PRIMOS checks whether the configured value of NLBUF fits the available memory, and displays a warning message if it does not. When PRIMOS is in operation, a background process, BUFFER_SERVER, monitors the availability of locate buffers. Details of the messages and the process are in Chapter 7.

### ▶ NPUSR *number*

Sets the number of phantom users.

| *Parameter* | *Minimum* | *Maximum* | *Default* | *Recommended* |
|---|---|---|---|---|
| *number*, octal | 4 | 1700 | 4 | 10 or more |

*number* is the number of phantom users for which the system is to be configured. (*number* must be a positive octal integer.) The default value of 4 is also the minimum. If you configure fewer phantoms than this number, PRIMOS prints an error message and uses the default value instead. The maximum is $1700_8$ (960 decimal) minus the number of terminal, slave, and remote users (NTUSR, NSLUSR, NRUSR, and NTSUSR).

If you have PRIMENET, you must configure a phantom user for NETMAN. If your system is to be a gateway node, you must also configure a phantom for RT_SERVER. Phantoms are also required for despoolers, the Batch subsystem, the File Transfer Service (FTS), and other purposes.

### ▶ NRUSR *number*

Specifies the number of processes to be reserved for remote logins from the PRIMENET network.

| *Parameter* | *Minimum* | *Maximum* | *Default* | *Recommended* |
|---|---|---|---|---|
| *number*, octal | 0 | 1604 | 0 | As needed. |

*number* is the number of remote users for which the system is to be configured. (*number* must be a nonnegative octal integer.) The default is 0. The maximum is $1604_8$ (900 decimal). NTSUSR+NTUSR+NPUSR+NRUSR+NSLUSR cannot exceed $1700_8$ (960 decimal). NRUSR does not apply to terminals connected through modems to asynchronous AMLC or ICS lines.

### ▶ NSEG *number*

Specifies the number of segments available for all users.

| *Parameter* | *Minimum* | *Maximum* | *Default* | *Recommended* |
|---|---|---|---|---|
| *number*, octal | See below. | 40000 | 1776 | See below. |

*number* specifies the number of page maps to be allocated during system initialization. The maximum value is $40000_8$ (16384 decimal). The default, $1776_8$ (1022 decimal), is often too small. If you set *number* larger than $30000_8$, the NVMFS directive may interfere with NSEG. See NVMFS later in this chapter.

NSEG must guarantee at least three segments per configured process. Calculate the minimum value of *number* with the following formula:

*number* => 3 * (NTUSR + NPUSR + NRUSR + NSLUSR)

If this minimum is not met, the following warning message is displayed during cold start:

WARNING - *m* SEGMENTS MAY NOT BE ENOUGH FOR *n* USERS

where *m* is the number of segments and *n* the number of processes configured. Cold start then continues.

Increase the default value if the condition NO_AVAIL_SEGS$ is frequently signaled on your system.

More page maps may be available than the number of possible user segments (based on available space on the paging partition). If a process cannot get paging space for this reason, the error condition PAGING_DEVICE_FULL$ is signaled.

### ▶ NSLUSR *number*

Sets the number of slave processes (users).

| *Parameter* | *Minimum* | *Maximum* | *Default* | *Recommended* |
|---|---|---|---|---|
| *number*, octal | 0 | 1440 | 0 | As needed. |

Each user accessing files on the local system from remote systems requires a slave process for the duration of the access. These slave processes are allocated from the PRIMOS pool of 960 processes.

*number* is the number of simultaneous remote file accesses the local system wishes to support. If this pool is exhausted when a remote user makes an attach request, the E$NSLA (no NPX slaves available) error code is returned to that user.

The minimum value for *number* is 0, which is the default. The maximum value is $1440_8$ (800 decimal). NTUSR+NPUSR+NRUSR+NSLUSR+NTSUSR must be less than or equal to $1700_8$ (960 decimal) on 4000 series and 6000 series machines as well as on the 2850 and 2950 machines. On all other systems this limit is $1130_8$ (600).

### ▶ NTSASL *number*

Sets the maximum number of NTS assignable lines.

*number* — The maximum number of NTS assignable lines that can be used simultaneously on your system. The default is 0. The sum of NAMLC + NTSASL + NTUSR + NTSUSR must be less than or equal to $2000_8$ (1024).

### ▶ NTSUSR *number*

Specifies maximum number of simultaneous NTS terminal users.

*number* is the total number of NTS terminal users. The default is 0. The sum of NTUSR+NPUSR+NRUSR+NTSUSR+NSLUSR must be less than or equal to $1700_8$ (960) on a 2850, 2950, or the 4000 series or 6000 series systems. On all other systems this limit is $1130_8$ (600). The sum of NAMLC + NTSASL + NTUSR + NTSUSR must be less than or equal to $2000_8$ (1024).

### ▶ NTUSR *number*

Specifies number of terminal users, including the supervisor terminal.

| *Parameter* | *Minimum* | *Maximum* | *Default* | *Recommended* |
|---|---|---|---|---|
| *number*, octal | 1 | 1000 | NONE | As needed. |

*number* is the number of directly-connected terminal users for which the system is to be configured. (Do not include network users in this number.) The maximum value of *number* is $1000_8$ (512 decimal). NTUSR, which has no default value, must be included in the configuration file.

NTUSR+NPUSR+NRUSR+NSLUSR+NTSUSR must be less than or equal to $1700_8$ (960 decimal) on a 2850, 2950, or on 4000 series or 6000 series systems. On all other systems this limit is $1130_8$ (600). NTUSR+NAMLC+NTSUSR+NTSASL must also be less than or equal to $2000_8$ (1024 decimal). The sum of NAMLC+NTSASL+NTUSR+NTSUSR must be less than or equal to $2000_8$ (1024).

### ▶ NVMFS *number*

Sets the number of VMFA (Virtual Memory File Access) dynamic segments available in virtual address space for the system.

| *Parameter* | *Minimum* | *Maximum* | *Default* | *Recommended* |
|---|---|---|---|---|
| *number*, octal | 1 | 10000 | 144 | 400 |

The maximum value for *number* is $10000_8$ (4096 decimal). The default is $144_8$ (100 decimal).

The total number of segments available for both NSEG and NVMFS is $40000_8$ (16384 decimal). If you specify values for NSEG and NVMFS that total higher than the $40000_8$ (16384 decimal) maximum, NVMFS takes precedence over NSEG. For example, if you specify NSEG as $37200_8$ (16000 decimal) and specify NVMFS as $620_8$ (400 decimal), the system is configured with $620_8$ (400 decimal) NVMFS segments available and $37160_8$ (15984 decimal) NSEG segments.

VMFA segments are used by EPFs because they can be mapped dynamically. You may want to increase the default number of VMFA segments if users frequently complain that they get messages such as `Not enough segments` or `No space available from process class storage heap.`

### ▶ PAGING *pdev1* [... *pdev8*]

Specifies the paging partitions.

PAGING must be included in the configuration file. A minimum of one paging partition is required, and a total of eight are allowed. *pdev1* through *pdev8* are the physical device numbers of the paging partitions. See the *Operator's Guide to File System Maintenance* for details on construction of physical device numbers.

PRIMOS automatically calculates a strategy to divide paging activity among the available paging partitions. Paging activity is allocated to each paging partition in the ratio of its size to the total size of all the paging partitions. To adjust these ratios, use the PRATIO command, described in the *Operator's Guide to System Commands.*

If all available paging space is used, any attempt by a user process to obtain more memory causes the error condition PAGING_DEVICE_FULL$ to be signalled for that user.

### ▶ PAGINM *pdev1* [... *pdev8*]

Specifies *pdev1* through *pdev8* as the physical device numbers of the partitions that mirror the corresponding paging partitions specified in the PAGING directive and enables mirroring of file system partitions. This directive turns on paging device mirroring at system startup.

If a *pdev* in the PAGING directive is not to be mirrored, then a 0 must be used in the corresponding position in the PAGINM directive. The PAGINM directive, if used, must follow the PAGING directive in the configuration file. See the *Operator's Guide to File System Maintenance* for further information on mirroring.

### ▶ SMLC

At Rev. 20, SYNC became the preferred synonym for SMLC. For details on the SMLC directives, see the discussions of the SYNC directives, in the *System Administrator's Guide, Volume II: Communication Lines and Controllers.*

### ▶ SYNC

Here is a summary of the purposes of the four SYNC directives. For extensive and complete information, see the *System Administrator's Guide, Volume II: Communication Lines and Controllers.*

| | |
|---|---|
| SYNC CNTRLR | Associates a device address with a logical controller number and a communications protocol. |
| SYNC DSC | Specifies data set control information for SYNC, HSSMLC, or MDLC controllers. |
| SYNC ON | Enables configuration of synchronous communication drivers. |
| SYNC SYNCnn | Associates a controller's physical line with a logical line number. |

### ▶ SYSNAM *name*

Specifies *name* as the name of the system. This directive is required.

The system name uses the PRIMENET nodename syntax: *name* must be from one to six characters, of which the first is a letter, between A and Z. The other characters may be letters or numerals, or any of the seven characters **&-$._/#**. If PRIMENET or NTS is used, the system name becomes the network nodename.

If you specify SYSNAM with no *name*, PRIMOS prompts the operator to enter a system name at the supervisor terminal. If you omit SYSNAM, or specify a *name* that is too long or contains invalid characters, PRIMOS prints an error message and prompts for the system name.

The erase and kill characters in effect when the operator responds to the system name prompt depend upon the relative placement of the ERASE, KILL, and SYSNAM directives within the configuration file. See the discussion of erase and kill processing under the ERASE directive earlier in this chapter.

### ▶ TYPOUT [YES / NO]

Controls displaying of configuration directives at the supervisor terminal.

YES specifies that subsequent directives in the configuration file be displayed at the supervisor terminal as they are processed. Displaying continues until a TYPOUT NO or GO directive is encountered.

NO specifies that commands are not displayed as they are processed. Displaying is suppressed until a TYPOUT YES or GO directive is encountered. TYPOUT NO is the default.

You can put any number of TYPOUT directives into the configuration file to display selected directives.

### ▶ UPS *number*

Controls restart after a power failure.

| *Parameter* | *Minimum* | *Maximum* | *Default* | *Recommended* |
|---|---|---|---|---|
| *number*, octal | 0 | 177777 | 177777 | 100 |

An Uninterruptible Power Supply (UPS) maintains power to the CPU and memory during a power failure and then, on some systems, automatically performs a warm start. If your system has UPS, the value of *number* in the UPS directive determines what action is taken after the warm start.

*number* has the following values and meanings:

177777 — No UPS (default).

0 — Produces a warm start followed by a halt. The operator must intervene to bring up the system.

> 0 — Number of seconds, in octal, to delay after the warm start before the system comes up. No operator start is required. The number of seconds to delay after a warm start is the amount of time it takes for the disks to come up to the proper number of revolutions per minute. A value of $100_8$ (64 decimal seconds) is recommended for a storage module.

If your system does not have UPS, or if it does not automatically warm start, you should omit this directive.

### ▶ WIRMEM

Prints the size of wired memory (in kilobytes) at the supervisor terminal during cold start.

The size of wired memory changes during the operation of the system. The value displayed by WIRMEM, however, gives some idea of the relative memory cost of the selected configuration. The USAGE command tracks changing wired memory requirements.

# 9

# THE SCHEDULER

## INTRODUCTION

In an interactive, timesharing system, such as PRIMOS, the CPU is shared by many processes. A **process** defines the environment in which programs are executed; each terminal user, phantom, or slave is associated with a different process. Since a CPU can execute instructions from only one process at a time, the illusion of a shared CPU is implemented by limiting the amount of time any process may use the CPU before that process must wait for another turn.

The following discussion uses three classifications of processes:

- An **active process** is one that is currently using the CPU or has paused to wait for an I/O operation to complete.
- An **inactive process** is one that has nothing to do (for example, a terminal user process that is awaiting a new command following the "OK," prompt).
- A **pending process** is either one that has exhausted its allotment of CPU time and is waiting for another turn or a previously inactive process that is waiting for its first turn.

The **scheduler** controls the access that processes have to the CPU resources of the system. It manages the allotment of CPU time, selects pending processes to be made active, and limits the number of active processes on the system. Process scheduling, on Prime machines, depends on three components: the SCHED routine, the Backstop process, and the microcode Dispatcher.

- The **SCHED** routine (part of PRIMOS) converts an active or an inactive process to a pending process.
- The **Backstop** (part of PRIMOS) converts a pending process into an active one.
- The **Dispatcher** (part of firmware) controls which active process is executing on the CPU.

The scheduler functions according to certain parameters or attributes, some of which can be tuned by the System Administrator. This chapter covers only the SCHED and the Backstop components because the Dispatcher is part of the 50 Series™ architecture and is not under the control of the System Administrator. For more information on the Dispatcher, consult the *System Architecture Reference Guide.*

This chapter presents a basic explanation of how process scheduling works and discusses the differences in scheduler functioning and tuning before Rev. 23.0 and at Rev. 23.0. Although System Administrators have always been able to use certain commands to tune some scheduler parameters, at Rev. 23.0 the ability to tune the scheduler is expanded and refined.

# BASIC SCHEDULER CONCEPTS

## Timeslices

A **timeslice** is a predetermined period of CPU time for which a process is allowed to execute before it becomes a pending process. The **minor timeslice** is the amount of time a process can execute before becoming a pending process on the EligQ. (The EligQ and the other scheduler queues are explained below.) The default value of the minor timeslice depends on the type of CPU. (The default values are listed under the ELIGTS command in the *Operator's Guide to System Commands.*) The System Administrator can modify this systemwide value using the ELIGTS command or the -ELIGTS option to the SET_SCHEDULER_ATTRIBUTES (SSA) command. (The SSA command is described briefly later in this chapter and is fully documented in the *Operator's Guide to System Commands.*)

The **major timeslice** is the total amount of time a process can execute before becoming a pending process on the LoPriQ. The default value of the major timeslice depends on the type of CPU and is six times the default minor timeslice. (The default values are listed under the CHAP command in the *Operator's Guide to System Commands.*) The System Administrator can modify this value on a per-process basis using the CHAP command. To a limited degree, a user can also modify the value of his or her own major timeslice. See the *Operator's Guide to System Commands* and the *PRIMOS Commands Reference Guide* for more information on the CHAP command.

## Scheduler Queues

There are three **scheduler queues** where a pending process might wait for a turn to become active. These are the **HiPriQ** (high priority queue), **EligQ** (eligibility queue), and **LoPriQ** (low priority queue). Which queue a pending process uses depends, in part, on the status of that process's major timeslice.

The scheduler favors interactive processes, such as terminal users, over compute-intensive processes, like BATCH. When a user signals his or her associated process to execute a command (usually by pressing the RETURN key), the SCHED routine is invoked. SCHED insures that the process has a full minor and major timeslice, and that it becomes a pending process on the HiPriQ.

When a process has executed on the CPU for its minor timeslice, the SCHED routine is invoked again. If there is time remaining in the major timeslice, the process becomes a pending process on the EligQ with a new minor timeslice. If there is no major timeslice left, the process is given new minor and major timeslices and becomes a pending process on the LoPriQ.

The **Backstop** executes when no active process is using the CPU and scans the scheduler queues in order to select a pending process to make active. By default, if there are processes pending on the HiPriQ, the Backstop will make one of them active. If there are no processes pending on the HiPriQ, but there are processes on the EligQ, then an EligQ process is made active. A process pending on the LoPriQ only becomes active when both the HiPriQ and the EligQ are empty.

## Priority Levels

To help control the order in which processes execute, they are assigned **priority levels.** There are five priority levels, numbered 4 through 0. The highest level (level 4) is available only to system processes. Levels 3 through 0 are for user processes. The default priority level for users is 1. All users are at level 1 when they log in unless the default level has been reset by the CHAP command.

Priority levels control the order in which processes are made active. By default, when two processes of different priority levels are pending on the HiPriQ or EligQ, the process at the higher priority level becomes active first. However, in order to make sure that all processes get a turn to be made active from the LoPriQ, priority levels within the LoPriQ are assigned quotas that determine the maximum number of pending processes that may be made active from that level of the queue before the next lower priority level is checked.

In addition, there are two special purpose priority levels: IDLE and SUSPEND. The IDLE priority level can be applied only to phantoms. Phantoms at the IDLE priority level run only when no process is executing and no process is waiting to be made active. The SUSPEND priority level completely prevents a process from using CPU resources until its level is explicitly changed to some other level. Priority levels are modified on a per-process basis with the CHAP command. (For more information on priority levels, see the CHAP command in the *Operator's Guide to System Commands* and in the *PRIMOS Commands Reference Guide.*)

# IMPROVEMENTS TO THE SCHEDULER AT REV. 23.0

Prior to Rev. 23.0, there were some circumstances in which the scheduler provided less than optimal performance. This section describes three deficiencies in the scheduler that have been corrected at Rev. 23.0.

## Lack of Fairness Among Priority Levels

At times, it was possible for a lower priority process to run before and to be scheduled more frequently than a higher priority process. This was because prior to Rev. 23.0, each priority level had a permanent quota. At Rev. 23.0, there is a new method of assigning quotas to each priority level. The Backstop periodically recalculates quotas for each priority level of the LoPriQ based on the number of pending processes at that particular level.

As a result of these changes, a higher priority process consistently gets more access to CPU resources than a lower one doing a similar task. The System Administrator can now use priority levels predictably to control the relative access that a process or group of processes has to CPU resources.

## Process Starvation

Process starvation occurs when there are so many processes arriving at the HiPriQ that a process pending at the EligQ never becomes active, or when there are so many processes at the HiPriQ and the EligQ that a process on the LoPriQ never becomes active. Prior to Rev. 23.0, this sometimes happened because quotas were not used for the HiPriQ and the EligQ. This meant that as long as a process was pending on either one of these queues, that process always became active before a process on the LoPriQ. With a high arrival rate at the HiPriQ and the EligQ, processes take a long time to become active when pending on the LoPriQ.

Usually, not having quotas on the HiPriQ and the EligQ is desirable because it provides infinite service for interactive tasks, providing them with quicker access to CPU resources than that available to other tasks. Infinite service only becomes a problem when it causes process starvation, thus making some processes wait a very long time or keeping them from ever getting serviced at all.

Prior to Rev. 23.0, the only solution to this problem was to lower the minor and/or major timeslice to force processes down to the EligQ and LoPriQ more frequently. This increases system overhead, however, by causing more process exchange.

At Rev. 23.0, there is another solution to process starvation. You can cause quotas to be used on the HiPriQ and the EligQ by using the -SHORT_JOB or -QUEUE_RATIO options to the SET_SCHEDULER_ATTRIBUTES command. (By default the scheduler still provides infinite service to these queues.)

## Limited Tunability

Prior to Rev. 23.0, the System Administrator was able to use only three commands to tune the scheduler parameters: ELIGTS, CHAP, and MAXSCH.

- The ELIGTS command is used to adjust the value of the minor timeslice. The default value was set fairly low to help prevent process starvation but the System Administrator could alter that value if the default did not work for his or her system. Keeping the value low caused more processes to move down to the EligQ and the LoPriQ.
- The CHAP command is used to set users' priority levels and to adjust the major timeslices. The effect of user priority levels has already been discussed. The larger the major timeslice, the greater the number of times a process becomes a pending process on the EligQ before becoming a pending process on the LoPriQ. For some tasks, the major timeslice of an individual process can be increased, to decrease the elapsed time required to complete that task.
- The MAXSCH command controls the number of processes that may be active on the system simultaneously. The purpose of this command is to prevent the system from thrashing, which can occur if available memory resources are overcommitted. Whenever the number of active processes equals or exceeds the value of MAXSCH, pending processes on the EligQ and LoPriQ will not become active.

For more information on these three commands see the *Operator's Guide to System Commands*.

While these commands allowed some adjustments to be made, they failed to provide much flexibility and didn't allow the Administrator to specify certain details of scheduler function such as *how much* a higher priority level is favored over a lower one. At Rev. 23.0, it is possible to change the degree of favoritism or bias among priority levels. This is done using the -PRIORITY_BIAS or -PRIORITY_RATIO options to the SET_SCHEDULER_ATTRIBUTES command. By default the priority ratio for levels 0 through 4 is 1 to 2 to 4 to 8 to 16.

## Quota Calculation

As was mentioned earlier, at Rev. 23.0, the Backstop periodically calculates new quotas for each priority level within a queue for any queue with a non-infinite queue ratio. The quota for a priority level of a queue is a function of the number of processes pending at that level of that queue, the queue ratio value for that queue, and the priority ratio value for that priority level. Fairness, process starvation, and tunability are all addressed through this quota calculation function.

## TUNING THE SCHEDULER AT REV. 23.0

At Rev. 23.0, it is possible for Administrators to specify more than just the minor and major timeslice values, priority levels, and the maximum number of processes active on the system at one time. These four scheduler attributes are still tunable, but now two more attributes are tunable by the System Administrator: queue ratio and priority ratio. These attributes affect the use of quotas on the scheduler queues. By allowing input from the System Administrator into scheduler queue quota calculations, greater tuning flexibility is possible than before Rev. 23.0.

There are new tools available to assist in tuning the scheduler as well. Two new commands have been introduced, the SET_SCHEDULER_ATTRIBUTES (SSA) command and the LIST_SCHEDULER_ATTRIBUTES (LSA) command, as well as a new option to the USAGE command, -SCHED, to display information about the scheduler. The SSA and LSA commands are documented in detail in the *Operator's Guide to System Commands.* The SSA command is explained briefly in this chapter.

The -SCHED option to USAGE is also documented in the *Operator's Guide to System Commands* and some general guidelines for interpreting the output of this option are given in the last section of this chapter.

## THE SSA COMMAND

The SSA command allows the System Administrator to tune the PRIMOS scheduler more precisely to better meet the requirements of a particular site. With the SSA command, you can now make adjustments to the scheduler that allow greater control over how CPU resources are distributed among user priority levels and between shorter and longer jobs.

(When determining how to tune the scheduler attributes, refer also to the explanation of the -SCHED option of the USAGE command at the end of this chapter. Use of the -SCHED option to USAGE allows you to gather data on system performance that helps you determine what kind of adjustments to make to the scheduler attributes.)

The options to SSA are explained briefly below. For more information on the command syntax and options, see the *Operator's Guide to System Commands.*

The options to SSA are:

- -QUEUE_RATIO, which determines the relative distribution of CPU resources among the HiPriQ, EligQ, and LoPriQ.
- -SHORT_JOB, which tunes -QUEUE_RATIO to one of five predetermined sets of ratio values. This option adjusts the same ratios as does the -QUEUE_RATIO option, but in a simpler way.

- -PRIORITY_RATIO, which controls the distribution of CPU resources among the five process priority levels. -PRIORITY_RATIO determines how *much* favoritism is given to higher priority processes over lower priority processes.
- -PRIORITY_BIAS, which tunes -PRIORITY_RATIO to one of five predetermined sets of ratio values. This option tunes the same ratios as does the -PRIORITY_RATIO option, but it does so in a simplified way.
- -ELIGTS, which sets the systemwide parameter of ELIGTS in *n* milliseconds. -ELIGTS determines the length of the minor timeslice. The valid range is 4 through 32767. If no argument is specified, the value is set to the default.
- -MAXIMUM_SCHEDULED_JOBS, which sets the systemwide parameter of MAXSCH. -MAXSCH controls the number of processes simultaneously active on the system. The valid range is 1 through the maximum number of configured users.
- -HELP, which displays command syntax.

## Queue Ratio

Queue ratio is the relative ratio of CPU service received by each of the three queues. The default values are 1 (LoPriQ) to infinite (EligQ) to infinite (HiPriQ). (These are the same values in effect by default in the pre-Rev. 23.0 scheduler.)

The default settings simply mean that the HiPriQ and EligQ are serviced until there are no more processes pending on them, and then one process is made active from the LoPriQ. As mentioned earlier in this chapter, having values of infinite for both the HiPriQ and EligQ is often desirable, but only if it doesn't lead to process starvation.

**The -QUEUE_RATIO and -SHORT_JOB Options:** At Rev. 23.0, you can modify the queue ratio values using one of two options to the SSA command: the -QUEUE_RATIO option or the -SHORT_JOB option.

The -QUEUE_RATIO option gives you the greatest range of choice but is more complex to use. When you use the -QUEUE_RATIO option, you can specify values for the three queues ranging from 1 through 1024. For the HiPriQ and the EligQ an infinite value may also be specified. Using the option with no values sets queue ratios to the default values (explained above).

The -SHORT_JOB option specifies ratios for the three queues as does -QUEUE_RATIO, but in a simpler way. It allows you to choose from among five sets of predetermined ratio values (listed in the *Operator's Guide to System Commands*). -SHORT_JOB with an argument of 4, for example, sets the ratios to the default values (1 to infinite to infinite) and is the setting that most strongly favors short, interactive jobs. A setting of 0 will still favor short, interactive jobs but will allow longer jobs a greater share of CPU resources than the default setting.

If your system is experiencing problems with process starvation, lowering the -SHORT_JOB setting should help to reduce them.

## Priority Ratio

Priority ratio controls the distribution of CPU resources among the five process priority levels, numbered 0 (lowest priority level) through 4 (highest priority level) by determining the *amount* of bias or preference given to higher priority levels over lower priority levels. The default settings are 1 to 2 to 4 to 8 to 16. (These are the settings in pre-Rev. 23.0 versions of PRIMOS.) The default values give more bias to higher priority levels by increasing the amount of favoritism given to each level by a factor of two over the next lower level.

**The -PRIORITY_RATIO and PRIORITY_BIAS Options:** The Rev. 23.0 scheduler allows the System Administrator to adjust the priority ratio using one of two options to the SSA command, the -PRIORITY_RATIO option or the -PRIORITY_BIAS option. (For details on command syntax, see the *Operator's Guide to System Commands.*)

The -PRIORITY_RATIO option allows you to specify values for priority ratio ranging from 1 through 256 for each of the five levels.

-PRIORITY_BIAS allows you to set the priority ratio to any one of five sets of priority ratio values, much the same way that -SHORT_JOB allows you to set -QUEUE_RATIO to one of five predetermined sets of values. -PRIORITY_BIAS is just a simpler way of setting the priority ratio.

The -PRIORITY_BIAS option takes one of five possible arguments: 0, 1, 2, 3, or 4. A value of 0 means that there is no bias among the priority levels: that is, all levels receive the same amount of CPU resources. A value of 4 provides the greatest amount of bias among the priority levels: that is, the service ratio from level-to-level increases by a factor of four. A value of 2 is the default setting and this is equal to the default -PRIORITY_RATIO values (which increase by a factor of two) of 1 to 2 to 4 to 8 to 16. If your system is running at the default, and you want to give more CPU resources to higher priority processes, set the -PRIORITY_BIAS to 3 or 4. If you want CPU resources to be distributed more evenly among the priority levels, set the -PRIORITY_BIAS to 1 or 0.

# THE -SCHED OPTION TO USAGE

There is another tool available to help System Administrators tune the PRIMOS scheduler: the new -SCHED option to the USAGE command. The -SCHED option generates a report that can be helpful in setting queue ratios and priority ratios. It displays the following four categories of information about the HiPriQ, the EligQ, and the LoPriQ:

- Total percent of arrivals – gives the number of arrivals at each queue since cold start as a percentage of the total arrivals at all three queues since cold start.
- Arrivals per second – gives the number of arrivals per second at each queue (or level within a queue) during the interval monitored.
- Aggregate waiters per second – gives the number of processes that waited on each queue (or level within a queue) for any portion of the interval, divided by the number of seconds in the interval.
- Relative delay – gives a relative indication of delay, which is loosely related to the amount of time processes wait at a queue or level within a queue when the system is reasonably loaded. (For systems with low CPU utilization, this quantity is not very meaningful.)

Following is a sample display of USAGE -SCHED output:

```
OK, USAGE -SCHED
[USAGE Rev. 23.0 Copyright (c) 1987, Prime Computer, Inc.]
Type "START" to continue.

OK, START

   System S35
 15 Mar 90 14:35:02.15  dTIME=    9.05    CPU=      4.50    I/O=      0.26
 Up since 15 Mar 90 12:16:32 Thursday  CPUtot=   2652.80 I/Otot=   3979.80

    %CPU   %Idl1   %Idl2  %Error    %I/O   %Ovlp    IO/S    PF/S   PIO/S
   24.87   75.16   70.41    0.44    0.24    0.00    1.21    3.87    0.00

                  Total                 Aggregate  Relative
 Sched. Queue      %Arr      Arr/S     Waiters/S     Delay
 ---------------  ------  ---------  ------------  --------
 High Priority     46.21      12.26         12.26
 Eligibility       44.42       2.32          2.32
 Low Priority       9.37       0.66          0.77      3.88
       level 4      0.65       0.00          0.00      0.00
       level 3      1.06       0.00          0.00      0.00
       level 1      7.61       0.66          0.66      4.53
       level 0      0.05       0.00          0.11      0.00
OK,
```

## Using USAGE -SCHED to Help Tune the Scheduler

In general, you should leave the scheduler at the default value for queue ratio as long as process starvation is not occurring. Assign process priority levels with the CHAP command to give higher priority users more CPU resources. You can also combine the use of CHAP with the use of the -PRIORITY_RATIO or -PRIORITY_BIAS options to specify the *degree* of bias the higher priority levels receive.

The USAGE -SCHED display may indicate when further tuning might be useful. For example, it can help you determine when your system is experiencing process starvation. Or it could tell you what kind of jobs are running on your system (e.g., interactive, compute-intensive, or some combination) and, therefore, how to adjust for them. These and other examples are explained further in the following paragraphs.

---

**Caution**

Be alert to factors that may cause the USAGE -SCHED display to give misleading figures. For example, applications that call RECYCL can influence the numbers reported by the USAGE -SCHED option because RECYCL causes a process to wait on the LoPriQ. On a lightly loaded system, RECYCL could be called many times in a short interval depending on what applications are running. If this happens, it can skew the output you receive for USAGE -SCHED and make it seem as if many longer jobs are running on your system, when actually there are just many jobs that frequently call RECYCL. ECL is an example of an application that calls RECYCL a great deal, possibly resulting in misleading output in the USAGE -SCHED display.

---

**Process Starvation:** Process starvation can be detected by monitoring with USAGE -SCHED. Starvation is occurring during the sample period when the aggregate waiters per second for that period is equal to the arrivals per second for that period plus the aggregate waiters per second for the previous period. This conclusion is only valid when you monitor the system for constant time intervals (at least one minute long) and when the percent of CPU utilization is fairly high.

If you determine that you are having a problem with process starvation on your system, generally the simplest thing to do is to use the -SHORT_JOB option with the DOWN argument. This forces the scheduler to notify processes on the LoPriQ more frequently by adjusting the three queue values to the next lowest -SHORT_JOB level. You can also use any of the other -SHORT_JOB arguments, or use -QUEUE_RATIO if -SHORT_JOB does not provide the best choice of ratios for your system.

**Relative Delay:** Tuning the priority ratio affects the relative delay among priority levels in a queue. Priority ratio tuning can also be seen in the percent of CPU time used by individual processes. You can use relative delay to judge how good the service is for each level within a queue.

When a system is consuming 100% of the CPU, large increases in relative delay from one sample to the next might indicate that you are trying to use more CPU resources than exist. In this situation, you should use the CHAP command to set one or more less important processes to IDLE or SUSPEND.

**System Characterization:** The number of arrivals per second and, to a lesser extent, the total percent of arrivals can give you information about the kind of jobs running on your system. This information can be useful in achieving your tuning goals. For instance, when you are running almost entirely batch jobs or phantoms, you see very few arrivals at the HiPriQ. (Refer to the Caution above.)

When few or no interactive users are on the system, you can increase the value of the minor timeslice (ELIGTS). This should improve system efficiency by reducing the amount of process exchange. When tuning to favor interactive users (significant arrivals at the HiPriQ), set the value of the minor timeslice to the amount of CPU time required to execute a short, interactive task on your system.

# APPENDICES

# A

# OBSOLETE AND RARELY USED COMMANDS AND DIRECTIVES

This appendix describes two obsolete commands, AMLC, an octal method of configuring asynchronous lines, and LOOK, a debugging tool; five rarely-used configuration directives, FILUNT, MAXPAG, PREPAG, RWLOCK, and VPSD; and eight obsolete configuration directives, ALTDEV, AMLBUF, NTSABF, NTSBUF, REMBUF, PAGDEV, PRATIO, and TPDUMP.

## OBSOLETE COMMANDS

The commands AMLC and LOOK, described below, are obsolete.

### ▶ AMLC

The AMLC command uses octal bitstrings to configure both terminal and assigned asynchronous lines on the AMLC and ICS controllers. At Rev. 20.2, the AMLC command was replaced by the SET_ASYNC command, a more straightforward way of configuring your asynchronous lines. Although the AMLC command is still supported, its use is no longer recommended.

A complete description of the AMLC command may be found in *System Administrator's Guide, Volume II: Communication Lines and Controllers.*

### ▶ LOOK

The LOOK command, which provides access to any segment in the system, is intended as a debugging tool for systems engineers and field analysts. Although System Administrators rarely use the LOOK command, you may use it as a monitoring command.

The LOOK command can be issued only at the supervisor terminal and must be preceded by an OPRPRI 1 command and followed by an OPRPRI 0 command.

The command format of LOOK is as follows:

**LOOK [*-usernumber* [*segnumber* [*access* [*mapseg*]]]]**

The meanings of the parameters are as follows:

| Parameter | Meaning |
|---|---|
| *-usernumber* | Number of the user owning the segment. The default is user 1. The hyphen must precede the number. |
| *segnumber* | Number of the segment to be examined. The default is $6000_8$ (the Ring 0 stack segment for the user). |
| *access* | Access rights to be granted (as in the SHARE command). The default is $200_8$ (read only). |
| *mapseg* | Segment of user 1's address space into which the specified segment is to be mapped. The default is $4001_8$. |

**WARNING**

Misuse of the LOOK command can destroy system data. The LOOK command can place system integrity at risk if you attempt to examine a segment that does not exist, write to a segment that does exist, or map either shared or stack segments with write permission. The REALLY? prompt is issued for a LOOK command whose request is considered to be risky or dangerous to system integrity. A YES response allows the operation to proceed.

## RARELY USED DIRECTIVES

The FILUNT, MAXPAG, PREPAG, RWLOCK, and VPSD configuration directives are not obsolete, but their use should be avoided. These directives are discussed next.

### ▶ FILUNT 0 *max-unit*

Sets the maximum number of file units available to each user.

The arguments have the following values and meanings:

| | |
|---|---|
| 0 | Prior to Rev 19.4 this parameter specified the maximum number of file units guaranteed to be available to each user. Because the default number of file units available per user is $77772_8$ (decimal 32762), it is assumed there will be sufficient file units so that no units need be guaranteed. This parameter must be present, but its value is not used. |

*max-unit* Maximum number of units any one user may have open at one time. The default is $77772_8$ (decimal 32762).

If the FILUNT directive is not specified in the configuration file, the default value is used.

You should omit this directive from the configuration file and use the default number of file units.

### ▶ MAXPAG *number-of-pages*

Specifies the number of pages of physical memory (starting from page 0) to be used after cold start.

| *Parameter* | *Minimum* | *Maximum* | *Default* | *Recommended* |
|---|---|---|---|---|
| *number-of-pages*, octal | 400 | 100000 | all mem | Omit the directive |

*number-of-pages* is the highest physical page number that will be used, starting from physical address 0. After cold start, PRIMOS uses the first *number-of-pages* of physical memory (that is, from page 0 through *number-of-pages* minus 1). (One page is 2048 decimal bytes of memory; 512 decimal pages is 1 megabyte of memory.)

The value of *number-of-pages* must be between $400_8$ (256 decimal) and $100000_8$ (32786 decimal). If *number-of-pages* is not specified or if MAXPAG is not in the configuration file, all available memory is used.

The MAXPAG directive is intended primarily for performance testing. It allows a system with a large memory to function as if it had a smaller memory. It may also be used to allow PRIMOS to run successfully on a system that has some bad memory in higher addresses, by making that memory unavailable. Normally, you should not use MAXPAG.

Use the following table as a reference for setting MAXPAG.

| *Memory (MBytes)* | *Memory (pages)* | *MAXPAG argument (octal)* | *Memory (MBytes)* | *Memory (pages)* | *MAXPAG argument (octal)* |
|---|---|---|---|---|---|
| 1 | 512 | 1000 | 14 | 7168 | 16000 |
| 2 | 1024 | 2000 | 16 | 8192 | 20000 |
| 3 | 1536 | 3000 | 20 | 10240 | 24000 |
| 4 | 2048 | 4000 | 24 | 12288 | 30000 |
| 6 | 3072 | 6000 | 28 | 14336 | 34000 |
| 8 | 4096 | 10000 | 32 | 16384 | 40000 |
| 10 | 5120 | 12000 | 64 | 32768 | 100000 |
| 12 | 6144 | 14000 | | | |

If your system contains an arrangement of memory boards that produces holes in physical memory (rather than providing a contiguous block of memory), set MAXPAG as if these

holes contained actual memory. For example, specify `MAXPAG 4000` (which is 4 MB of memory) if your system has 3.5 MB of memory with a .5 MB hole in the middle.

If you specify a value for MAXPAG that results in the system using less than the total amount of available physical memory, the following message is displayed at the supervisor terminal:

```
System NOT configured with maximum possible memory:
only using mK BYTES, when nK BYTES are available.
```

The message is only a warning, and the MAXPAG directive is obeyed. If you receive this message and you want to use all your available memory, either increase the setting of MAXPAG or remove the directive from the configuration file.

### ▶ PREPAG *pages*

When a page fault occurs and there are no unused memory pages, PRIMOS makes available the three pages that were least recently used by writing them out to disk from memory. To change the default number of pages that are written out, use the PREPAG directive. The argument, *pages*, cannot be less than one or more than the number of pages available for paging. Unless your Prime System Analyst recommends changing the default, omit this directive from the configuration file.

### ▶ RWLOCK *code*

Specifies the system-default setting of the file system read/write lock.

The default value for *code* is 1. The permitted values and their meanings are:

| | |
|---|---|
| 0 | 1 reader *or* 1 writer, with the writer having exclusive control. |
| 1 | N readers *or* 1 writer, with the writer having exclusive control. |
| 3 | N readers *and* 1 writer. |
| 5 | N readers *and* N writers. |

You should omit this directive from the configuration file, because many subsystems and utilities will not work if you change the read/write lock default setting. To change the read/write lock of a file, or a set of files, use the PRIMOS command RWLOCK instead of this directive.

### ▶ VPSD

The VPSD directive wires the Prime assembly language debugger (also known as the kernel debugger) into PRIMOS at system startup. This debugger, formerly used for debugging the

operating system, does not specify any useful system configuration in a production environment.

It is not expected that a typical user environment will need the VPSD directive.

## OBSOLETE CONFIGURATION DIRECTIVES

The following configuration directives have been replaced with other commands or directives or are simply ignored.

### ▶ ALTDEV *pdev* [*records*]

Specifies the alternate paging partition and, optionally, its size.

At Rev. 21.0 the PAGING directive replaced ALTDEV.

*pdev* The physical device number of the paging partition.

*records* A 16-bit nonzero unsigned integer that indicates the number of records to be used for paging (out of the total number of records available for paging on the partition). The argument is ignored if it is zero, a negative number, or larger than the number of available records. If the argument is ignored (or if *records* is not specified), the entire available space for paging is used.

### ▶ AMLBUF *line* [*in-buff-size* [*out-buff-size* [*dmq-size*]]]

Sets the size of an asynchronous line's I/O buffers.

The AMLBUF directive is obsolete at Rev. 22.0. Use the CAB command, described in the *System Administrator's Guide, Volume II: Communication Lines and Controllers.*

### ▶ NTSABF *line in-buff-size out-buff-size xoff-lag xon-lag*

Sets the size of I/O buffers and flow control thresholds for remote NTS assignable lines connected to the local system.

The NTSABF directive is obsolete at Rev. 22.0, but may be useful for setting flow control thresholds. The CAB command replaces NTSABF; both are described in the *System Administrator's Guide, Volume II: Communication Lines and Controllers.*

▶ **NTSBUF** ***line in-buff-size out-buff-size xoff-lag xon-lag***

Sets the size of I/O buffers and flow control thresholds for remote NTS terminal users connected to the local system.

The NTSBUF directive is obsolete at Rev. 22.0, but may be useful for setting flow control thresholds. The CAB command replaces NTSBUF; both are described in the *System Administrator's Guide, Volume II: Communication Lines and Controllers.*

▶ **REMBUF** ***in-buff-size out-buff-size***

Sets the sizes of the terminal input and output buffers for remote users.

The REMBUF directive is obsolete at Rev. 22.0. Use the CAB command, described in the *System Administrator's Guide, Volume II: Communication Lines and Controllers.*

▶ **PAGDEV** ***pdev*** **[*records*]**

Specifies the paging partition and, optionally, its size.

At Rev. 21.0 the PAGING directive replaced PAGDEV.

Up to 230,000 records can be used for paging. The values and meanings of the arguments are as follows:

*pdev* The physical device number of the paging partition.

*records* A 16-bit nonzero unsigned integer that indicates the number of records to be used for paging (out of the total number of records available for paging on the partition). The argument is ignored if it is zero, a negative number, or larger than the number of available records. If the argument is ignored (or if *records* is not specified), the entire available space for paging is used.

If all available paging space is used, any attempt by a user process to obtain more memory causes the error condition PAGING_DEVICE_FULL$ to be signaled for that user.

▶ **PRATIO** ***n***

Specifies the ratio of use of the alternate paging partition in relation to the primary paging partition.

At Rev. 21.0 the PRATIO operator command replaced the PRATIO directive.

The PRATIO directive is ignored.

## ▶ TPDUMP [YES / NO]

The TPDUMP directive is obsolete at Rev. 22.0, and if present is ignored.

TPDUMP previously controlled the ability to take a tape dump before completion of a forced shutdown. The directive is no longer necessary, and a system is always ready for a tape dump following a forced shutdown.

**Notes**

You no longer need to follow the tape dump with a WARMSTART.

Users of ROAM-based products should not use WARMSTART.

The CRASH_AUDIT facility no longer requires the TPDUMP directive.

# INDEX

# INDEX

## D

## E

## F



## G



## H



## I



## K



## L

## Q



## R



## S

## T



## U



## V



## W