Prime™

System Administrator's Guide

Revision 20.2

DOC5037-4LA

System Administrator's Guide

Fourth Edition

by

Bernard Gilman

and

Lois Anne Conrad

This guide documents the software operation of the Prime Computer and its supporting systems and utilities as implemented at Master Disk Revision Level 20.2 (Rev. 20.2).

Prime Computer, Inc.
Prime Park
Natick, Massachusetts 01760

COPYRIGHT INFORMATION

CREDITS

Editor — Kathe Rhoades

Project Support — Jacki Forbes

Design and Graphic Support — Therese Bacharz

Production Support — Judy Gordon

PRINTING HISTORY -- System Administrator's Guide

Edition	Date	Number	Software Release
First Edition	July 1982	DOC5037-190	19.0
Update 1	January 1983	UPD5037-191	19.1
Update 2	June 1983	UPD5037-192	19.2
Update 3	February 1984	UPD5037-193	19.3
Second Edition	February 1985	DOC5037-2LA	19.4
Third Edition	January 1986	DOC5037-3LA	20.0
Fourth Edition	August 1986	DOC5037-4LA	20.2

In document numbers, L indicates loose-leaf.

CUSTOMER SUPPORT CENTER

Prime provides the following toll-free numbers for customers in the United States needing service:

1-800-322-2838 (within Massachusetts)	1-800-541-8888 (within Alaska)
1-800-343-2320 (within other states)	1-800-651-1313 (within Hawaii)

HOW TO ORDER TECHNICAL DOCUMENTS

Follow the instructions below to obtain a catalog, price list, and information on placing orders.

United States Customers

Call Prime Telemarketing,
toll free, at 1-800-343-2533,
Monday through Friday,
8:30 a.m. to 5:00 p.m. (EST).

International

Contact your local Prime
subsidiary or distributor.

This warning applies to the 9950, the 2250, and to all 50 Series processors manufactured after October 1, 1983:

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 warning applies to all other processors described in this book:

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. As temporarily permitted by regulation, it has not been tested for compliance with the limits for Class A computing devices persuant to Subpart J of Part 15 of FCC rules, which are designed to provide reasonable protection against such interference. 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.

Contents

APPENDIXES

About This Book

The System Administrator's Guide is intended to help a System Administrator make decisions and set up procedures that

- Enable a System Administrator to plan and configure a Prime computer system and users' environments
- Provide users with a smoothly functioning system
- Enable operators to deal with the day-to-day running of the system
- Help users and operators deal with unexpected problems

If you have any administrative responsibility for a Prime system, this book is intended for you. However, you are expected to have some familiarity with Prime systems before reading this book. If you are not familiar with the PRIMOS® operating system, read the Prime User's Guide (DOC4130-4LA) and its Rev. 20 update package (UPD4130-41A). These documents explain Prime's file management system and describe essential commands and utilities.

RELATED DOCUMENTATION

Other Prime documention that will be of help to you includes the following:

- Operator's System Overview (DOC9298-1LA) and its Rev. 20 update package (UPD9298-11A). This book introduces the series of operator's guides and describes computer-room operation of Prime systems.

- Operator's Guide to System Monitoring (DOC9299-2LA). This book describes how to monitor system activity and respond to system and user messages.

- Operator's Guide to File System Maintenance (DOC9300-2LA). 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 System Backups (DOC9301-1LA) and its update packages for for Rev. 20 (UPD9301-11A and UPD9301-12A) and Rev. 20.2 (UPD9301-13A). This book describes how to save information on disk or tape and how to restore that information later.

- Operator's Guide to the Batch Subsystem (DOC9302-2LA). This book describes how to set up, monitor, and control the Batch subsystem.

- Operator's Guide to System Commands (DOC9304-2LA). This book details many of the operator commands.

- Operator's Guide to the Spooler Subsystem (DOC9303-1LA). This book describes how to set up, monitor, and control the Spooler subsystem.

- PRIMOS Commands Reference Guide (DOC3108-5LA) and its update packages for Rev. 20 (UPD3108-51A) and Rev. 20.2 (UPD3108-52A). This book is a detailed reference of user commands.

- Network Planning and Administration Guide (DOC7532-2LA) and PRIMENET Guide (DOC3710-193LA), as well as PRIMENET™ Guide update packages for Rev. 19.4 (UPD3710-31A), Rev. 20 (UPD3710-32A), and Rev. 20.2 (UPD3710-33A).

- ICS User's Guide (DOC10094-1LA). This book provides detailed information on Prime's Model 2 (ICS2) and Model 3 (ICS3) Intelligent Communications Subsystems.

- Your CPU handbook, for example the Prime 2550 Handbook (DOC10073-2LA) and the Prime 9955 Handbook (DOC8887-2LA).

USING THIS BOOK

This book is organized into three major parts that are designed to help you plan, set up, and maintain your system. You should read Parts I and II before actually creating your system, and Part III before your system is completely operational. Chapter 1 in Part I summarizes the Rev. 20.2 features and changes that may be of interest to the System Administrator. It also contains an overview of all PRIMOS directories that are referred to in this guide.

Part I, PLANNING THE SYSTEM, contains the following information on planning for your system:

- Planning the configuration of PRIMOS by use of configuration directives.

- Deciding what aspects of the User Profile system to use and planning your User Profile Data Base. Access Control Lists (ACLs) are an important aspect of users' profiles.

- Dividing your disk space among your system needs (especially for paging) and users. When you use quotas on top-level user directories, you limit the amount of disk space each directory can use. Without quotas, users compete for space on a first-come, first-served basis.

- Coordinating security for logins and access of data.

- Planning Spool and Batch subsystems.

- Allocating hardware and software resources.

Part II, CREATING THE SYSTEM, explains how to get the system up and running. Chapters contain information on the following topics: installation, setting access to the software, details of configuration directives, configuring asynchronous lines, and adding users.

Part III, MAINTAINING THE SYSTEM, describes how to perform the following tasks to keep the system running smoothly: maintaining the environment, backing up the system, helping users with problems, adding and modifying software, and monitoring the system.

Appendix A explains how to write external login and logout programs. Appendix B lists error and program messages that the PRIMOS preloader and the PRIMOS initialization sequences can generate. Appendix C lists error and information messages that EDIT_PROFILE displays. Appendix D lists obsolete and rarely used commands and directives. Appendix E describes the procedure for tracing asynchronous lines back to the controller.

PRIME DOCUMENTATION CONVENTIONS

Command formats, statement formats, and examples throughout this guide use the following conventions. Examples illustrate the uses of these commands and statements in typical applications. You can enter terminal input in either uppercase or lowercase.

Convention	Explanation	Example
UPPERCASE	In command formats, words in uppercase indicate the actual names of commands, statements, and keywords.	SLIST
lowercase	In command formats, words in lowercase indicate items for which the user must substitute a suitable value.	LOGIN user-id

Underlining
in
formats
	If a command or statement has an abbreviation, it is indicated by underlining. When the command itself contains an underscore, the abbreviation is shown below the full name, and the name and abbreviation are placed within braces.	LOGOUT
		
		{ SET_QUOTA }
		{ SQ }

Underlining in examples	In examples, user input is underlined but system prompts and output are not.	OK, RESUME MY_PROG This is the output of MY_PROG.CPL OK,
Brackets [ ]	Brackets enclose a list of two or more optional items. Choose none or one or more items.	USAGE [-ALL] [-BRIEF]

Braces
{ }
	Braces enclose a list of items. Choose only one.	CLOSE { filename }
		{ ALL }

Ellipsis ...	An ellipsis indicates that the preceding item may be repeated.	item-x[,item-y]...
Parentheses ( )	In command or statement formats, parentheses must be entered exactly as shown.	DIM array (row,col)
Hyphen -	Wherever a hyphen appears as the first character of an option, it is a required part of that option.	SPOOL -LIST
Apostrophe '	An apostrophe preceding a number indicates that the number is in octal.	'200

PART I

Planning the System

1 Overview

This chapter describes the following:

- Responsibilities of the System Administrator
- Directories that comprise PRIMOS
- New features of Rev. 20.2

ROLE OF THE SYSTEM ADMINISTRATOR

System administration is the organization and management of computer systems. The System Administrator has the following responsibilities:

- Plans and sets up the system, including the environments of the system's users
- Sets the policy for the use of the system
- Makes the system secure

Planning is particularly important for the administrator, because good planning makes day-to-day operations run more smoothly.

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, in your installation several people may share the job of administering the system. The System Administrator may even double as an operator or a user. As the System Administrator, you may be asked to perform one or more of the following tasks:

- Create or revise a system configuration file
- Allocate disk space and software resources
- Determine the attributes of individual users who must have a user profile before they can log in to the system
- Set access rights on MFD's, system directories, and top-level user directories to make the system secure
- Set up the Spooler and the Batch subsystems
- Configure asynchronous lines
- Set schedules for and perform backups
- Maintain system hardware and software
- 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 separately chargeable products.

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

The Master File Directory (MFD) on the command device holds all non-chargeable software (top-level UFDs), plus the following four important files.

The BOOT File: This file contains the bootstrapping procedure for the system, and it is used with every new boot, or startup, of the system.

The BOOT_RUN_FILE_TREENAME File: This file contains the boot runfile pathname for booting PRIMOS from disk.

The BADSPT File: 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. This file exists only on partitions that have badspots. Whenever data is written on a disk, PRIMOS scans the BADSPT file to ensure that no information is copied onto unusable records.

The DSKRAT File: This file is the Disk Record Availability Table, which contains a list of available records on the partition. This table is dynamic; that is, it changes constantly as the partition's records are used or freed. A new DSKRAT file is automatically created every time a partition is made. The PRIMOS disk repair command (FIX_DISK) and the PRIMOS file system use the DSKRAT file. The DSKRAT file takes the name of the partition on which it resides.

Required Directories

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

The Directory CMDNC0: The UFD 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. Any commands that do not appear in CMDNC0 (such as ATTACH, RDY, and LOGOUT) are internal PRIMOS commands.

CMDNC0 also contains the configuration file (CONFIG) and the system PRIMOS.COMI startup file.

The Directory LIB: The UFD LIB contains all static-mode libraries available on the system.

The Directory LIBRARIES*: The UFD LIBRARIES* contains all library EPFs.

The Directory PRIRUN: The UFD 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.

The Directory SAD: The UFD 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 EDIT_PROFILE to create the SAD.

The Directory SEARCH_RULES*: The UFD SEARCH_RULES*, which the installation program SYSTEM>INSTALL.STD.COMI automatically creates, holds the default system search rules file, ENTRY$.SR, and the ADMIN$.ENTRY$.SR file.

The Directory SYSTEM: The UFD SYSTEM contains all shared subsystem software, such as FORMS, and compilers for high level languages such as COBOL. SYSTEM contains the LOGIN_SERVER.COMI and SET_LSR_ACLS.CPL files after you install them. The latter file sets ACLs on each directory in the System Administration Directory to grant LUR access to the Login Server. The Login Server is a new feature of Rev. 20.2. This directory also contains the programs INSTALL.STD.COMI, which installs PRIMOS, and INSTALL.ALL.COMI, which installs chargeable software.

Other Important Directories

The following directories may also be under your control, but are not required to run PRIMOS:

Directory	Description
BACKUP*	Contains the files that comprise the BRMS utility, which is described in the Operator's Guide to System Backups.
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 must be installed from this directory into the MFD as BOOT.
DIAG	Contains the files that comprise diagnostic tools that Customer Service Representatives use.
DOS	Contains the obsolete single-user operating system, PRIMOS II, in the file DOS.SAVE.
DOWN_LINE_LOAD*	Contains the files that are loaded into the Intelligent Communication Subsystem (ICS) controllers when the system is booted.

Directory	Description
INFO20.2	Contains the files that summarize the major changes in the current Revision.
HELP*	Contains HELP files for PRIMOS commands.
LOGREC*	Contains system event log files.
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).
SERVERS*	Contains the files LOGIN_SERVER.RUN and LOGIN_SERVER.ENTRY$.SR, which are required to run the Login Server, a new feature of Rev. 20.2.
SPOOLBIN	Contains a file that enables a System Administrator to specify a password to SPOOLQ, recompile the module, and reload the Spooler subsystem from binary files that are also in this directory.
SPOOLQ	Contains the files that control the environments of printer operations. This includes files that monitor the spooler and determine user privileges. This directory also contains copies of files submitted to be printed as PRTxxx files.
SYSCOM	Contains parameter insert files for compilers.
SYSOVL	Contains files required by COBOL and the data files used by the FORTRAN 77, PASCAL, and PL/I Subset G compiler default driver programs.
T&MRUN	Contains test and maintenance programs used by Customer Service Representatives.
TOOLS	Contains files and programs that can perform such tasks as converting Rev. 20.0 to Rev. 20.2. This directory also contains the driver programs for the PL/I Subset G, Pascal, and FORTRAN 77 compilers.

Other Optional Directories

You may receive the following directories with PRIMOS if they were ordered as separate products:

Directory — Description

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.

PRIMENET* — Contains all files needed to run networks and the network event log files.

REV. 20.2 FEATURES

This section summarizes the Rev. 20.2 features and changes that are of interest to the System Administrator. For details on all aspects of Rev. 20.2, see the Software Release Document, Rev. 20.2. For details on Rev. 20.2 PRIMOS commands, see the Rev. 20.2 update to the PRIMOS Commands Reference Guide.

The online Rev. 20.2 INFO files reside in the INFO20.2 directory. These files contain new information on PRIMOS and on other Prime software.

Login Server

The Login Server is a system server that handles all terminal login attempts for local and remote users that were previously handled within PRIMOS. Unlike other system servers, the Login Server does not require server support functions and does not require you or a user to intervene. The Login Server is brought in when the system is booted. If, for some reason, the Login Server does not start or stops after it is started, you can enter the new START_LSR system command at the supervisor terminal to start it.

The user interface that the Login Server provides to PRIMOS at login or logout is the same as PRIMOS provided prior to Rev. 20.2. The Login Server provides the following services for a system:

- Receives input from all terminal lines on which a user can log in
- Responds to all commands, such as the DATE command, that can be issued without logging in
- Processes and validates login requests
- Passes remote login requests to the appropriate remote node
- Arranges to associate a user who has logged in successfully with a terminal process and also obtains terminal buffers for that user
- Reassumes control of a line that becomes inactive when a user logs out

The Login Server has no interaction with users who are logged in and have had their lines associated with a process or other server except when a user who is logged in attempts to log in again. When this happens, the Login Server is called upon to read the SAD and to validate and initialize the user. Users cannot explicitly call the Login Server.

The Login Server accepts the following commands from a logged-out line:

Command	Description
DATE	Displays the current calendar date and clock time.
DELAY	Defines a time function that delays printing a character after a carriage return (RETURN) has been output to the terminal.
DROPDTR	Drops the DTR (Data Terminal Ready) signal associated with a terminal line.
LOGIN	Admits a user onto the system.
USRASR	Allows the system console to function as a user terminal.

For a detailed description of these commands, refer to the PRIMOS Commands Reference Guide.

As the System Administrator, you need to know the following about the Login Server:

- You must configure (with the NPUSR directive described in Chapter 10, CONFIGURATION DIRECTIVES) a phantom for the Login Server.

- When you execute the STATUS USERS command, the Login Server is listed. The Login Server runs under the name LOGIN_SERVER and its process type is listed as LSr.

- The LOGIN_SERVER.RUN file and the LOGIN_SERVER.ENTRY$.SR file are installed in the SERVERS* directory, a new Prime-supplied system directory that requires $REST:LUR access rights. See Chapter 9, SETTING SYSTEM ACCESS.

- At cold start, ACLs on each directory in the SAD are set automatically to allow the Login Server to access them. Do not remove these ACLs, which are preserved by EDIT_PROFILE. Also, you do not have to create an entry for the Login Server in the SAD.

- Executing the new START_LSR supervisor terminal command starts the Login Server, and executing the new STOP_LSR supervisor terminal command (described in the Rev. 20.2 update to the Operator's Guide to System Commands) stops the Login Server. If you enter the START_LSR command while the Login Server is running, the supervisor terminal displays a message that the Login Server cannot be spawned. You cannot stop the Login Server with the LOGOUT -n command.

- If the Login Server stops, it sends the following message to all logged-out terminals:

  ```
  Logins are blocked -- Login Server is logged out. (lsr)
  ```

  If the STOP_LSR command shuts down the Login Server, the supervisor terminal also displays a phantom logout message. The Login Server logs out under the user ID under which it was running. If an internally detected error causes the Login Server to stop, the supervisor terminal displays an error message. Users who try to log in while the Login Server is down receive no messages at their terminals.

- If the Login Server logs out abnormally after it is started and users cannot log in, a search rules problem may be indicated. Check that all entries in the SEARCH_RULES*>ENTRY$.SR file are on the command device, that all pathnames are correct, and that the ENTRY$.SR file contains no typographical errors. From the supervisor terminal, fix the search rules and start the Login Server with the START_LSR command.

Search Rules

The ENTRY$.SR default system file is now in a new directory called SEARCH_RULES*. The installation program SYSTEM>INSTALL.STD.COMI automatically creates the new directory and copies SYSTEM>ENTRY$.SR into it. You can delete the copy of ENTRY$.SR that is in the SYSTEM directory after you check that ENTRY$.SR is in the SEARCH_RULES* directory.

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

```
-PRIMOS_DIRECT_ENTRIES
```

Because the installation program performs these actions, you do not have to do anything to ensure that PRIMOS search rules function properly. You must only be aware that the default ENTRY$.SR file is now in SEARCH_RULES* if you, an operator, or a user decide to modify that file. Do not change the ADMIN$.ENTRY$.SR file, however.

When you type the LIST_SEARCH_RULES command, the first rule in the ENTRY$ search list is -PRIMOS_DIRECT_ENTRIES. The LIST_SEARCH_RULES command now pauses and displays the --More-- prompt after each screen of text. The LIST_SEARCH_RULES command option -NO_WAIT disables pausing between pages.

EDIT_PROFILE

You can use EDIT_PROFILE to define the attributes of User 1 (SYSTEM). At cold start, PRIMOS reads profile information in the SAD and uses it to initialize attributes of the supervisor terminal. The user name for the supervisor terminal is SYSTEM and the project with which SYSTEM is affiliated is the project specified in the profile.

Making an entry for user SYSTEM in the SAD enables you to set the Initial Attach Point and assign specified values for command environment attributes, including the number of dynamic and static segments for the user SYSTEM. If you do not make an entry in the SAD for User 1, PRIMOS uses the system defaults for the profile. If the SAD does not exist or an entry for SYSTEM is not present or the project is invalid, the following messages are displayed at startup time:

```
Can't attach to the SAD:  Not found.  (nlogin)

Profile data cannot be initialized from the SAD for the supervisor.
System defaults are being used.
```

If you have entry an in the SAD for SYSTEM that you are using for administrative purposes, PRIMOS uses that entry to initialize profile data for User 1. You should create another, different identifier to replace the SYSTEM identifier. The SYSTEM identifier is reserved for User 1.

If the SAD contains a user profile for SYSTEM, the following message is displayed at startup time:

```
Initializing profile data for the supervisor from the SAD.
```

The Initial Attach Point for User 1 is set when the SAD is read, which is before the PRIMOS.COMI file is processed. If PRIMOS.COMI contains an attach point other than CMDNC0, the supervisor terminal will be left at that attach point unless there is an ATTACH CMDNC0 command at the end of the PRIMOS.COMI file. If the Initial Attach Point is not CMDNC0, programs in the PRIMOS.COMI file that must be run from CMDNC0 should include either an ATTACH command to CMDNC0 or pathnames with CMDNC0 in the pathname.

Handling Asynchronous Lines

The SET_ASYNC command, new at Rev. 20.2, is a decimal-based replacement for the AMLC command.

The CONVERT_AMLC_COMMANDS, also new at Rev. 20.2, is a utility that translates AMLC commands to the equivalent SET_ASYNC commands.

For details on CONVERT_AMLC_COMMANDS and SET_ASYNC, see Chapter 11, CONFIGURING ASYNCHRONOUS LINES.

Changes to the DISLOG Directive

Prior to Rev.20.2, the DISLOG directive, which automatically logged out a user whose line was disconnected, applied to all asynchronous lines even though only a few of these lines must be logged out when they lose carrier. For example, random noise on asynchronous lines can induce a drop in the carrier signal for terminals that are directly connected. These users can be randomly and unintentionally logged out.

You can now configure DISLOG for each asynchronous line. The directive DISLOG line_number enables DISLOG on the selected line number. Specify one DISLOG directive for every line that you want to log out when that line is disconnected.

It is important to note that one DISLOG directive that follows another cannot override that first directive. For example, a configuration file with a DISLOG YES directive followed by DISLOG NO causes every line to be set for DISLOG. Also, a DISLOG directive that sets all lines takes precedence over a directive that sets a single line. For example, when the DISLOG command in a CONFIG file has several per-line DISLOG directives and a DISLOG YES directive, DISLOG YES takes precedence and is enabled on every line.

2 Planning the System Configuration

The PRIMOS operating system contains code that manages the following:

- Access for up to 255 processes
- Segmented virtual address space for programs up to 32 megabytes per user
- Input/output control
- File system
- Interactive terminal users and phantom user non-interactive jobs
- Communications systems

In addition, utilities (such as BIND) and languages (such as FORTRAN) are brought into user memory as needed.

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 Chapter 17, SYSTEM MONITORING.

THE SYSTEM CONFIGURATION FILE

The system configuration file is composed of a series of configuration directives, one per line. The file, which is usually named CONFIG, must be stored in the CMDNC0 directory. The file must contain the COMDEV, PAGDEV, NTUSR, NPUSR, and GO directives.

After you have brought up PRIMOS, you can modify the configuration file with a text editor such as ED or EMACS. The next time you cold start the system, the modified file is used for configuration and the modifications take effect.

If you are using Rev. 20 disks on your system, you cannot create or modify a configuration file under PRIMOS II because PRIMOS II cannot write on Rev. 20 disks. See Chapter 10, CONFIGURATION DIRECTIVES, for directions on how to start up PRIMOS without a configuration file.

TYPES OF CONFIGURATION DIRECTIVES

Configuration directives can be grouped into the following five general categories:

- 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 D, 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.

The sections that follow describe the configuration directives. For full details, see Chapter 10, CONFIGURATION DIRECTIVES.

NECESSARY DIRECTIVES

Necessary directives set the command device (COMDEV), paging partition (PAGDEV), number of users (NTUSR), and the end of the configuration file (GO).

Command Device

The COMDEV directive specifies which partition is the command device. The command device is the partition on which CMDNCO resides, and is, therefore, the partition that is searched when a user invokes an external PRIMOS command.

The argument to COMDEV is the physical device number of the partition that is initially assigned as logical device 0. (This partition is listed first in the output from the STATUS DISKS command.) See the Operator's Guide to File System Maintenance for details on constructing physical device numbers.

Split command disks are not recommended because they incur excessive use, thus unbalancing the system workload. However, on smaller systems (Prime 2350™ and 2450™), split command disks may be necessary. (A split command disk is a partition that contains both the CMDNCO directory and paging space.)

Primary Paging Partition

Each system must have one, possibly two, partitions reserved for paging. (The paging partition is also referred to as the paging device or the paging disk.) If your system has two paging partitions, one is called the primary paging partition and the other is called the alternate paging partition. (For information on alternate paging partitions, see the section below, Specifying an Alternate Paging Partition.)

The PAGDEV directive specifies which partition is the primary paging partition. This directive must be included in the configuration file.

A paging partition can be a split disk (that is, it also contains storage space for user files). See Chapter 4, DISKS AND TAPE DRIVES, for further details on paging partitions, split disks, and determining the size of paging partitions.

Number of Users

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

- Terminal users (NTUSR)
- Phantom users (NPUSR)
- Remote users (NRUSR)
- Slave users (NSLUSR)

Remote and slave users are for PRIMENET only. The total number of configured users of all types must be less than or equal to 255.

You can issue a STATUS USERS command to display information about each user currently on your system. At Rev. 20.2, the Login Server, which is configured as a phantom user (described below), is listed as a user.

Terminal Users: The NTUSR directive sets the number of terminal users. You can configure up to 255 ('377) terminal users (this 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. However, it also increases the size of memory required for PRIMOS (wired memory), causes more paging, and degrades system performance in direct relation to the number of excess terminal users configured.

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 1, which is its default value. If you set NPUSR to 0, PRIMOS displays the following messages:

```
Warning: "NPUSR 0" in configuration file. Login Server cannot be
run unless NPUSR is >0.

No logins are possible on current configuration.
```

You must configure phantoms for each of the following:

- The Login Server, a new feature of Rev. 20.2, which is described in Chapters 1 and 5
- Each printer (1 per spooler)
- The Batch monitor
- Batch queues (a maximum of 1 per queue)
- The PRIMENET server NETMAN (if your system has PRIMENET) and the route-through server RT_SERVER (if your system is a gateway node for PRIMENET)
- Other communications products (DPTX, RJE, FTR, PRIME/SNA™)
- The PRIMIX™ process manager (if your system has PRIMIX)

You should 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.

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 up to 63 ('77) 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.

The maximum number of slave users you can configure depends on whether your system is using a Route-through server (for PRIMENET) or the File Transfer Service (FTS).

- If you are using both FTS and Route-through, you can specify up to 59 ('73) slave users.
- If you are using either FTS or Route-through (but not both), you can specify up to 61 ('75) slave users.
- If you are using neither FTS nor Route-through, you can specify up to 63 ('77) slave users.

Consult with your Prime System Analyst to set 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.

Note

> Although you may configure a maximum of 63 remote users and 63 slave users, the total number of both remote and slave users that can be active at any one time cannot exceed 63 ('77). Attempts to exceed that limit produce error messages saying that the resource is temporarily unavailable.

End of File

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

USEFUL DIRECTIVES

Useful directives set parameters for the alternate paging partition, utilization of paging partitions, utilization of memory, assignable asynchronous lines, buffers, and event logging.

Specifying an Alternate Paging Partition

To specify an alternate paging partition, use the ALTDEV directive. The alternate paging partition may be located on any disk controller.

If you expect a high amount of paging on your system, you may want to use an alternate paging partition on another disk drive to equalize disk I/O. To set the ratio of use of the paging partitions, see the next section, Utilization of the Paging Partitions.

Utilization of the Paging Partitions

On a system with two paging partitions, the primary paging partition should ideally be on a disk drive that is not used frequently (other than for paging). For example, the primary paging partition should not be on the same drive as user files that are frequently accessed.

If the two paging partitions are on separate disk drives, performance may be improved because PRIMOS, by default (that is, if the PRATIO directive is not used), evenly balances its use of the two paging partitions. However, if the amount of non-paging activity on one disk drive is higher than on the other, the System Administrator can balance the overall activity of the drives by using the PRATIO directive to increase the use of the less active drive for paging.

The PRATIO directive determines how often the alternate paging partition is used in relation to the primary paging partition. This ratio is approximately n times out of 10. (The default of n is 5, which means the alternate paging partition is used approximately half the time.)

To use the alternate paging partition more often, set n to a higher number (up to '12). Setting n to a lower number causes the alternate paging partition to be used less often.

If n is set to 0, the alternate paging partition is not used until the primary paging partition is full. Conversely, if n is set to '12, the primary paging partition is not used until the alternate paging partition is full. If both paging partitions are full, any attempt by a user to acquire more virtual memory causes the PAGING_DEVICE_FULL$ error condition to be signalled.

Amount of Memory to Use

At cold start, PRIMOS validates all physical memory. The MAXPAG directive then specifies that the first n pages of physical memory (starting from physical page 0) will be used when PRIMOS is running. (n is the argument to MAXPAG.) Thus, after your system is running, it uses for physical memory the pages from physical page 0 through physical page number n minus 1. (One page equals 2048 bytes of memory.)

If the MAXPAG directive is not included in the configuration file, all available memory is used.

For example, assume that your system has 4096 decimal pages (8 megabytes) of physical memory. If you specify MAXPAG 4000, your system will use only the first 2048 pages (that is, physical page 0 through physical page 2047), which means you have only 4 megabytes of memory for use. To use all 4096 pages, you would have to either specify MAXPAG 10000 or omit MAXPAG from the configuration file.

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 printed 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 value for MAXPAG or omit MAXPAG from the configuration file.

Note

> If your system has an arrangement of memory boards that produces holes in physical memory rather than providing a contiguous block of memory, you must set MAXPAG as if these holes contained actual memory. For example, if your system has 3.5 megabytes of memory with a 0.5 megabyte hole in the middle, use MAXPAG with an argument of 4000 (which specifies 2048 pages, or 4 megabytes, of memory) so that all 3.5 megabytes of actual memory are used.

Assignable Asynchronous Lines

The NAMLC directive sets the number of buffers for assignable asynchronous lines. Assignable asynchronous lines are used by user programs or the spooler to communicate with serial devices such as serial printers. The default value of NAMLC is 0. The number of assignable asynchronous lines plus the number of terminal users cannot exceed 255 ('377).

To define a line as assignable, use the SET_ASYNC command with the -ASGN YES option. For details on the SET_ASYNC command, see Chapter 11, CONFIGURING ASYNCHRONOUS LINES.

Assignable asynchronous lines often need to have their buffer sizes changed from the default values with the AMLBUF directive. This directive is described in the later section, Changing Buffer Sizes.

Event Logging

The two types of event logging mechanisms are system event logging and network event logging. Both are discussed in detail in Chapter 17, SYSTEM MONITORING, and in the Operator's Guide to System Monitoring.

The LOGREC directive enables or disables system event logging; the NETREC directive does the same for network event logging. The arguments and their effects are the same for both directives.

Event logging is enabled when the argument's value is 0 (the default). If the value is positive, event logging is enabled, but a message is printed at the supervisor terminal warning that the directive no longer sets a quota on the logging file (as it did prior to Rev. 19).

If the argument is a negative value (such as '177777), event logging is disabled.

Note

> By using the EVENT_LOG command, you can enable or disable event logging while PRIMOS is running. However, if the command device is write-protected, you should disable event logging with the LOGREC and NETREC directives rather than with the EVENT_LOG command. The reason is that the event-logging file is on the command device. If event logging is enabled at cold start, the event loggers try to write the cold-start event (that is, the fact that PRIMOS started up) to the event-logging file, thus causing an error.

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 operation, it provides an indication of the memory used for a particular system configuration.

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 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).

By using the NLBUF directive, you can allocate from 8 ('10) to 256 ('400) LOCATE buffers. Allocating more LOCATE buffers decreases disk I/O. However, additional LOCATE buffers 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 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. More buffers should be configured if the USAGE command reports a LOCATE miss rate of greater than 10% (in the "%Miss" field) and a paging rate of less than 5 page faults per second (in the "PF/S" field). For more information on LOCATE buffers, see the Operator's Guide to System Monitoring.

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). You may want to increase this number if users frequently receive such messages as "Not enough segments" or "No space available from process class storage heap." You can specify a maximum of 1024 ('2000) 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 processer.

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 only typed 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 the Prime-supplied defaults.

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 Prime 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 caused 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 10, CONFIGURATION DIRECTIVES, for the conditions that must be met for the user process to be logged out.

MEMHLT NO is recommended, but 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 Chapter 13, EQUIPMENT AND ENVIRONMENT.

> 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 procedure. 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.

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 specifying 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: By default, users are not logged out if their terminal lines are disconnected or if their terminals are turned off. To retain this default setting, either include DISLOG NO in the configuration file or else omit the directive.

If you want all users to be logged out when their terminals or terminal lines are disconnected, specify DISLOG YES in the configuration file.

If you want selected users to be logged out when their terminals or terminal lines are disconnected, specify DISLOG line_number in the configuration file.

It is recommended that you use DISLOG YES or DISLOG line_number if you have lines configured for Auto Speed Detect (ASD) so that the 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) 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).

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.

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. For further information on determining buffer sizes, see Chapter 11, CONFIGURING ASYNCHRONOUS LINES.

Terminals and Assignable Lines: Local terminals and assignable asynchronous lines have three buffers: input, output, and DMQ (Direct Memory Queue). Their default sizes (in halfwords) are 128 ('200) for input buffers, 192 ('300) for output buffers, and 32 ('40) for DMQ buffers. The AMLBUF directive changes the default buffer sizes.

Most terminals use the default buffer sizes. Terminals for special purposes (such as OAS, FORMS, DPTX, or PRIME/SNA) may have performance improved by modifications to the buffer sizes. For example, a terminal used for FORMS might have its input buffer changed to 832 ('1500), its output buffer to 1024 ('2000), and its DMQ buffer to 128 ('200).

The required buffer sizes for assignable asynchronous lines vary with the specific device used. Your Prime System Analyst can help you decide which buffer sizes, if any, need to be changed.

For further details on buffers and the AMLBUF directive, see Chapter 11, CONFIGURING ASYNCHRONOUS LINES.

Remote Users: The REMBUF directive sets the input and output buffer sizes for all remote users of your system. (The DMQ buffer sizes for remote users are set by their local system.) For input buffers, 130 ('202) halfwords (260 decimal bytes) is both the default and the minimum size. For output buffers, 65 ('101) halfwords (130 decimal bytes) is both the default and the minimum size. Total size of all input buffers plus output buffers cannot exceed 768,000 ('2734000) halfwords (1536 decimal kilobytes).

Optimum settings for remote users' buffers depends upon the network configuration and what kinds of operations the users are performing.

To allow faster transfer of terminal data for block-mode devices, double the size of the input buffer to 520 bytes ('404 halfwords). You can significantly increase throughput by doubling the size of the output buffer to 516 bytes ('402 halfwords). This is especially true where users are logging in remotely across a ring network.

For remote login over a Public Data Network (PDN), you must set the sizes of the input and output buffers to the octal value of the packet size that you configured with the CONFIG_NET utility. (For information on CONFIG_NET, see the Network Planning and Administration Guide.

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) 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 Chapter 11, CONFIGURING ASYNCHRONOUS LINES.

ICS Controllers: The ICS INPQSZ directive changes the size of the input queues for ICS controllers from the default value of 63 ('77). 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 Chapter 11, CONFIGURING ASYNCHRONOUS LINES.

AMLC Programmable Clock

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

To specify the programmable clock speed for an AMLC line, use the AMLC command to set bits 8-10 of the configuration argument to 4 (bit pattern 100). If bits 8-10 are set to 4 and the AMLCLK directive is not in the configuration file, the line assumes the default speed of 9600 ('22600) baud. See Chapter 11, CONFIGURING ASYNCHRONOUS LINES, for details on the AMLC 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 octal value for one minute (which is 600 tenths of a second) is '1130 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 octal value '2260. To set gracetime to three minutes, use the octal value '3410.

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.

Changing the Baud Rate: The ASRATE directive sets the baud rate of the supervisor terminal. Most hard-copy supervisor terminals have a baud rate of 300, which is the default of ASRATE.

If you have a screen terminal (such as a PST 100™ or a PT200™) as your supervisor terminal, you may want to use one of the other available baud rates of 110, 1200, and 9600.

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

Changing Buffer Sizes: Most hard-copy supervisor terminals have an input buffer of 256 ('400) bytes and an output buffer of 384 ('600) bytes. If your supervisor terminal runs at greater than 300 baud, you can use the ASRBUF directive to increase the output buffer size.

Synchronous Lines

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

For details on the four SMLC directives, see Chapter 10, CONFIGURATION DIRECTIVES.

Note

> As of Rev. 20, SYNC is a synonym for SMLC. For example, specifying SYNC ON is the same as specifying SMLC ON.

Uninterruptible Power Supply

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

If your system has lines connected to an ICS controller, you may use the following three directives that configure ICS lines:

- ASYNC JUMPER, which selects the available line speeds
- ICS CARDS, which verifies an asynchronous Line Adapter Card configuration of an ICS2/3 controller
- ICS INTRPT, which sets the asynchronous interrupt rate

In addition, the ICS INPQSZ directive sets the size of the ICS input queue buffers. This directive is described in the section above, Changing Buffer Sizes.

Line Speed Selection: Lines connected to an ICS controller can run at the speeds shown in Table 2-1. Speeds are measured in bits per second (bps).

Table 2-1
Valid Speeds for ICS Lines

50	150	1200	4800
75	200	1800	7200
110	300	2400	9600
134.5	600	3600	19200

Lines can be set to these speeds in the following ways:

- 110, 134.5, 300, 1200 bps: These speeds are always available to ICS lines by using the AMLC or ASSIGN AMLC command with the following four configuration values:

Speed	Configuration
110	2013
134.5	2113
300	2213
1200	2313 (default)

- 9600 bps: This speed is available as the default rate for the programmable clock. If the AMLCLK directive is in the configuration file and does not specify 9600 bps, then the programmable clock cannot be used to set an ICS line to 9600 bps. Use the ASYNC JUMPER directive instead. (The directive AMLCLK 22600 specifies 9600 baud, because the value is specified in octal.)

  To set an ICS line to the speed defined by the programmable clock, use the AMLC or ASSIGN AMLC command with a configuration value of 2413. You may do this even if the AMLCLK directive specifies a speed other than 9600 bps. However, if the specified speed is not one of the valid speeds for an ICS line given in Table 2-1 above, any attempt to set an ICS line to the programmable clock speed produces an error message.

- Other speeds: To set an ICS line to a speed other than 110, 134.5, 300, 1200, or 9600, use the following procedure:

  1. Pick three valid ICS line speeds from Table 2-1. (Do not select 110, 134.5, 300, or 1200 because they are always available, as indicated previously.)

2. Use the ASYNC JUMPER directive to specify the three speeds at system start up. See Chapter 10, CONFIGURATION DIRECTIVES, for a detailed description of the ASYNC JUMPER directive.

3. For each ICS line that requires one of the selected speeds, issue an AMLC or ASSIGN AMLC command with the configuration value as follows:

Selected Speed	Configuration
First (speeda)	2513
Second (speedb)	2613
Third (speedc)	2713

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 Chapter 11, CONFIGURING ASYNCHRONOUS LINES.)

Verification of ICS2/3 Configuration: ICS2/3 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 ICS2/3 controller, that controller's configuration is not checked at cold start.

For further details on the ICS CARDS directive, see Chapter 10, CONFIGURATION DIRECTIVES, or the ICS User's Guide.

3
Planning the User Environment

When you plan your system, you must decide what kind of environment you will create for your users. This chapter discusses the topics that are essential for this task, including the following:

- User profiles, which define the attributes of an individual user
- The User Profile Data Base, which contains information on all users and projects on your system
- Access Control Lists, which provide security for directories and files

The chapter also shows you how you can design your data base, and gives examples of setting up different kinds of data bases.

USER AND PROJECT PROFILES

The System Administrator plans the User Profile Data Base (as explained later in this chapter) and then creates it using the EDIT_PROFILE utility (as explained in Chapter 12, USING EDIT_PROFILE). The Administrator also uses EDIT_PROFILE to keep the data base up-to-date.

Each user must have a user profile in the User Profile Data Base before that user can log in to the system.

The User Profile Data Base also contains profiles for one or more projects. A project is composed of a set of users who share certain characteristics or are accounted for together.

Each system must have at least one project. If you want only one project on your system, you can establish a system default project as follows:

1. Use EDIT_PROFILE to create a default project. The project is named DEFAULT.

2. Allow DEFAULT to remain the only project on the system. DEFAULT as the only project provides the following advantages:

   - EDIT_PROFILE automatically registers all users as members of project DEFAULT when you add them to the system.

   - EDIT_PROFILE asks no questions relating to projects.

   - Users never have to specify a project ID at login.

Projects can customize user environments. Because users can be members of more than one project, a user's attributes and environment can vary according to which project ID the user supplied at login. Thus, a user who has different needs (such as different directories and different access rights) for different jobs can meet these needs automatically by logging in as a member of a particular project.

The System Administrator can delegate a separate administrator (called a Project Administrator) for each project. The Project Administrator then assumes administrative responsibility for that project, within the limits the System Administrator set for that project. By appointing one or more Project Administrators, the System Administrator equitably delegates the responsibilities and the workload of administration among a number of people.

Advantages of User Profiles

The advantages of user profiles include the following:

- Provides a secure method of identifying and validating users

- Provides administrative control over users

- Provides an interface with the Access Control List mechanism for file system protection

- Allows the grouping of users with similar characteristics for purposes of accounting and file system control

- Allows the creation of a unique environment for each user

DEFINING USER PROFILES

A user profile consists of one set of system attributes and one set of attributes for each project to which the user belongs.

Supervisor Terminal Profile

At Rev. 20.2, you can use EDIT_PROFILE to define the attributes of User 1 (SYSTEM). At cold start, PRIMOS reads profile information in the SAD and uses it to initialize attributes of User 1. The user name for the supervisor terminal is SYSTEM and the project with which SYSTEM is affiliated is the project specified in the profile.

Making an entry for user SYSTEM in the SAD enables you to set the Initial Attach Point and assign specified values for command environment attributes, including the number of dynamic and static segments for User 1. If you do not make an entry in the SAD for user SYSTEM, PRIMOS uses the system defaults for the profile. If the SAD does not exist or an entry for SYSTEM is not present or the project is invalid, the following messages are displayed at startup time:

```
Can't attach to the SAD:  Not found.  (nlogin)

Profile data cannot be initialized from the SAD for the supervisor.
System defaults are being used.
```

If you have entry in the SAD for SYSTEM that you are using for administrative purposes, PRIMOS uses that entry to initialize profile data for User 1. You should create another, different identifier to replace the identifier SYSTEM because the identifier SYSTEM is reserved for User 1. If the SAD contains a user profile for SYSTEM, the following message is displayed at startup time:

```
Initializing profile data for the supervisor from the SAD.
```

System Attributes

When you invoke the EDIT_PROFILE command from the supervisor terminal to create a SAD, you are prompted for the name of the System Administrator. Entering the name SYSTEM enables a user at the supervisor terminal to run EDIT_PROFILE. At Rev. 20.2, PRIMOS uses the entry for SYSTEM to initialize profile data of User 1. If you have an entry in the SAD that you are using for administrative purposes, you should create another identifier, different from SYSTEM, for these purposes.

Note

> It is recommended that you do not create a SAD that enables you to run EDIT_PROFILE from the supervisor terminal if the terminal is accessible to users and security is a concern. When EDIT_PROFILE prompts you for the name of the System Administrator, enter a name other than SYSTEM. This enables you to run EDIT_PROFILE from a user terminal under an identifier known only to the System Administrator and prevents users who can access the supervisor terminal from corrupting the SAD.

When you add a user to the system (using the EDIT_PROFILE ADD_USER command), you must specify a set of system attributes for the user. A user's set of system attributes consists of the following:

- A user ID
- A login password, which may be null
- A default affiliation with a project (optional)
- Membership in up to 16 systemwide ACL groups (optional). ACL groups are defined in the section below, ACCESS CONTROL LISTS.

These system attributes are stored in the system data base and take effect every time the user logs in. Thus, regardless of how many sets of project attributes a user has, the user always has the same set of system attributes.

Project Attributes

After adding a user to the system data base, you must enter the user as a member of at least one project. If your system has only project DEFAULT, EDIT_PROFILE automatically makes the user a member of that project.

The project attributes that you can define for a user are the following:

- An Initial Attach Point (also called the origin directory), which is the directory to which the user is attached at login.
- Membership in up to 16 project-specific ACL groups (optional). These project groups are in addition to the user's systemwide ACL groups. A user can be a member of up to 32 ACL groups.
- Four sets of command environment limits (also called EPF attributes). For details, see the section below, Command Environment Limits.

You do not have to define specific project attributes for each member of the project. You can create for each project a default Initial Attach Point, default ACL groups, and command environment limits. Any project member for whom you did not define specific project attributes uses the project default attributes. For example, if you defined two ACL groups for the project profile but assigned no ACL groups for user JILL, JILL assumes the two project ACL groups at login. Thus, an EDIT_PROFILE LIST_USER command for JILL shows "<none>" as her ACL groups, but a PRIMOS LIST_GROUP command shows her associated with the two project-specific ACL groups.

Note

> Default project attributes also apply to SYSTEM, the user name of the supervisor terminal (User 1). For SYSTEM, you should supply an Initial Attach Point, usually the UFD CMDNC0, rather than let the supervisor terminal be affiliated with a default IAP or an IAP designated for users. The entry under the name SYSTEM in the SAD is read at startup time and the Initial Attach Point that you specify in the SAD is the directory to which the supervisor terminal (User 1) is attached.

Project attributes are valid only when the user logs in as a member of that particular project. The four ways a user can log in as a project member are as follows:

- If the user supplies the project ID by using the -PROJECT option of the LOGIN command, the user is logged in as a member of the project.
- If a user's system attributes include a default project and if the user does not supply a project ID at login, the user is logged in as a member of that default project.
- If a user's system attributes do not include a default project and if the user does not supply a project ID at login, the user is prompted by PRIMOS for a project ID.
- If project DEFAULT is the only project on the system, all users are automatically logged in as members of project DEFAULT.

Command Environment Limits

Command environment limits determine the resources that a user has when using EPFs (Executable Program Formats). EPFs are dynamic runfiles (programs) that are assigned by PRIMOS, at runtime, to any free segments. Users can suspend EPFs and then reinvoke them without loss of data by running the EPFs in different command levels of PRIMOS and in any segments not already in use by other live EPFs.

The four command environment limits are the following:

- Maximum number of command levels
- Maximum number of live program invocations per command level
- Maximum number of private dynamic segments
- Maximum number of private static segments

Command environment limits exist on the system level, project level, and user level. Project limits can be the less than or equal to the system limits, but cannot be greater than the system limits. Similarly, a user in the project can have the same or lower limits as the project, but cannot use more resources than the project limits.

In addition to limits, each project profile may have a set of default values. These default values must be equal to or less than the values of the project's limits.

The system also has a set of default values. The system supplies the following default values:

- 10 command levels
- 10 program invocations per level
- 32 private dynamic segments
- 32 private static segments

Assigning User Command Environment Limits: When you create a project with EDIT_PROFILE, you must first define the command environment limits for that project. Then, when defining the project profile, you can define the command environment attributes for the project. The project attributes must be equal to or less than the project limits.

When you add a user to a project, you define the user's command environment limits, in one of two ways:

- Assign the user a specific set of command environment limits. The limits must be equal to or less than the project's limits.
- Do not assign the user a specific set of command environment limits. If you select this option, when the user logs in as a member of the project, the user assumes the project profile's attribute values (not the project limit values) as command environment limits. If the project profile has no defined attributes, the user assumes the system default values.

Users can find out their assigned limits by using the PRIMOS LIST_LIMITS command.

Table 3-1 lists the recommended defaults, minimum values, and maximum values that the System Administrator can assign to each of the four attributes of a project or user.

Table 3-1
Command Environment Limits

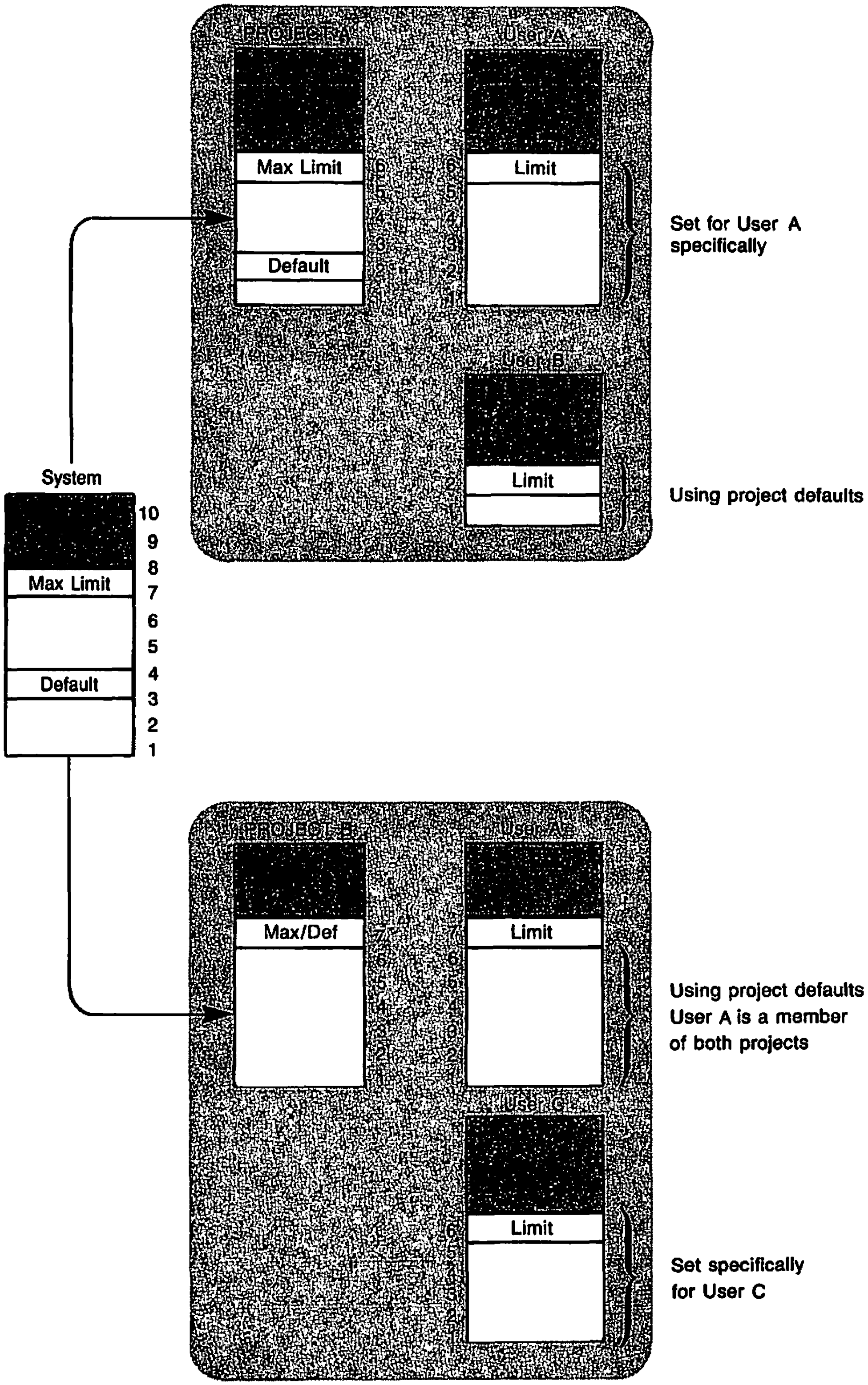
Command Attribute	Minimum Value	Maximum Value	Recommended Value
Command levels	1	100	10
Live invocations per level	1	100	5
Private dynamic segments	16	504	40
Private static segments	8	496	40

Notes

The sum of the static and dynamic private segments cannot exceed 512.

To submit batch jobs, a user must have at least two command levels. A user's batch jobs will fail if the user is set up with only one command level.

Figure 3-1 shows command environment attributes on a system with two projects and three users. As the figure shows, the project limits can be the same as the system limits, but cannot be greater than the system limits. Similarly, a user in the project can have the same or lower limits as the project, but cannot use more resources than the project limits.



Set for User A specifically

Using project defaults

Using project defaults User A is a member of both projects

Set specifically for User C

Setting Command Environment Defaults and Limits
Figure 3-1

Figure 3-2 shows a system with one project and two users. User A is using project default limits for command level and breadth, while User B has had these limits set.

	System		Project		User A (Uses Defaults)	User B (Set Specifically)
	LIMITS	DEFAULT	LIMITS	DEFAULT	LIMITS	LIMITS
Command Levels	100	10	15	5	5	8
Programs Per Level	100	10	10	5	5	8
Private Dynamic Segments	504	32	80	40	40	60
Private Static Segments	496	32	80	40	40	50

Command Levels
Figure 3-2

User Profiles at Login

When a user logs in, the PRIMOS login program uses the user's system attributes and project attributes to establish the user's environment.

The user's environment can be further customized with a user-supplied login program. This program must be named LOGIN.RUN, LOGIN.SAVE, LOGIN.CPL, or LOGIN.COMI. PRIMOS looks for the program in this order. The user login program must be stored in the user's origin directory. The user login program can perform such tasks as the following:

- Set terminal characteristics, such as the erase and kill characters
- Change the system OK, and ER! prompts
- Activate abbreviation and/or global variable files
- Run other user-defined programs

See Chapter 5, SECURITY, for a detailed explanation of the login procedure.

THE USER PROFILE DATA BASE

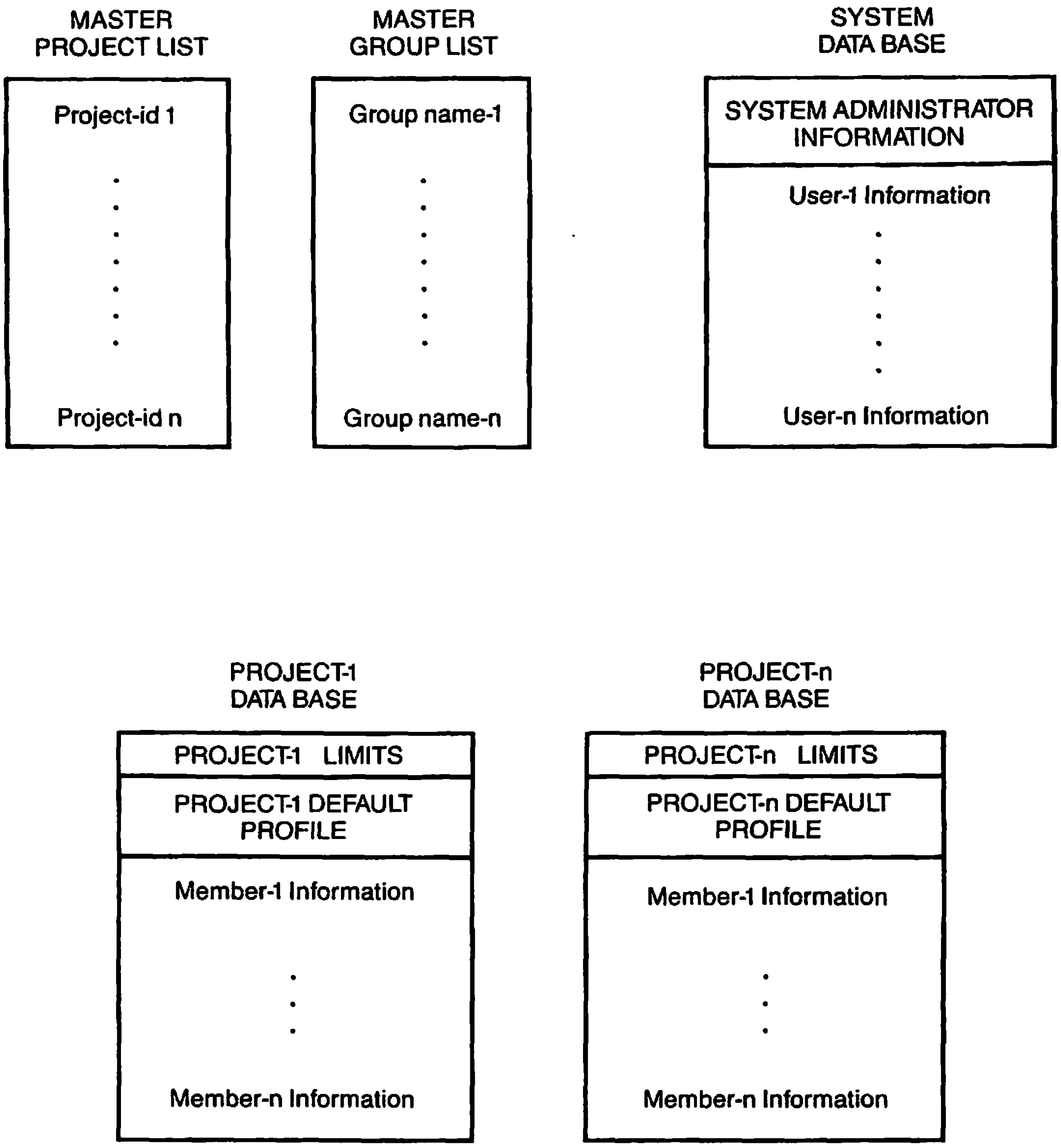
The User Profile Data Base can be described from the point of view of the system, the user, and the System Administrator.

From the system's point of view, the data base is a directory named SAD (for System Administration Directory) that resides in the MFD of the system's command partition.

From the System Administrator's point of view, the data base is a collection of four types of lists:

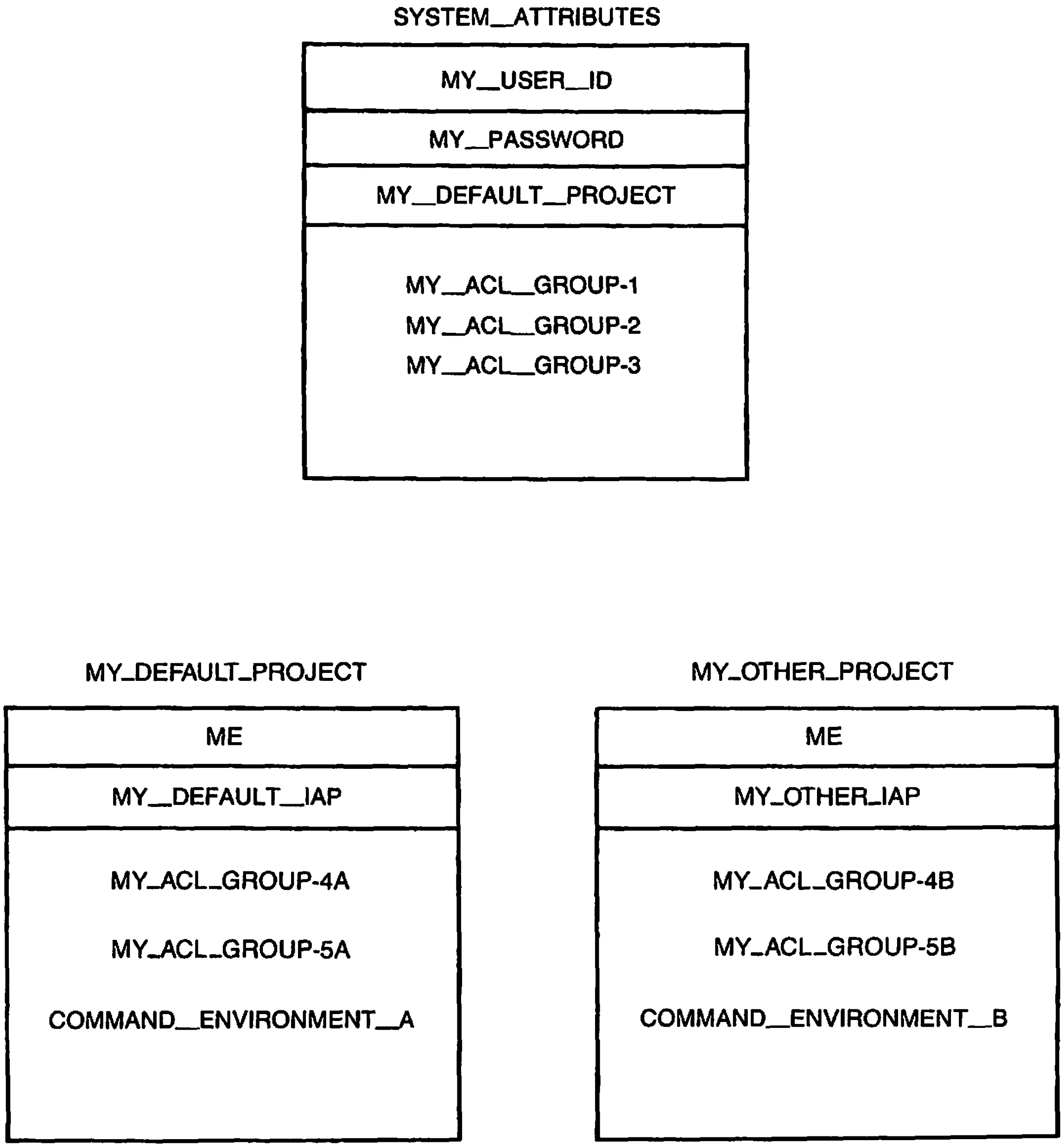
- A master list of every project name that you define for your system.
- A master list of every ACL group name that you define for your system. If you are not using ACLs on your system, this list is not part of your data base.
- The system data base, which contains an entry for every user that you define to the system, beginning with the System Administrator. The entry lists the user's system attributes.
- One or more project data bases. Each project that you define has its own separate project data base. Project data bases are described in the following section.

Figure 3-3 illustrates the User Profile Data Base from the point of view of the System Administrator.



User Profile Data Base From Administrator's Viewpoint
Figure 3-3

From the user's point of view, the data base is two or more sets of the user's attributes: one set of systemwide attributes used at the system level during every session, and one or more project-specific sets used when logged in as a member of a project. The user's point of view resembles that sketched in Figure 3-4.



User Profile Data Base From User's Viewpoint
Figure 3-4

Project Data Bases

At login, a user is always assigned the attributes listed for that user in the system data base. In addition, the user is also assigned the attributes of the project with which that user is associated during a terminal session. These project attributes are stored in project data bases, with each project having its own project data base.

A project data base contains five types of material:

- The user ID of the Project Administrator. (The Project Administrator does not have to be a member of the project.)
- A list of the project limits. The limits consist of the four command environment limits and a list of all the ACL group names designated for this project (assuming you are using ACLs). The list provides a pool of project-specific group names that you or the Project Administrator can assign to the project profile and to users.
- A list of the project profile. The profile consists of the command environment attributes, the project default Initial Attach Point, and the project default ACL groups. If the latter two items are defined, users who are assigned no specific Initial Attach Point or ACL groups use the project defaults instead. If the command environment attributes are not set for the project, users who are assigned no specific command environment limits use the system defaults instead.

Note

> If the project profile does not contain a default Initial Attach Point, you must assign each user an Initial Attach Point, or project members cannot log in.
>
> You should not rely on a default Initial Attach Point that is assigned to other users as the attach point for the supervisor terminal (User 1). The usual Initial Attach Point for the supervisor terminal (User 1) is UFD CMDNC0.

- An entry for each user who is a member of the project. The entry contains the user's project attributes, as defined above in the section, Project Attributes.

System attributes (as defined in the system data base) are always assigned to each user on an individual basis, but project attributes may be assigned to a project member by default. Assigning project attributes by default allows you to combine the security of individual user IDs and passwords with the convenience of group access to the file system.

Example of a Project Data Base: As an example of using project defaults for project members, consider the data base for Project DENMARK, portrayed in Figure 3-5.

```
Project:  DENMARK

Project Administrator:  CLAUDIUS


Project Limits:
  ACL Groups:  .DANES  .PRINCES
  Command Environment Limits:
    Command levels:       20
    Programs per level:  20
    Dynamic segments:    100
    Static segments:     100


Project Profile (project defaults):

  Default IAP:  <DRAMA>DENMARK>ELSINORE
  Default ACL Groups:  .DANES
  Command Environment Attributes:
    Default command levels:      5
    Default programs per level:  5
    Default dynamic segments:   40
    Default static segments:    40


User ID:  CLAUDIUS
  IAP:
  ACL Groups:
  Command Environment Limits:

User ID:  HAMLET
  IAP:
  ACL Groups:
  Command Environment Limits:

User ID:  GERTRUDE
  IAP:
  ACL Groups:
  Command Environment Limits:

User ID:  HORATIO
  IAP:
  ACL Groups:
  Command Environment Limits:
```

Data Base for Project DENMARK
Figure 3-5

As Figure 3-5 shows, no member of the project has a specifically assigned Initial Attach Point. Instead, project members share one Initial Attach Point, the directory <DRAMA>DENMARK>ELSINORE. They also share membership in a common ACL group, .DANES. In addition, command environment limits have not been set for any user. All project members use the profile command environment attributes as their limits.

The directory ELSINORE, the ACL group .DANES, and the command environment attributes are project defaults: they were defined for the project by the System Administrator, and therefore did not need to be defined for each member of the project.

All project members, like all other users on the system, also have a set of system attributes, which includes a unique user ID and a login password. This arrangement provides good login security and also makes it possible to change any user's profile, if the need for special privileges arises.

As an example of a need for special privileges, suppose that Project Administrator CLAUDIUS determines that he and HAMLET need special access rights to a group of files, and that HAMLET needs greater command environment resources because he will be debugging a particularly large test program.

Using the EDIT_PROFILE CHANGE_USER command, CLAUDIUS edits the project DENMARK data base and assigns himself and HAMLET to the group .PRINCES, one of the ACL groups provided by the System Administrator as part of the project limits. CLAUDIUS then specifically sets HAMLET's four command environment limits to values greater than the project defaults (but still less than the project limits).

Figure 3-6 shows the project data base after the changes made by CLAUDIUS. All members except for HAMLET and CLAUDIUS still share the project default attributes. HAMLET and CLAUDIUS still share the default origin directory, but they now have their own ACL groups rather than the default ones. HAMLET, in addition, has specific command environment limits, which can be displayed with the EDIT_PROFILE LIST_USER command.

Note that the System Administrator took no part in making the changes. This is another advantage of projects: their use allows Project Administrators to perform much of the day-to-day administration that the System Administrator would otherwise have to do.

```
Project: DENMARK

Project Administrator:  CLAUDIUS


Project Limits:
  ACL Groups:  .DANES  .PRINCES
  Command Environment Limits:
    Command levels:      20
    Programs per level:  20
    Dynamic segments:   100
    Static segments:    100


Project Profile (project defaults):

  Default IAP:  <DRAMA>DENMARK>ELSINORE
  Default ACL Groups:  .DANES
  Command Environment Attributes:
    Default command levels:       5
    Default programs per level:  5
    Default dynamic segments:   40
    Default static segments:    40


User ID:  CLAUDIUS
  IAP:
  ACL Groups:  .DANES  .PRINCES
  Command Environment Limits:

User ID:  HAMLET
  IAP:
  ACL Groups:  .DANES  .PRINCES
  Command Environment Limits:
    Command levels:      10
    Programs per level:  10
    Dynamic segments:    70
    Static segments:     70

User ID:  GERTRUDE
  IAP:
  ACL Groups:
  Command Environment Limits:

User ID:  HORATIO
  IAP:
  ACL Groups:
  Command Environment Limits:
```

Project DENMARK After Changes
Figure 3-6

ACCESS CONTROL LISTS

An Access Control List (ACL) is a mechanism for controlling access to a file or directory. The ACL contains a list of users and/or ACL groups, together with their access rights to the object that the ACL is protecting. To list contents of an ACL, use the LIST_ACCESS command.

Types of Access Rights and Identifiers

The access rights that can be granted by an ACL are shown in Table 3-2.

Rights in an ACL may be granted to the following identifiers:

- A user ID. This ID identifies an individual user.
- An ACL group. This group consists of a number of users grouped together for purposes of file access. The name of an ACL group always begins with a period (for example, .STAFF). For details on ACL groups, see the section below, ACL Groups.
- The special ID $REST. This ID identifies all other users (that is, any user who is not identified by an individual ID or is not a member of an ACL group listed in the ACL).

Rights may be granted by any user who has Protect access to the object and List access to its parent directory.

Rights may be provided in the following ways:

- By setting a specific ACL on the object with the SET_ACCESS command. Because specific ACLs are not separate objects but are linked to the object they protect, they do not appear when you issue the LD command.
- By creating an access category. This is a named file system object containing an ACL that protects whatever objects (within its own directory) you choose to link it to. Access category names have the suffix .ACAT and can be listed with the LD command.
- By using default protection. Default protection is provided by the parent directory (or its parent) if no specific ACL or access category has been set on an object.

Protection may be overridden by a priority ACL, which is set by the System Administrator or by an operator at the supervisor terminal. For details on priority ACLs, see Chapter 9, SETTING SYSTEM ACCESS.

Table 3-2
ACL Access Rights

Symbol	Right	Applies To	Meaning
R	Read	Files	File may be read or executed.
W	Write	Files	File may be modified.
X	Execute VMFA	Local EPF runfiles (no effect on remote EPF files)	Executable Program Format (EPF) file can be executed, but cannot be copied with the standard file system utilities. Read (R) access automatically includes X access.
U	Use	Directories	User may attach to directory.
L	List	Directories	Directory contents may be listed.
A	Add	Directories	Directory entries may be added.
D	Delete	Directories	Directory entries may be deleted.
P	Protect	Directories	Access rights and file attributes may be changed.
ALL		Files and Directories	All of the above rights.
NONE		Files and Directories	No access allowed.

Within an ACL, individual rights take precedence over group rights, and group rights take precedence over $REST rights. For example, assume the following ACL is in effect:

JANE	ALL
JOHN	LUR
.OTHERS	URW
.SOME	LURA
$REST	U

Individual rights take precedence: JANE has ALL rights, and JOHN has only LUR rights, whether or not JANE or JOHN are members of the .OTHERS or .SOME groups. Group rights are additive: if BILL is a member of both .OTHERS and .SOME, his rights are LURWA. $REST applies only to those users not mentioned in the ACL. (If $REST is not specified in an ACL, $REST:NONE is assumed.)

If a priority ACL is in effect, any user mentioned in the priority ACL (including $REST) takes the rights granted by the priority ACL. Otherwise, the user retains the rights from the regular ACL.

ACL Groups

An ACL group is a list of users who are grouped together for file access purposes. The name of an ACL group always begins with a period (for example, .STAFF or .ACCOUNTING). Thus, when reading an ACL, it is easy to tell which IDs represent individual users and which represent ACL groups.

There are two kinds of ACL groups: system-based and project-based. Both kinds of ACL groups are registered in the system data base. Project-based ACL groups are also registered in a project data base.

A system-based ACL group forms part of the user's entry in the system data base. The system-based ACL group is active every time the user logs in, regardless of which project the user logs in to. System-based ACL groups are often used for global system access. For example, .SUPER_USER might have ALL access to system UFDs.

Project-based ACL groups are part of the user's entry in a project data base. A user's project-based groups are active only when the user logs in as a member of that particular project.

A project ID often has a corresponding ACL group that contains all members of the project. For example, the project OPERATIONS might use an ACL group, called .OPERATIONS, for its members. In addition, project-based ACL groups may be used to distinguish the rights that each group within the project needs.

In a given ACL, individual rights override group rights. For example, assume the following ACL protects a directory:

```
JOHN:LUR
.JARS:ALL
```

JOHN has only LUR rights to the directory, even if he is a member of group .JARS.

Group rights, however, are additive. For example, assume the following ACL protects a directory:

```
.PROJECT_LEADERS:PD
.PROJECT_MEMBERS:ALURW
```

Any user who belongs to both groups has PDALURW (that is, ALL) access.

Defining ACL Groups

To define an ACL group, the System Administrator first uses EDIT_PROFILE to enter the name of the ACL group in the system data base. Such an entry can occur whenever a new user or project is added to the system data base, or when the attributes of an existing project or user are changed.

Then either the System Administrator or the Project Administrator uses EDIT_PROFILE to define various users as members of the group. Groups and their memberships are altered as needed. The data base can thus be kept up-to-date to reflect the current needs of the system.

Reasons for Using ACLs

It is recommended that you use ACLs as the primary means of providing file system security on your system. ACLs provide the following advantages:

- Better file system security than passwords
- An easy-to-use interface for users and programs to set and modify file system access
- Common access for specified groups of users under administrative control

Failure to use ACLs results in the following:

- Poor security on your User Profile Data Base
- Inability to use projects (other than project DEFAULT) on your system
- Decreased security on other subsystems

For a comparison of the security provided by ACLs and passwords, see Chapter 5, SECURITY. For further details on ACLs, see the Prime User's Guide.

DESIGNING YOUR DATA BASE

Before you use EDIT_PROFILE to create your User Profile Data Base, you should sketch out its design and parameters. The three general steps that you should take are the following:

1. Determine how you can group users or projects.
2. Determine the degree of security you want for your system.
3. Draw up lists of users and projects.

The following sections discuss these steps. The final section details the rules for user attributes.

Grouping Users

When considering how to group your users and projects, ask yourself some of the following questions:

- What groups do your users seem to fall into?
- Are there some logical dividing lines you might use to divide users into projects or to assign ACL groups?
- Are there any obvious candidates for Project Administrators? If so, what users would be in their projects, and what sort of ACL groups might those users need?

Determining the Degree of Security

The next question to consider is the degree of security you want for your system. Your answer to this question may determine the type of User Profile Data Base you create. The degree of security on a system may depend on the use of projects.

There are three main types of systems:

- A tightly controlled system with strong security locks at the system level. An example is an applications development group, where full access to any given set of files is restricted to a small set of people. This type of system is shown as Example 1 of the following section, EXAMPLES OF DATA BASES.
- A loosely controlled system with very little security at the system level. An example is a system used by a small business, where all users are allowed access to most of the data. Such a system is shown as Example 2 in EXAMPLES OF DATA BASES.

- A mixed system that combines tight security on some projects (and for some users) with a looser environment for other users. An example is a college, where it may be desirable to give one set of users (the faculty) greater access and privilege than would be given to another set of users (the students). This type of system is shown as Example 3 in EXAMPLES OF DATA BASES.

For more information on security considerations, see Chapter 5, SECURITY.

Drawing Up User and Project Lists

After you know how you want to organize your data base, draw up some lists of users and projects. These lists will help you visualize your system more precisely. The lists can also serve as reminders when you create the data base with EDIT_PROFILE. Table 3-3 lists the rules you need to follow for defining user and project attributes.

The following procedure assumes the use of several projects on the system. The procedure, however, can still be applied if you will use only the system default project named DEFAULT.

1. Draw up a master list of the projects you want on your system. For each project include the following:

   - The project ID.

   - The name and the user ID of the Project Administrator.

   - The project limits. These limits, which are mandatory, are the four command environment limits and a master list of all the ACL groups you want to make available for assignment for the project profile or to users.

   - The project profile. Any of the three attributes of the profile are optional. The attributes are the default Initial Attach Point, the default ACL groups, and the default command environment limits.

2. Create a master list of anyone who will be a system user. You and your Project Administrators can then assign people to projects from this list.

   After this step, you should have a master list of projects (from Step 1), a master list of system users, and a list of users for each project. (PRIMOS can efficiently accommodate up to 20,000 users in a project.)

   Figure 3-7 shows a sample form for creating a master user list. Figure 3-8 shows a sample form for creating a project data list.

3. Fill in the master list of users as follows:

   a. For each user, define a user ID and a temporary password. (You may choose to have users share IDs and passwords, or you may want a separate ID and password for each user.) You can either assign the IDs yourself, or distribute forms on which users can request the ID of their choice.

      Note

      If your system will be part of a network, you may want to have one person coordinate all user IDs on the network, to make certain that each ID is unique across the network. Further guidelines to network planning are given in the Network Planning and Administration Guide.

   b. List all the projects to which the user should be assigned.

   c. Decide which project (if any) is to be the user's default project.

   d. List the systemwide ACL groups (if any) to which you want this user to belong.

   e. List the command environment limits for the user if they are to be different from the project defaults.

4. Fill in each project user list. For each user, specify

   - The user ID

   - The Initial Attach Point (unless the user will use the project default)

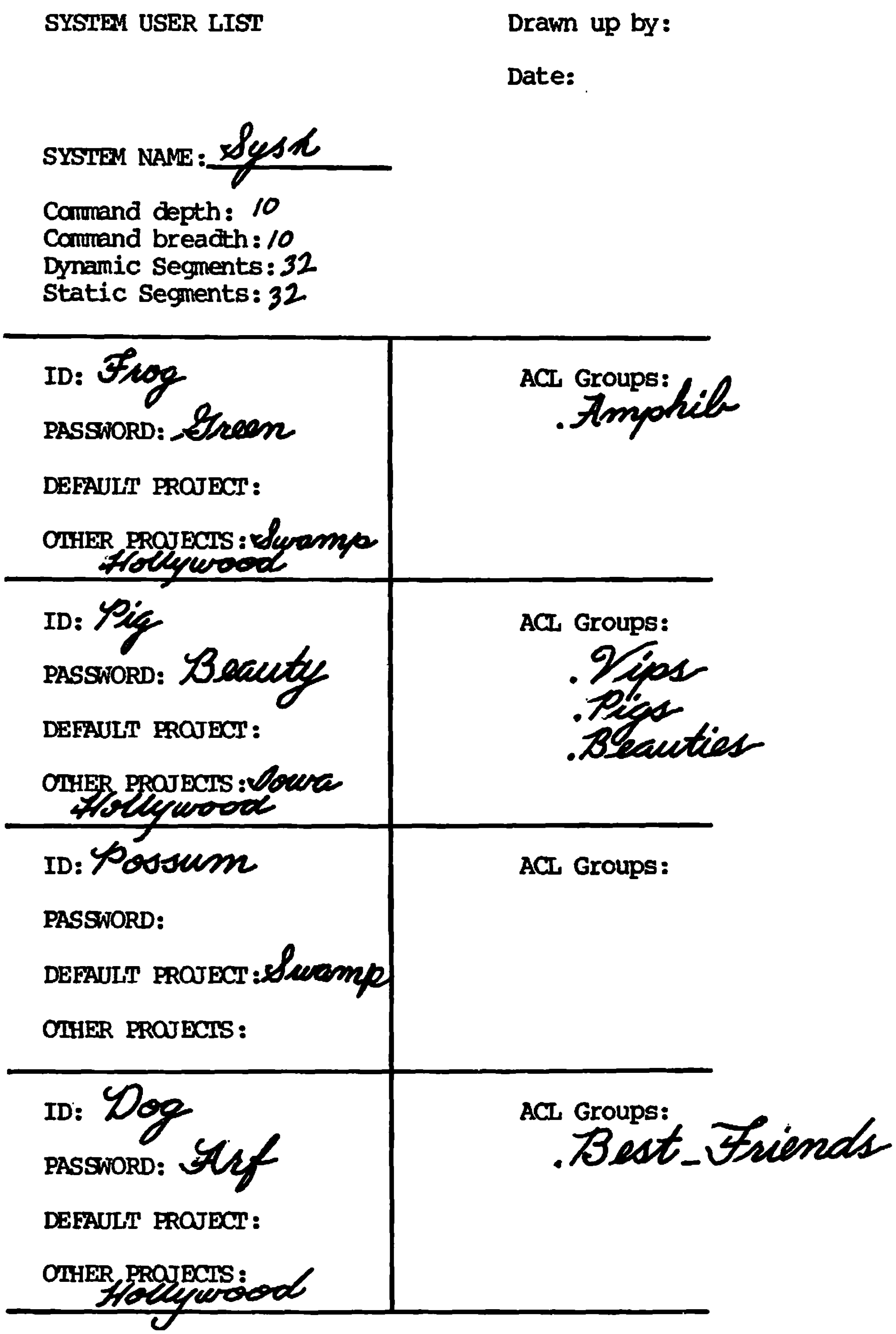
   - A list of project-specific ACL groups (if any) to which you want the user to belong

   - A list of command environment limits (if other than the default)

When your lists are complete, you are ready to set up the User Profile Data Base for your system. Chapter 12, USING EDIT_PROFILE, explains how to use the EDIT_PROFILE utility to build your data base.

Table 3-3
Rules for User and Project Attributes

User ID	Must be 1-32 characters long; must begin with an alphabetic character; can contain only letters, digits, periods (.), underscores (_), and dollar signs ($).
User password	Must be 0-16 characters long. A password with no characters is a null password (that is, the user enters only a carriage return). (In EDIT_PROFILE, the NO_NULL_PASSWORD command allows or prohibits null passwords and the MINIMUM_PASSWORD_LENGTH command sets a minimum length.) May contain any characters except PRIMOS reserved characters, which are defined in the Prime User's Guide.
ACL group name	Must be 2-32 characters long; must begin with a period; can contain only letters, digits, periods (.), underscores (_), and dollar signs ($).
Project name	Follows the same rules as user IDs.
Command environment limits	Command levels, 1-100; live program invocations per level, 1-100; private dynamic segments, 16-504; private static segments, 8-496; combined private dynamic and static segments, maximum of 512.
System Administrator	At any given time, only one user ID can represent the System Administrator on a system. (Any number of people may share administrative duties by using that user ID.)
Project Administrator	Only one Project Administrator (PA) for any project at any given time; a PA may administer more than one project at one time; the System Administrator may also act as PA for any number of projects; if the only project in use is DEFAULT, the System Administrator is automatically its PA; EDIT_PROFILE automatically registers all PAs as members of a systemwide group named .PROJECT_ADMINISTRATORS$. (Because no user can belong to more than 16 systemwide groups, a PA can belong to only 15 other systemwide groups.)



SYSTEM USER LIST

Drawn up by:

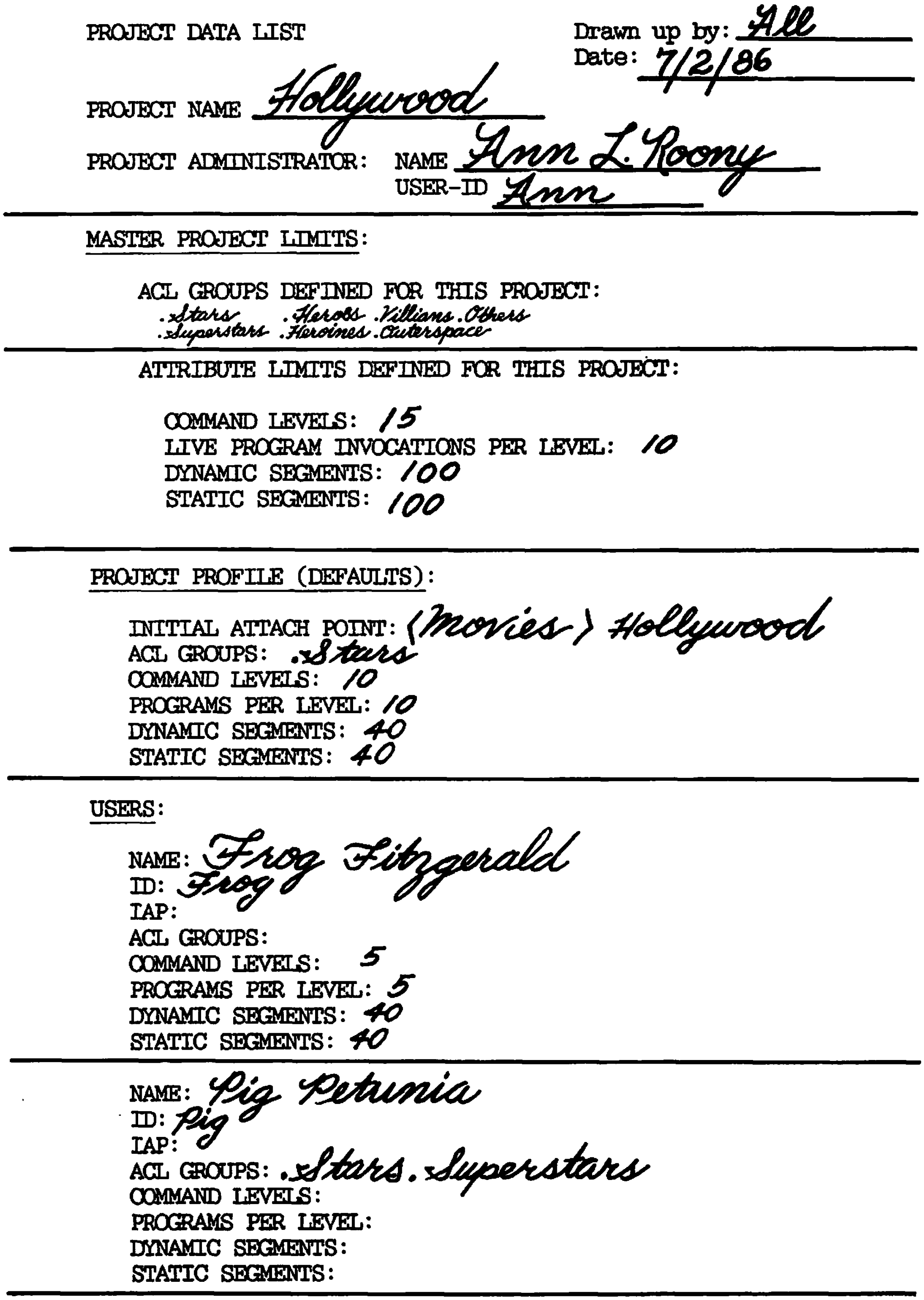
Date:

SYSTEM NAME: Sysd

Command depth: 10
Command breadth: 10
Dynamic Segments: 32
Static Segments: 32

ID: Frog PASSWORD: Green DEFAULT PROJECT: OTHER PROJECTS: Swamp Hollywood	ACL Groups: .Amphib
ID: Pig PASSWORD: Beauty DEFAULT PROJECT: OTHER PROJECTS: Iowa Hollywood	ACL Groups: .Vips .Pigs .Beauties
ID: Possum PASSWORD: DEFAULT PROJECT: Swamp OTHER PROJECTS:	ACL Groups:
ID: Dog PASSWORD: Arf DEFAULT PROJECT: OTHER PROJECTS: Hollywood	ACL Groups: .Best_Friends

Sample System List
Figure 3-7



PROJECT DATA LIST

Drawn up by: All
Date: 7/2/86

PROJECT NAME Hollywood

PROJECT ADMINISTRATOR: NAME Ann L. Roony
USER-ID Ann

MASTER PROJECT LIMITS:

ACL GROUPS DEFINED FOR THIS PROJECT:
.Stars .Heroes .Villians .Others
.Superstars .Heroines .Outerspace

ATTRIBUTE LIMITS DEFINED FOR THIS PROJECT:

COMMAND LEVELS: 15
LIVE PROGRAM INVOCATIONS PER LEVEL: 10
DYNAMIC SEGMENTS: 100
STATIC SEGMENTS: 100

PROJECT PROFILE (DEFAULTS):

INITIAL ATTACH POINT: (Movies) Hollywood
ACL GROUPS: .Stars
COMMAND LEVELS: 10
PROGRAMS PER LEVEL: 10
DYNAMIC SEGMENTS: 40
STATIC SEGMENTS: 40

USERS:

NAME: Frog Fitzgerald
ID: Frog
IAP:
ACL GROUPS:
COMMAND LEVELS: 5
PROGRAMS PER LEVEL: 5
DYNAMIC SEGMENTS: 40
STATIC SEGMENTS: 40

NAME: Pig Petunia
ID: Pig
IAP:
ACL GROUPS: .Stars.Superstars
COMMAND LEVELS:
PROGRAMS PER LEVEL:
DYNAMIC SEGMENTS:
STATIC SEGMENTS:

Sample Project List
Figure 3-8

EXAMPLES OF DATA BASES

The following three examples illustrate different types of systems and how their administrators employ user profiles, projects, and ACLs for their particular needs.

Example 1: The following example illustrates a tightly controlled system.

Team A, Team B, and Team C compete with each other. They must share the same partition, but no team is allowed to see what the other two are doing.

1. The System Administrator (SA) checks the partition and finds out that there are about 30,000 records for the teams to share.

2. The SA creates three UFDs named A, B, and C. He sets a quota of 10,000 records on each UFD, which establishes the space for each team.

3. The SA creates the data base for the three teams. Using EDIT_PROFILE, he creates a project named ALPHA. The SA designates user AMY as the Project Administrator (PA) for project ALPHA and sets up three ACL groups for project ALPHA to use. The first ACL group, .TEAMA, contains all the project members. The other two, .SOMEA and .OTHERA, are left empty for the PA to use as she wishes. With these groups, she can limit access within the project's directories to particular subgroups of project members. The SA then defines the command environment limits for project ALPHA.

4. The SA registers all members of Team A as users of the system, and as members of project ALPHA. UFD A is the Initial Attach Point of anyone in Team A. (If UFD A had subdirectories, any of those subdirectories could serve as Initial Attach Points for team members.)

5. The SA creates project BETA for Team B and project GAMMA for Team C. In the same manner that he registered members of Project ALPHA, the SA registers the members of Teams B and C in their respective projects.

6. The SA uses the PRIMOS SET_ACCESS command to create ACLs that set access protection on the A, B, and C top-level UFDs.

   The ACL for UFD A is set as follows:

   ```
   AMY: ALL
   .TEAMA: DALURW
   $REST: NONE
   ```

The ACL gives Project Administrator AMY ALL rights to her project's UFD, including the right to set protection on any subdirectories she may create. All other project members (.TEAMA) have the right to do everything except set or change the protection on files or directories. (AMY may later give them protection rights over individual subdirectories.) Any user who is not AMY or who is not a member of .TEAMA is included in the $REST special identifier and has no access rights. A user in $REST cannot attach to UFD A, use its files, or gain any information about its contents.

ACLs for UFDs B and C are similar.

7. The SA keeps full control of the MFD by setting an ACL on it that reads as follows:

```
SYS_ADMIN: ALL
$REST: U
```

This ACL allows the system's users to attach to the partition, but does not let them list or read the contents of the MFD. The ACL also denies the Project Administrators the right to change access to their top-level UFDs. To do that, the Project Administrators need both L (List) and U (Use) rights.

8. The SA continues to add users to the system, and to set up new projects as needed. If one of the three teams is dissolved, the SA will remove that team's project from the system.

Meanwhile, the three PAs take care of administrative chores within their own projects. AMY, for example, can use EDIT_PROFILE to put three project members into the .SOMEA group. However, if she asks EDIT_PROFILE to access project BETA (or, for that matter, the nonexistent project DELTA), she gets only the message: "Not a valid project."

Similarly, all members of Team A can work at will within their own directory and its 10,000 records. The other two directories, however, are invisible to them. Members of Team A cannot attach to directories B or C, cannot list or read any information from them, and cannot copy information in or out of them. Thus, Team A members are completely isolated from the B and C directories by the ACLs on those directories.

ACLs and projects last until you change or delete them. For example, suppose that Teams A, B, and C suddenly have to cooperate on a new and large project. The System Administrator sets up a new project called DELTA. Users belong to DELTA as well as to their original project. A member of Team A, for example, belongs to projects ALPHA and DELTA, while someone from Team B is a member of BETA and DELTA projects. At login, users specify which project they want to work on by supplying the project ID to the LOGIN command, as in the following example:

```
LOGIN ALAN -PROJECT DELTA
```

The ACLs for the new project build on the ACLs already established, so that they look like the following:

```
AMY: ALL
.TEAMA: DALURW
.TEAMB: DALURW
.TEAMC: DALURW
```

This method of setting up project DELTA is especially appropriate if any of the following applied:

- The three older projects are still ongoing.
- There is enough disk space for project DELTA to occupy.
- The accounting department wanted to keep the four projects separate.

Example 2: The following example illustrates a loosely controlled system.

A small group of people work cooperatively in a very friendly environment. They have a computer dedicated to their use, on which they share administrative responsibilities.

This group uses the simplest possible system. They have one default project (automatically named DEFAULT) to which everyone in the group belongs. No ACL groups are defined. The command environment limits are set to 10 command levels, 10 invocations of programs per level, 100 dynamic segments, and 100 static segments to allow plenty of freedom.

The ACLs on their MFDs read as follows:

```
SYS_ADMIN:  ALL
$REST:  DALURW
```

ACLs on top-level UFDs read as follows:

```
$REST:ALL
```

If a user needs special protection on a particular directory, that user sets it.

One person is known to the system as System Administrator. However, nothing prevents other members of the group from using the System Administrator's user ID and performing administrative tasks, assuming they know the System Administrator's password.

If this group decided to network their computer with other computers, they would probably want to add some protection. They could do this without disturbing their own rights as follows:

1. The SA adds an ACL group, .US, to the system. She registers all the system's users as members of that group.

2. The SA changes the ACLs on the system's MFDs to read as follows:

   ```
   SYS_ADMIN: ALL
   .US: DALURW
   $REST: LUR
   ```

   The new ACL does not restrict the rights of the original users because they are all members of the group .US. Users of other systems would have restricted privileges. They can attach to directories on these partitions, list directory contents, and read files. Other ACLs set on lower directories could grant additional rights either to all users of the network or to particular users or groups from other systems.

Example 3: The following example illustrates a mixed system.

The math department at a small college has bought a Prime computer. They plan to use it for four undergraduate courses, two graduate courses, and several research projects. In addition, the math faculty will use the computer for writing papers and articles, keeping records, and other tasks. The department head will act as SA.

1. The SA sets up a default project (named DEFAULT) for faculty members, graduate students working on research products, and whatever guests may visit the system.

2. The SA sets up six additional projects, one for each of the six math courses whose students will use the computer. As research projects are defined, she may set up projects for them as well.

   Professor Jones, who teaches the two graduate courses, chooses to act as Project Administrator for his two projects. The department secretary acts as Project Administrator for the other courses.

3. The SA sets up one systemwide ACL group, .FACULTY, and places all faculty members in the group. She defines project-based ACL groups for each math course: .M105, .M210, etc. For members of project-based ACL groups, the SA restricts the number of command levels to 5, the number of live invocations of EPFs per level to 5, and the number of both dynamic and static segments to 40. These limits are sufficient for the needs of the undergraduate students, and ensures that enough system resources are available for the graduate students and faculty who require more computing power, even when the system is most heavily used. For the graduate courses, she defines a few other ACL groups that may be used for joint projects.

   After the system is established, teams of a transitory nature may arise. These teams may want security for their work, but there will be no accounting or administrative need to create a formal project for them. In these cases, the SA can create new system-based ACL groups for the teams to use during their lifetime.

4. The SA sets up the top-level UFDs on the system, protecting them with the following ACL:

   ```
   SYS_ADMIN: ALL
   .FACULTY: DALURW
   $REST: U
   ```

   She sets a quota on each UFD, to prevent arguments over space usage.

   The faculty members, who can create (and protect) subdirectories as they need them, establish one subdirectory for each of the six courses that will use the computer. They then inform the SA and PAs what those directories are and what protection they want on them.

   For example, Professor Black wants her students to work cooperatively on projects. She wants her course directory ACL to read as follows:

   ```
   BLACK: ALL
   .FACULTY: LUR
   .M210: DALURW
   ```

Professor White does not want students in his course to share information. He wants his course directory's ACL to read as follows:

```
WHITE: ALL
.FACULTY: LURA
.M108: LURA
```

Professor White then creates an individual subdirectory for each student to work in. He sets an ACL on each of these directories that reads as follows:

```
(student-id): DALURW
```

In this way, the students cannot see each other's work. However, they can read the messages Professor White places in the course directory and can also place messages there themselves.

5. When the term begins, the students for each course are enrolled in their respective projects.

   Because their Initial Attach Point must be controlled by their project affiliations (and because one student may be enrolled in more than one course), students must specify project IDs when they log in, as in the following example:

   ```
   LOGIN J2943 -PROJECT M105
   ```

   When the term ends, either the students are removed from the projects or the projects are removed from the system.

   The students can remain in the system data base until they graduate. While they are enrolled in courses, their project affiliation and their presence in ACL groups allow them to work on the system. At other times, they have either no access or very limited access, depending on whether the SA has set the system to require a valid project ID for login.

4
Disks and Tape Drives

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.

Providing optimum efficiency and security for your disk space requires the following decisions and responsibilities in setting up disk space:

- Knowing the type and storage capacity of your disks
- Dividing your total disk space into subdivisions (called partitions) and distributing the partitions on your system
- Allocating paging space, which includes making such decisions as whether to use one or two paging partitions, how much space to allocate for paging, and whether to use split disks for paging
- Allocating user space by setting 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.

Two 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 disks (with the FIX_DISK utility).
- Chapter 17 of this guide, SYSTEM MONITORING, describes how to monitor your system's disk space.

DISK TYPES AND STORAGE CAPACITIES

At Rev. 20.2, Prime supports four types of disks:

- Storage Module Disks (SMDs)
- Fixed-Media Disks (FMDs)
- Cartridge Module Devices (CMDs)
- Diskettes

Except for diskettes, each type has 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

Storage Module Disks (SMDs) are removable platters enclosed in 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 five and 19 usable surfaces, respectively.

Fixed-Media Disks

Fixed-Media Disks (FMDs), also called Winchester disks, are permanently fixed to and enclosed in dust-free drives. Prime supports eight storage capacities for Winchesters: 60, 68, 120, 158, 160, 315, 496, and 675 megabytes. The 68 and 158 megabyte versions are available only on the Prime 2250™ . The 60 and 120 megabyte versions are available only on the Prime 2350 and Prime 2450.

Cartridge Module Devices

Cartridge Module Devices (CMDs) come in one of three storage capacities: 32, 64, and 96 megabytes of memory. Each CMD is made up of one removable cartridge platter (two surfaces, one of which is used for data) and either one, three, or five fixed surfaces that are permanently attached to the cartridge drive.

Diskettes

Diskettes, also called floppy disks, are small, bendable (hence "floppy") disks that are inserted like cartridges into disk drives that usually support low-memory capacity systems. All diskettes that Prime supports have four sectors (or records) per track and a total of 304 records (896 bytes/record). The memory capacity for each diskette is roughly 272 kilobytes.

DIVIDING DISK PACKS

Before you format your disks, you must decide the following:

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

Your two goals in this process are as follows:

- 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 workload evenly among your disk drives and controllers

Dividing Total Disk Area Into Partitions

As the person who knows the nature of your users' work, you are most qualified to create partitions for user groups. When you create partitions, you should know the following:

- The number of users in your logical user groups. For example, how many users are in the payroll group, the manufacturing group, the inventory control group?

- 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 amount of security required by each group. How much confidential information is handled by each group?

- 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. Try to set up all, or nearly all, of your disk space as one partition. This approach places the extra storage on the inner surfaces of the partitions, not on the outside surfaces that are not being used.

Planning ahead means being prepared for any group's work to increase or become an urgent project at your installation. That group's partition would therefore require more disk space. If you allotted ample space to each partition at the outset, the group does not have to stop work so that its partition can be reformatted.

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 using large partitions are

- Holding large data bases

- 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: Small partitions can provide one more level of data security, in addition to protect rights (in ACL systems), or owner rights (in directory-password systems). Smaller partitions also provide a convenient way to guarantee read protection as well as write protection.

Other advantages of using small partitions are as follows:

- Less data is lost if the partition is damaged or 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.
- More flexibility in deciding how many directories you want to have online at any given time.

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

Backup Considerations: In disk-to-disk backups, source and target partitions must be of equal size. Therefore, you might want to standardize the sizes of your partitions as much as possible.

For example, an 80-megabyte drive has five surfaces, and a 300-megabyte drive has 19. If you have one 80 and one 300, the only way you can do disk-to-disk backups is to have partitions of two, three, or five surfaces (that is, partitions that fit equally well on either drive). Larger partitions on the 300 have to be backed up on tape. However, if you had two 300 drives, each disk pack could be one partition; the two drives could still back each other up.

For more information on backups, see the Operator's Guide to System Backups and Chapter 14 of this guide, BACKUPS.

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 more efficient.

For example, if you have five partitions, two drives, and two controllers, you might place the three smallest partitions on one drive and the two largest 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 occur simultaneously.

On the other hand, if the smaller partitions are used more heavily than the larger ones, you might want to divide the smaller partitions among the two drives. 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.

Four PRIMOS commands that monitor system operations and data storage information are AVAIL, LIST_QUOTA, STATUS, and USAGE (especially the USAGE -DISKS -FREQ 30 command). These commands, along with other commands and information on monitoring the system, are discussed in Chapter 17, SYSTEM MONITORING, as well as in the PRIMOS Commands Reference Guide and the Operator's Guide to System Monitoring.

Pre-Rev. 20 Partitions

Rev. 18 and Rev. 19 partitions can be used on a Rev. 20 system. However, to gain the advantages of Rev. 20; you must convert pre-Rev. 20 partitions to Rev. 20. Rev. 18 and 19 partitions cannot use Rev. 20's hashed directories and the Date/Time Created and Date/Time Accessed attributes for files and directories. (Rev. 18 partitions also cannot use quotas and ACLs.) To convert pre-Rev. 20 partitions to Rev. 20 partitions, use the MAKE utility.

Although it is not likely that you will change Rev. 20-format partitions to a Rev. 18- or Rev. 19-format, you can perform such a conversion with the -DISK_REVISION option of MAKE. See the Operator's Guide to File System Maintenance for details.

A Rev. 20 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 system and attached to a Rev. 20 partition cannot display the Rev. 20 Date/Time Created and Date/Time Accessed file and directory attributes with the LD command.

ALLOCATING PAGING SPACE

Paging is a system process that divides large programs and data files into subdivisions called pages. PRIMOS moves pages between the paging partition and main memory as the pages are needed (a process known as demand paging). For example, when running a large program, PRIMOS divides the program into page blocks, loads the page that is needed immediately for processing, and stores the other pages on the paging partition until the processor needs them. When PRIMOS needs another part of the program, it reads a page from the paging partition and loads it into main memory.

Paging space, therefore, can be thought of as a temporary storage area where memory contents are held while waiting to be used by PRIMOS. Paging frees up main memory space for other users and increases the processor's speed and efficiency.

As System Administrator, you are responsible for determining how to set up your paging space. The two decisions you must make are whether to use one or two paging partitions, and how much space to allocate for paging. You must also use the MAKE utility to create the partition (or partitions) used for paging. For details on MAKE, see the Operator's Guide to File System Maintenance.

Using One or Two Paging Partitions

A system can use one or two partitions for paging. The PAGDEV directive tells PRIMOS which partition is the primary paging partition. If a second partition is used for paging, that partition (known as the alternate paging partition) is indicated with the ALTDEV directive. For further information on these directives, see Chapter 10, CONFIGURATION DIRECTIVES.

Because paging is part of your disks' workload, the choice of where to put paging partitions is part of the general task of trying to balance the workload across the system. If you have two or more disk drives, the primary paging partition should ideally be on a drive that is used infrequently. 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 if only 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 size of the segments.

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

The two methods for determining the amount of paging space are as follows:

- Use the rule of thumb given in the next section.
- 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. Small systems and lightly loaded systems will probably set paging space closer to the minimum than to the maximum figure. Large or heavily loaded systems will probably set their paging space closer to the maximum than to the minimum figure.

Rule of Thumb: A good rule of thumb for determining the amount of space you need for paging is to allocate one disk surface for paging for every 8 to 10 users. The number of users is the sum of the NTUSR, NRUSR, NPUSR, and NSLUSR configuration directives.

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

```
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 as follows:

```
    MIN_SPACE = PRIMOS + SHARED_PRODUCTS + (NUSR * 232)
```

where

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

PRIMOS — is the number of pages used by PRIMOS. To find this value, see the procedure described below.

SHARED_PRODUCTS — is the total number of pages used by the shared products on your system. Table 4-1 lists the number of pages per product and pages per user needed for each shared product. Calculate the figures for each shared product in use on your system. Use the total of these figures as the number for SHARED_PRODUCTS.

NUSR — is the sum of the configuration directives NTUSR + NPUSR + NRUSR + NSLUSR.

232 — is 29 segments per user * 8 pages per segment.

Use the following procedure to find out the value of the PRIMOS variable for the formula:

1. Edit the file PRIMOS.COMO in the top-level UFD named PRIMOS.

2. Use the Editor's search command (for example, LOCATE in ED) to for the word "RECORDS" (in uppercase). The number in front of the phrase "PAGE RECORDS" is the value for PRIMOS.

The following example illustrates this procedure:

```
    OK, ED PRIMOS>PRIMOS.COMO
    EDIT
    LOCATE RECORDS
    ***  45 SEGMENT(S),    82 WIRED PAGES,    936 PAGE RECORDS ***
```

In this example, 936 is the value for PRIMOS.

Table 4-1
Space Required by Shared Products

Product	Per-Product Pages	Per-User Pages
BASIC	0	48
BASICV	56	24
CBL	376	184
COBOL	72	8
DBG	224	64
DBMS	194	178
DPTX	53	(see Notes)
ED	24	8
EDB	0	8
EMACS	384	52
FED/FORMS	160	280
FTN	0	40
FTS	152	272
LOAD	0	16
MIDASPLUS ™	320	200
PMA	0	16
POWERPLUS	95	563
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

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: PT45™ , 64; PT46, 56; OWL, 53; 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).

Running Out of Virtual Memory

Following are the three conditions under which a system can run out of virtual memory, and their respective solutions.

- If a system runs out of system segments (indicated by the error condition NO_AVAILABLE_SEGS$), the NSEG or NVMFS configuration directives may need to be increased.

- If the paging partition becomes full (indicated by the condition PAGING_DEVICE_FULL$), the size of the paging partition must be increased.

- If a user runs out of segments (indicated by the message "Not enough segments"), the number of static or dynamic segments for that user may need to be increased with EDIT_PROFILE.

Using the PRATIO Directive to Enhance Performance

If your system does not use an alternate paging partition (specified with the ALTDEV directive), all paging activity occurs on the primary paging partition. If, however, you have an alternate paging partition, PRIMOS by default balances paging activity between the primary and alternate paging partitions. Performance can be improved somewhat if both paging partitions are on separate disk drives. Performance is improved slightly if the disk drives are on separate disk controllers.

If you have two paging partitions, PRIMOS by default performs approximately half of its paging space allocation on the primary paging partition and the other half on the alternate paging partition. This ratio is adequate for many installations. However, some installations perform more non-paging disk I/O on one disk drive than on another. For these installations, it may be desirable to use the PRATIO directive to reduce the paging space allocation on the frequently used disk drive and increase paging use on the less frequently used disk drive.

The PRATIO directive (described in Chapter 10, CONFIGURATION DIRECTIVES) determines how often the alternate paging partition is used in relation to the primary paging partition. This ratio is approximately n times out of 10. The default value of n is 5, which means the alternate paging partition is used approximately 50% of the time (5 times out of 10) and the primary paging partition is used the other 50% of the time.

To use the alternate paging partition more often, set n to a number between '6 and '12 (10 decimal). Setting n to a number lower than 5 causes the alternate paging partition to be used less often.

If the disk drive containing the primary paging partition is referenced twice as often as the disk drive containing the alternate paging partition, changing the PRATIO number to '7 or '10 (8 decimal) may improve system performance by balancing the use of the disk drives.

To determine the relative use of disk drives and controllers, use the USAGE command, described in the Operator's Guide to System Monitoring.

Split Paging Disks

A split disk is a partition that is split between paging space and file system storage. The paging space therefore takes up only part of the partition.

If you use storage module disks (which generally have no badspots) for paging, full paging partitions are recommended over split paging disks. (A badspot is a defective spot on the disk surface that cannot hold data.) If, however, the disk has badspots, you must either split the disk (so that can hold the badspot file) or use it only for file system storage.

Fixed-media disks (Winchester disks) are more likely to have badspots than storage module disks. If you use fixed-media disks, you will probably have to use split paging disks.

There are three general circumstances when you should consider using a split disk:

- If you have a very small system whose paging space requirement does not come close to taking up a full surface.

- If your disk surface has badspots. PRIMOS can handle badspots in the paging space, but cannot write the badspot file that handles them in the paging area itself. The disk is therefore split because paging takes place on the larger portion of the partition, while badspot handling (and perhaps other storage) takes place on the smaller portion. Note that in the case where the partition is split only because of the need to handle badspots, the file system portion is usually very small.

- If your system does a lot of paging and has a file system partition that is rarely used. Combining the two areas into a split disk can improve efficiency by compressing the paging space into a smaller number of cylinders.

For maximum utilization of a split paging disk, use the following two guidelines:

- The number of records for the paging portion of the disk should be evenly divisible by 16.

- Heavily used user or system files should not be stored in the file system storage portion of the disk.

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. The users, in turn, use your quota limits as a guide in setting quotas on their own subdirectories.

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 Prime User's Guide.

Note

Quotas cannot be placed on an MFD.

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 group or whether to give the groups more records than they need so that they can compete for the disk space.

The four major strategies for setting quotas on top-level directories are the following:

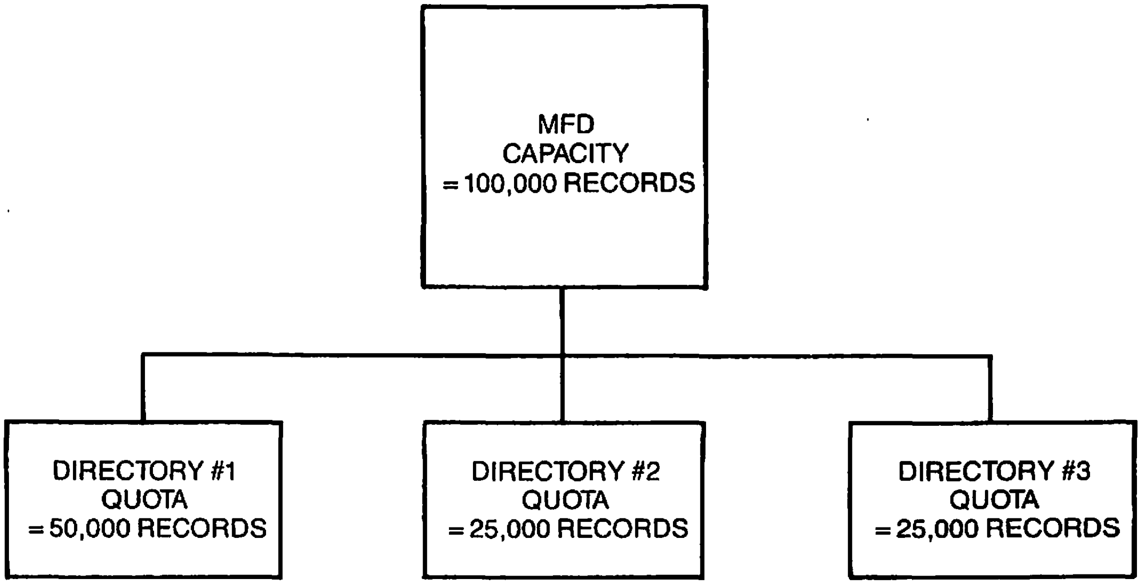
- The Exact strategy divides the exact number of user records among the top-level directories, thus guaranteeing that the partition's quota limit is not exceeded.

- The Overcommitted strategy maintains competition among users by setting the total number of records on the directories above the capacity of the partition.
- The Undercommitted strategy reserves space by setting the directory quotas below the capacity of the partition.
- The Unregulated strategy sets no quota on one or more directories.

The Exact Strategy: Use the Exact strategy when you want to distribute all the disk's space precisely among users.

For example, suppose your partition (MFD) has a capacity of 100,000 records that are reserved for users' work space. 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. Thus, if the partition has three top-level directories, you might give one directory 50,000 records and the other two directories 25,000 records each, according to which user group needed more space. After setting the quotas, you would monitor which top-level directories were using their space and modify the quotas accordingly.

Figure 4-1 illustrates the Exact strategy.



The Exact Strategy
Figure 4-1

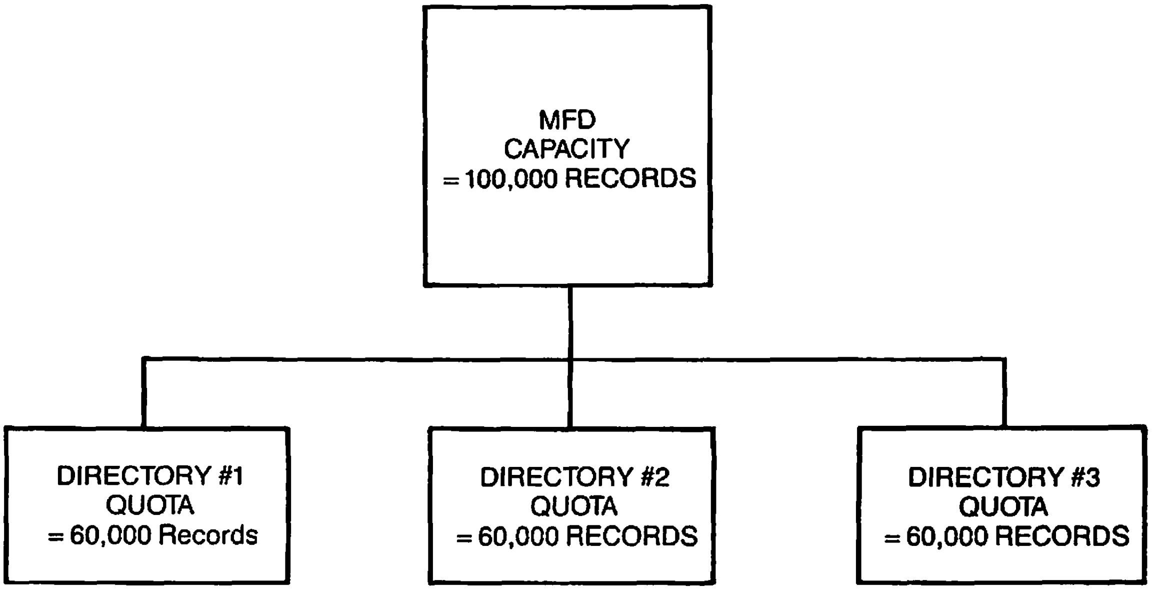
The Overcommitted Strategy: Use the Overcommitted strategy when you want to create competition among users, thus preventing them from underutilizing their disk space.

Competition for disk space can be maintained within a quota system by setting the quotas above the record capacity of the partition. Under this strategy, users are more inclined to use as much space as they need without feeling restrained by the limits. This strategy is particularly useful if you know your system has more than enough space to handle all your users' needs.

The disadvantage of the Overcommitted strategy, however, is that users may waste space by keeping unnecessary files and subdirectories.

As an example of the Overcommitted stategy, if you had a 100,000-record partition that contained three top-level directories, you could allocate 60,000 records to each top-level directory.

Figure 4-2 illustrates the Overcommitted strategy.



The Overcommitted Strategy
Figure 4-2

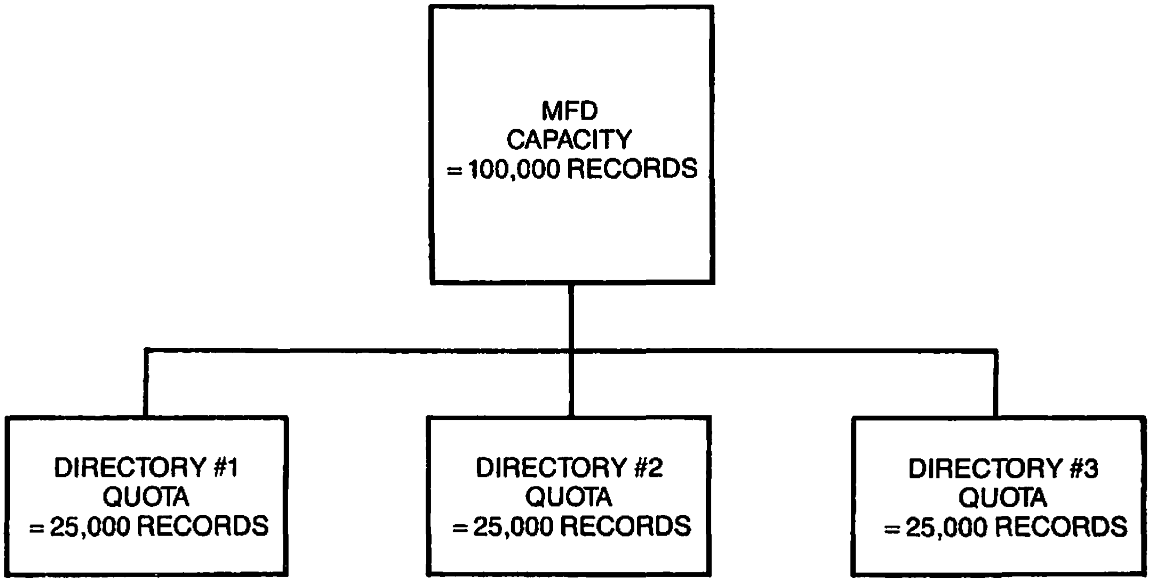
The Undercommitted Strategy: The Undercommitted strategy is the most strict and generally has the opposite effect of the Overcommitted strategy.

When disk space is scarce on your system, you can both reserve space and prevent users from exceeding the disk space capacity by setting the quotas below the partition's capacity.

This strategy creates an incentive for the users to be more efficient, reserving their space for essential data and deleting unneeded data. It also guarantees extra space on the system for emergency storage.

Using the 100,000-record partition of the previous examples, you could set a quota of 25,000 records on each of the three directories, thus ensuring that you would always have 25,000 records in reserve.

Figure 4-3 illustrates the Undercommitted strategy.



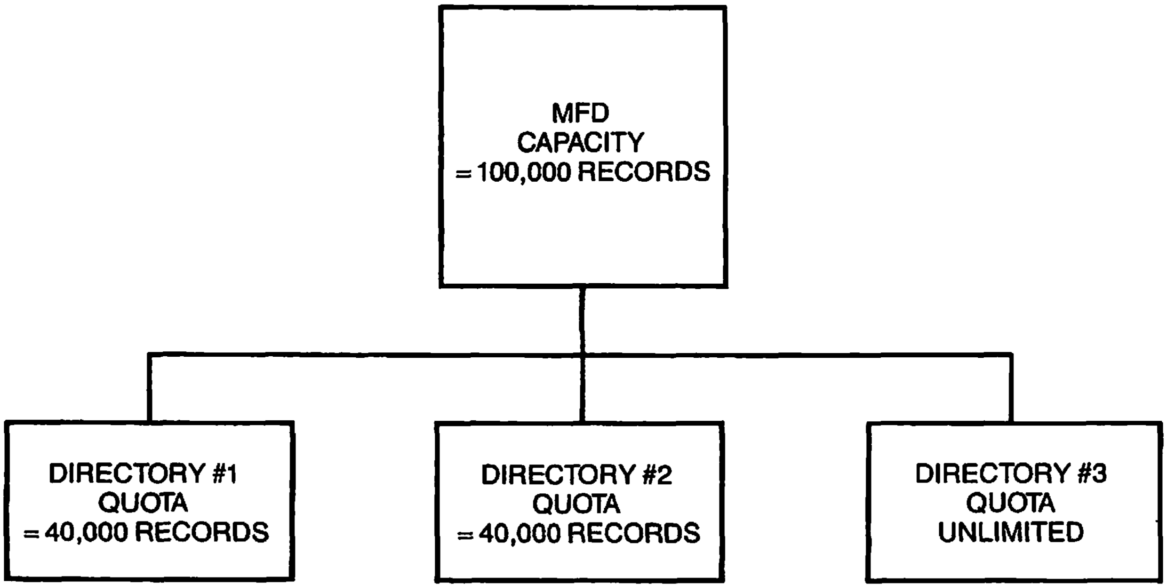
The Undercommitted Strategy
Figure 4-3

The Unregulated Strategy: The least rigid strategy is the Unregulated strategy, where no quota is set on one or more directories. (Setting a quota of 0 on a directory is the same as not setting a quota on it.)

The storage capacity of a nonquota directory is limited only by the physical capacity of the partition. Setting no quota on a directory gives users the impression that their allotment of disk space is unlimited.

You might use the Unregulated strategy if you have a special user group which, by the nature of its work, must be trusted with an "unlimited" amount of disk space. With a 100,000-record partition, two of your directories could each be set at 40,000 records, and the third would have no quota set.

Figure 4-4 illustrates the Unregulated strategy.



The Unregulated Strategy
Figure 4-4

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.

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 Prime User's Guide and Chapter 17 of this guide, SYSTEM MONITORING.

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 Prime 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 Prime User's Guide for explanations and illustrations of how to calculate storage availability.

Recovering From Quota Overloads: If you try to store data that will cause a quota to be exceeded, PRIMOS returns the message "Maximum quota exceeded" and does not allow you to store the material.

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

Using Quotas to Speed Up the LD and LIST_QUOTA Commands

The performance of the LIST_QUOTA and LD commands can be improved significantly by placing a quota on top-level directories. A quota causes PRIMOS to maintain up-to-the-minute quota information. Quota information is therefore readily available and does not have to be collected each time the LIST_QUOTA and LD commands are issued. 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.

MAGNETIC TAPE DRIVES

The SETMOD command determines how tape drives are assigned. By using this command, you determine if users can assign tape drives from their terminals or whether they must ask the operator.

The SETMOD command has the following 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 changes the default state. When the system is in this mode, the ASSIGN command channels 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. Use this format if you do not want users in 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 in 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.

5
Security

Because no two computer installations have exactly the same security requirements, you are responsible for the security of the system. This chapter provides some guidelines to help you establish this security.

SECURITY FOR YOUR SYSTEM

Computer security consists of hardware security and software security.

Hardware Security

Hardware security consists of security for the physical plant and security for the equipment. Security for the physical plant , which is discussed in greater detail in Chapter 13, EQUIPMENT AND ENVIRONMENT, includes the following:

- Controlling access to the computer room
- Setting up maintenance schedules
- Seeing that measures are taken to ensure the physical safety of the machines, their operators, and their users

Security for the equipment (which includes terminals, printers, and modems) involves keeping track of its use outside the computer room. To ensure security for the equipment, use the following guidelines.

- Keep an up-to-date inventory of all equipment. Each item entry should include a brief description, serial number, and current location. You may want to keep a separate inventory of tapes and disks.

- Label all portable equipment by using indelible ink or by engraving it.

- Set up procedures to control the movement of equipment. This is especially important if equipment is sometimes taken out of the building or off-site. Someone in authority should always know where any piece of equipment is at any given time.

Software Security

Software security consists of security against illegal access to the system itself and security against illegal access to data after login. Maintaining software security requires making two main types of decisions:

- How to control access to the system itself, so that unauthorized users cannot log in and use the system (called login security or system access)

- How to control a user's access to files and directories after login (called data security or data access)

The PRIMOS operating system allows you to implement login security through the User Profile system, and to implement data security through Access Control Lists (ACLs). The following comparison between these controls and the previous password system of security control demonstrates the superiority of the present system.

Security Control by Passwording

Before Rev. 19, both login security and data security were handled by passwording. Although it provided a measure of security, the password system as implemented had some drawbacks:

- There was no default protection mechanism.

- Every user had the same inherent rights to the system. These rights were alterable only one file or directory at a time and required active intervention by the creator of the file or directory.

- Login and data security were rudimentary because passwords had to be communicated — by word of mouth, in writing, or in source code — to anyone who needed to use them.

Starting at Rev. 19, the User Profile system controls login security (system access) and Access Control Lists (ACLs) provide data security (data access).

The ACL security system has the following advantages:

- Default protection can be supplied, both for the system and for individual files and directories.
- Default is closed (that is, no access).
- Access rights are set on a user-by-user basis. Thus, every user has a set of specially tailored access rights. No two users need have the same rights.
- Access to the system is controlled by a single person, the System Administrator.
- Access to data can be controlled by a single person, either by the owner (creator) or by an external administrator.
- After access controls are set on a file or directory, no password or other transferable information is required for a guest user to use the data.
- A password can be a requirement for login. This password can be different for every user.
- Passwords are recorded in the machine in an encrypted (scrambled) form, so that they cannot be read by humans or be easily decrypted.
- If required, passwords can still be set on directories.

LOGIN SECURITY

Starting at Rev. 19, the User Profile system provides login security. The User Profile system consists of a data base that you build with the EDIT_PROFILE utility. (EDIT_PROFILE is explained in Chapter 12 , USING EDIT_PROFILE.)

The User Profile Data Base includes an entry for every authorized user. Each user entry is a set of tables mapping the user ID, the login password, and the project (or projects) to which this user ID is allowed access. When a user attempts to log in, PRIMOS consults this data base and determines if the user should be allowed access to the system. If the user is allowed access, the data base also contains the user's system and project access rights.

In addition to the User Profile system, you can write an external login program that controls where, when, and how a user can log in. If you have written external login programs for pre-Rev. 19 systems, you should convert them to take advantage of the more powerful security features available after Rev. 20. For more information on external login programs, see Appendix A, EXTERNAL LOGIN AND LOGOUT PROGRAMS.

User IDs

A user ID must be registered in the User Profile Data Base. See Chapter 3, PLANNING THE USER ENVIRONMENT, for the rules governing user IDs.

Your system is more secure if every person has a unique user ID. You may decide, however, that a group of people may use the same user ID. People in such a group share the same system restrictions and privileges. Allowing several people to share one user ID decreases security, but this may be offset by the simplicity of providing an identical operating environment for many people in a single operation.

For optimum security, allocate user IDs that are not the given names or initials of your users. IDs that are given names or initials are less secure than IDs that are not as obviously associated with a specific user.

If you are attached to a network, you may want user IDs to be unique not only in your home system but also in the entire set of systems that access your system regularly. This includes not only the systems attached through the PRIMENET network, but also any other system that regularly accesses your system through PRIMENET or a Public Data Network (PDN). The EDIT_PROFILE command VERIFY_USER allows you to determine if and where a user ID is duplicated within your system pool.

Login Passwords

At login, a user must supply a login password. The two questions you must decide about how login passwords are to be used on your system are the following:

- Will null passwords be allowed?
- Will password echoing be allowed?

These two issues are discussed in the next two sections. See Chapter 3, PLANNING THE USER ENVIRONMENT, for the rules governing login passwords.

Setting Requirements for Passwords: If you use the EDIT_PROFILE NO_NULL_PASSWORD -OFF command, you allow users to have null passwords. In this case, no password is entered as part of the login procedure.

Allowing null passwords decreases security by a level. If users do not have to supply passwords, an unauthorized user can gain access to your system more rapidly, especially if all your user IDs consist of the given name or initials of the users. (Even if the project ID has to be supplied, there is a strong possibility that a clever or informed interloper can guess it correctly.) However, if every user has a unique password, which is known only to the user and changed at irregular intervals, it is much harder for an interloper to guess correctly all the parts of the login entry procedure.

For maximum security, it is recommended that you prohibit null passwords by using the EDIT_PROFILE command NO_NULL_PASSWORD -ON command (the -ON option is the default).

In addition, you can also use the MINIMUM_PASSWORD_LENGTH command of EDIT_PROFILE to establish a minimum length for login passwords. For example, you can require that all users have passwords that are at least six characters long.

Requiring Non-echoing Entry of Passwords: If a non-null password is required, the user can enter it in one of two ways:

- The user logs in by typing "LOGIN user-id password". Typing the password on the same line as the LOGIN command means the password is echoed (this is, appears) on the screen.

- The user logs in by typing only "LOGIN user-id". The password is omitted from the login line. In this case, PRIMOS prompts for the password. The password is not echoed, and is thus not displayed on the screen.

The non-echoing method of entering passwords is more secure because another user cannot discover the password by looking at the screen.

It is recommended that you enable the non-echoing of passwords by using the FORCE_PASSWORD command of EDIT_PROFILE. When FORCE_PASSWORD is in effect, a user who enters a password as part of the LOGIN command line receives an error message and is refused entry to the system.

Note

When users log in at half-duplex terminals, passwords are echoed, whether or not FORCE_PASSWORD has been enabled.

Suggesting That Users Change Passwords: The CHANGE_PASSWORD command allows users to change their login passwords at any time. Your system gains an additional measure of security if you encourage all users to change their passwords immediately after their first login. Changing the password at first login ensures that only one person (the person who performed the change) knows the password. In addition, you may want to encourage users to change their passwords periodically.

Passwords are held by the system in an encrypted form. They cannot be called out and read by anyone. A password changed for security reasons should not be written down or told to anyone. If it is, the security provided by password encryption is lost.

If users forget their passwords, the System Administrator cannot find out what the passwords were. The only remedy is to use EDIT_PROFILE to assign new passwords.

CHANGE_PASSWORD is documented in the Prime User's Guide and in the PRIMOS Commands Reference Guide.

Project IDs

Every user must be registered in the User Profile Data Base as a member of at least one project. During a terminal session, a user must be associated with a project ID. At login, the project ID may be supplied by the user or by the PRIMOS internal login program (the program obtains it from the User Profile Data Base).

You can provide a default system project for users who have no true project affiliation. If your system uses projects to provide special operating environments, you may require that users specify project IDs at login. If your system does not use projects, you must set up a default system project, as explained in Chapter 12, USING EDIT_PROFILE. All users become members of the default project.

User IDs With Multiple Projects: A user ID may be a member of several projects, each of which gives a different set of access rights and restrictions. The maximum number of projects with which a user ID can be associated is the number of projects on the system. A system can have a maximum of 4096 projects, which means that a user potentially could be a member of 4096 projects.

When a user is a member of more than one project, the user can specify a particular project on the command line by using the -PROJECT option of the LOGIN command. For example, suppose that user JOE is associated with two projects, ALPHA and OMEGA. The command line

```
    LOGIN JOE -PROJECT ALPHA
```

logs user JOE to the project ALPHA, while the command line

```
    LOGIN JOE -PROJECT OMEGA
```

logs in the same user (or another user using the same user ID) to project OMEGA.

A user who is a member of several projects and who does not specify a project ID at login is assigned to the default project indicated in his or her user profile. If the user profile does not contain a default project, the user is prompted for a project ID. Requiring that users provide project IDs at login adds another level of security to your system.

Degrees of Login Security

As the preceding discussion suggests, requiring only a user ID for login provides the least security. Requiring a user ID, a long non-null password, and a project ID provides the most security.

Network Security

For information on setting up security over a network, see the Network Planning and Administration Guide.

The Login Server

At Rev. 20.2, all terminal login attempts for local and remote users that were previously handled within PRIMOS are handled by the Login Server. Unlike other system servers, the Login Server does not require server support functions and does not require you or a user to intervene. The Login Server starts when the system is booted. If, for some reason, the Login Server does not start or stops after it has been started, you can enter the system command START_LSR at the supervisor terminal to start it. (Refer to the Rev. 20.2 update to the Operator's Guide to System Commands for a description of this command.)

The user interface that the Login Server provides to PRIMOS at login or logout is the same as PRIMOS provided prior to Rev. 20.2. The Login Server provides the following services for a system:

- Receives input from all terminal lines on which a user can log in
- Responds to all commands that can be issued on such lines without logging in, for example, the DATE command
- Processes and validates login requests
- Passes remote login requests to the appropriate remote node

- Arranges to associate a user who has logged in successfully with a terminal process and also obtains terminal buffers for that user

- Reassumes control of the line that becomes inactive when a user logs out

The Login Server has no interaction with users who are logged in and who have had their lines associated with a process or other server except when a user who is logged in attempts to login again. When this happens, the Login Server is called upon to read the SAD and to validate and initialize the user. Users cannot explicitly call the Login Server.

The Login Server accepts the following commands from a logged-out line:

Command	Description
DATE	Displays the current calendar date and clock time.
DELAY	Defines a time function that delays printing a character after a carriage return (RETURN) has been output to the terminal.
DROPDTR	Drops the DTR (Data Terminal Ready) signal associated with a terminal line.
LOGIN	Admits a user onto the system.
USRASR	Allows the system console to function as a user terminal.

For a detailed description of these commands, refer to the PRIMOS Commands Reference Guide.

As the System Administrator, you need to know the following about the Login Server:

- You must configure (with the NPUSR directive described in Chapter 10, CONFIGURATION DIRECTIVES) a phantom for the Login Server.

- When you execute the STATUS USERS command, the Login Server is listed. The Login Server runs under the name LOGIN_SERVER and its process type is listed as LSr.

- The LOGIN_SERVER.RUN file and LOGIN_SERVER.ENTRY$.SR file are installed in the new Prime-supplied system directory SERVERS* that requires $REST:LUR access rights. See Chapter 9, SETTING SYSTEM ACCESS.

- At cold start, ACLs on each directory in the SAD are set automatically to allow the Login Server to access them. Do not remove these ACLs, which are preserved by EDIT_PROFILE. Also you do not have to create an entry for the Login Server in the SAD.

- Executing the new supervisor terminal command START_LSR starts the Login Server, and executing the new supervisor terminal command STOP_LSR stops the Login Server. (See the Rev. 20.2 update to the Operator's Guide to System Commands for a description of these commands.) If you enter START_LSR while the Login Server is running, the supervisor terminal displays a message that the system cannot spawn the Login Server. You cannot stop the Login Server with the LOGOUT -n command.

- If the Login Server stops, it prints the following message to all logged-out terminals:

  ```
  Logins are blocked -- Login Server is logged out. (lsr)
  ```

  If the STOP_LSR command shuts down the Login Server, the supervisor terminal also displays a phantom logout message. The Login Server logs out under the user ID under which it was running. If an internally detected error causes the Login Server to stop, the supervisor terminal displays an error message. Users who try to log in while the Login Server is down receive no messages at their terminals.

- If the Login Server logs out abnormally after it is started and users cannot log in, a search rules problem may be indicated. Check that all entries in the SEARCH_RULES*>ENTRY$.SR file are on the command device, that all pathnames are correct, and that the ENTYRY$.SR file contains no typographical errors. From the supervisor terminal, fix the search rules and start the Login Server with the START_LSR command.

The Login Procedure

PRIMOS responds to a LOGIN command line as follows:

1. The Login Server checks the supplied user ID, password, and project ID in the system data base to verify that the person attempting to log in is an authorized user of the system. (If the user fails validation, the login procedure terminates and

Steps 2-4, described below, are not performed.) PRIMOS also establishes the user's membership in ACL groups that are active during this session.

If the configuration directive LOGBAD YES is in the configuration file, a message about the unsuccessful login is printed at the supervisor terminal. If LOGBAD is not enabled, no message is displayed when the LOGIN process fails.

2. PRIMOS searches for a site-supplied external login program and executes it if it exists. If the program does not exist, PRIMOS proceeds to Step 3.

   The external login program must be a static-mode program named LOGIN (no suffix is allowed) and must reside in CMDNC0. This program may perform further validation tests for login and, if the user fails these tests, may log out the user. The program may also perform other operations, such as executing an accounting program. For more information on external login programs, see Appendix A, EXTERNAL LOGIN AND LOGOUT PROGRAMS.

3. PRIMOS attaches the user to the user's origin directory.

4. PRIMOS searches the user's origin directory for a user-supplied login program (named LOGIN.RUN, LOGIN.SAVE, LOGIN.CPL, or LOGIN.COMI) and runs the program if it exists. A login program further constructs the user's environment by performing such tasks as enabling a global variable file and/or an ABBREV file, setting terminal characteristics, and executing other programs or commands. For more information on user login programs, see the Prime User's Guide. The user is now logged in and ready to work on the system.

Examples of User Validation at Login

The steps that PRIMOS takes for user validation at login (Step 1 above) depends on what information the user supplies on the command line.

The next three sections contain examples that illustrate how the system, through the internal login program, responds to different combinations of user information. All examples refer to the data base illustrated in Figure 5-1 and assume that null passwords and the entry of passwords on the command line are allowed.

Example 1: Full information is supplied on the command line. The command is

```
LOGIN FROG GREEN -PROJECT SWAMP
```

The system does the following:

1. Locates the user ID FROG in the system data base.

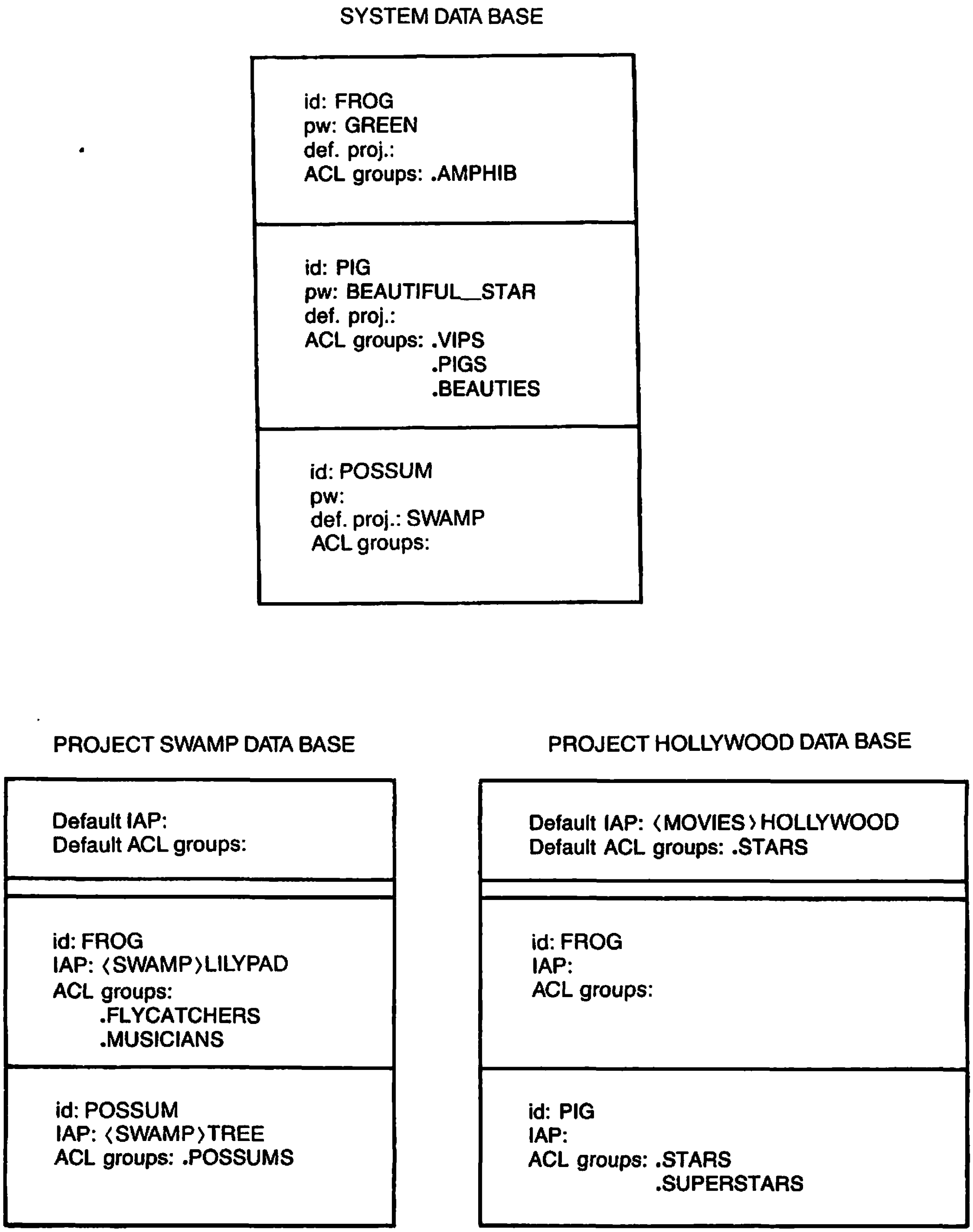
2. Verifies that GREEN is the password associated with FROG.

3. Checks the project data base for project SWAMP and finds FROG listed as a member.

4. Checks project SWAMP's data base for FROG's origin directory. The directory is <SWAMP>LILYPAD. The system attaches FROG to that directory.

5. Checks the system data base for systemwide ACL groups for FROG. It finds one -- .AMPHIB -- and marks FROG as a member of that group.

6. Checks project SWAMP's data base for project-specific ACL groups for FROG. It finds two (.FLYCATCHERS and .MUSICIANS) and adds them to .AMPHIB to create the list of ACL groups for FROG for this session.



SYSTEM DATA BASE

id: FROG
pw: GREEN
def. proj.:
ACL groups: .AMPHIB

id: PIG
pw: BEAUTIFUL__STAR
def. proj.:
ACL groups: .VIPS
.PIGS
.BEAUTIES

id: POSSUM
pw:
def. proj.: SWAMP
ACL groups:

PROJECT SWAMP DATA BASE

Default IAP:
Default ACL groups:

id: FROG
IAP: ⟨SWAMP⟩LILYPAD
ACL groups:
.FLYCATCHERS
.MUSICIANS

id: POSSUM
IAP: ⟨SWAMP⟩TREE
ACL groups: .POSSUMS

PROJECT HOLLYWOOD DATA BASE

Default IAP: ⟨MOVIES⟩HOLLYWOOD
Default ACL groups: .STARS

id: FROG
IAP:
ACL groups:

id: PIG
IAP:
ACL groups: .STARS
.SUPERSTARS

Sample Portion of User Profile Data Base
Figure 5-1

Example 2: Only the user ID is supplied on the command line. The command is

```
LOGIN POSSUM
```

The system does the following:

1. Finds the user ID POSSUM in the system data base.

2. Checks for POSSUM's password, and finds that the password is null. Because the password is null and because POSSUM is a valid user ID, the system does not prompt for the password.

3. Checks to see whether POSSUM has a default project, and finds that it is project SWAMP. Therefore, the system does not prompt for a project ID.

4. Checks project SWAMP's data base and finds POSSUM listed as a member with <SWAMP>TREE as his origin directory.

5. Attaches POSSUM to <SWAMP>TREE.

6. Finds no systemwide ACL groups for POSSUM in the system data base.

7. Checks project SWAMP's data base and finds that POSSUM is a member of the ACL group .POSSUM. .POSSUM becomes POSSUM's only ACL group for this session.

Example 3: No information is supplied on the command line, only the LOGIN command. The user later gives an incorrect ID. The command is

```
LOGIN
```

The system does the following:

1. Prompts for a user ID. The user responds: ORK.

2. Checks the User Profile data base and does not find ORK there.

3. Prompts for a password. ORK responds: FABLE.

4. The system terminates login and prints the error message, "Invalid user id or password; please try again."

Note that the system always prompts for a password when none is given, even if an invalid user ID has been supplied. This method increases login security because the user does not know whether the login rejection was caused by an invalid user ID or by an invalid password.

DATA SECURITY

Data security is the control of a user's access to data after login. PRIMOS provides three methods to protect the information contained in files and directories:

- Access Control Lists (ACLs) for ACL-protected directories
- Priority ACLs for partitions
- Passwords and the PROTECT command for password-protected directories

Although users can still employ passwords to control access to directories, the ACL system is recommended because it is safer and easier to use.

Access Control Lists

Access Control Lists (ACLs) are the cornerstone of the file system access control mechanism. Users with Protect access can protect their files and directories with ACLs.

ACLs can be set on a directory or a file. If an ACL is set on a directory, all file system objects contained in the directory are given the same protection by default. This default protection can be overridden by setting a specific ACL on a lower level file or directory with the SET_ACCESS command.

An ACL can provide access control both on a user basis and on a group basis. Both types of control can be combined in one ACL. An ACL can also use the $REST identifier to control access rights for all users who do not appear in the ACL by name or as group members.

When a user is included in an ACL both as a member of an ACL group and as a named user, the rights for the named user override the group rights. Thus, the user receives only those rights assigned by the user ID. This control can be used to either increase or decrease the rights of the named user.

After you have defined a group in the User Profile data base, any user can use that group name in an ACL. If a user is a member of more than one ACL group and the groups are in an ACL, the user receives the sum (logical union) of all access rights for those groups.

Note

> Users can use nonexistent group names in ACLs. Adding the nonexistent group does not achieve anything, however, because no members have been defined for the group. The rest of the ACL remains valid.

Because users' needs for ACL groups may change frequently, you may want to set up a mechanism for adding groups to the system or for changing the membership of groups.

By using the proper combination of ACLs, you can also prevent the unauthorized copying of licensed programs. The X (Execute VMFA) access right prevents local EPFs from being copied or read with a standard file system utility, but allows them to be executed.

For a discussion of ACLs in general, see Chapter 3, PLANNING THE USER ENVIRONMENT, or the Prime User's Guide. For guidelines to setting system-level ACLs (including an explanation of X access), see Chapter 9, SETTING SYSTEM ACCESS.

Priority ACLs

Whether your system is using ACLs, passwords, or a combination of both, you can set priority ACLs to govern access to any given partition on the system. Priority ACLs override all other data security mechanisms in PRIMOS. They are generally intended for temporary use, such as when you need to back up a partition. For a full discussion of priority ACLs, see Chapter 9, SETTING SYSTEM ACCESS.

Passwords

PRIMOS allows users to create directories as ACL directories or password directories. Access to password directories is controlled by owner and non-owner passwords.

Using password directories is generally not recommended because they are less secure than ACL directories. Because users must include the password when accessing the directory (for example, with the ATTACH command), the password can be discovered when it appears on a user's screen or in a user-written program that must attach to the directory.

Owners of password directories use the PASSWD command to change the directory's passwords and the PROTECT command to set access rights on the directory's files and subdirectories.

For details on setting protection on password directories, see the Prime User's Guide.

COORDINATING LOGIN AND DATA SECURITY

Because login security and data security can both be handled through the User Profile Data Base, the two can be coordinated easily. In particular, the use of projects and of project-based ACL groups provides a greater degree of security on the system.

The three general systems of security control are as follows:

- A loosely controlled system, with very little security at the system level. An example might be a system used by a small business, where all users have access to most of the data.

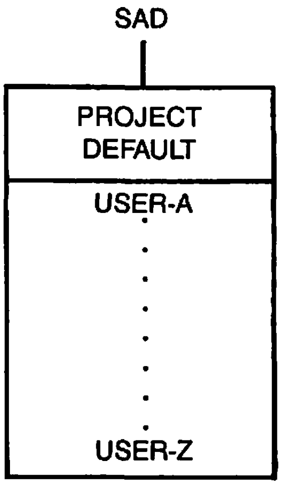
- A tightly controlled system, with secure ACLs at the system level. An example might be an applications development group, where full access to any given set of files is restricted to a small set of people.

- A mixed system, which combines tight security on some projects (and for some users) with looser security for other users. An example might be a college, where it would be desirable to give one set of users (the faculty) greater access and privilege than would be given to another set of users (the students).

Loosely Controlled Systems

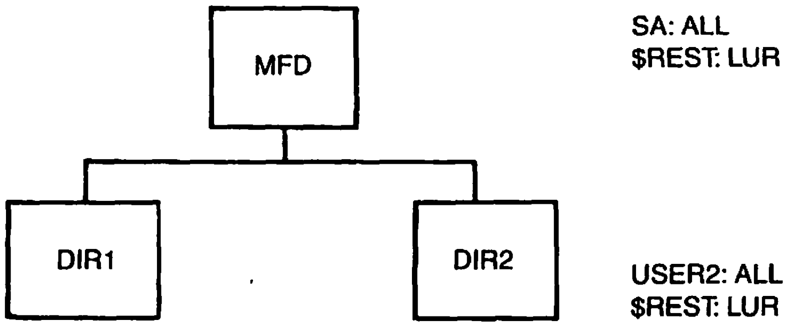
A loosely controlled system provides a level of control roughly analogous to the pre-Rev. 19 password system. Users are provided with user IDs, and with automatic membership in project DEFAULT. Users may also be grouped into systemwide ACL groups, as needed.

Figures 5-2a and 5-2b diagram a loosely controlled system.



Loosely Controlled System
Figure 5-2a

All users belong to project DEFAULT. ACLs can govern the rights of individuals or groups at any level of the file hierarchy, from the MFD to an individual file. No other projects are in use. No Project Administrators are required.



ACLs on a Loosely Controlled System
Figure 5-2b

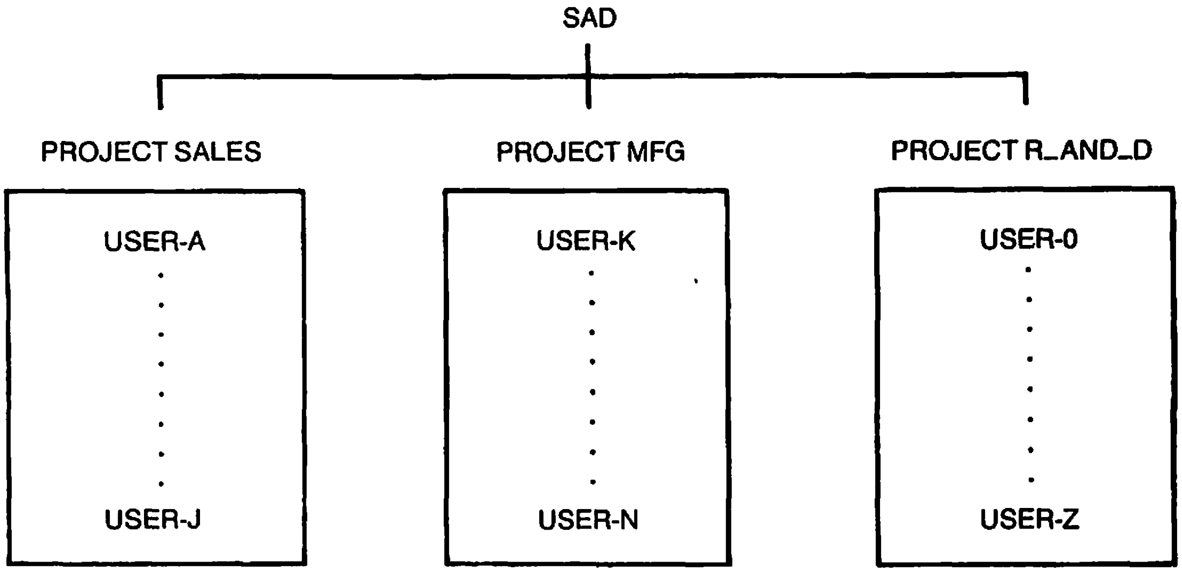
Tightly Controlled Systems

A more tightly controlled system uses projects and project attributes, in addition to the system-based attributes used in the loosely controlled system shown above.

When projects are used, every user can be required to log in with a project ID, in addition to a user ID and password. The project entry point and ACL controls can effectively restrict the user to a small subset of the whole system, without requiring specific ACLs to be set on specific file system objects.

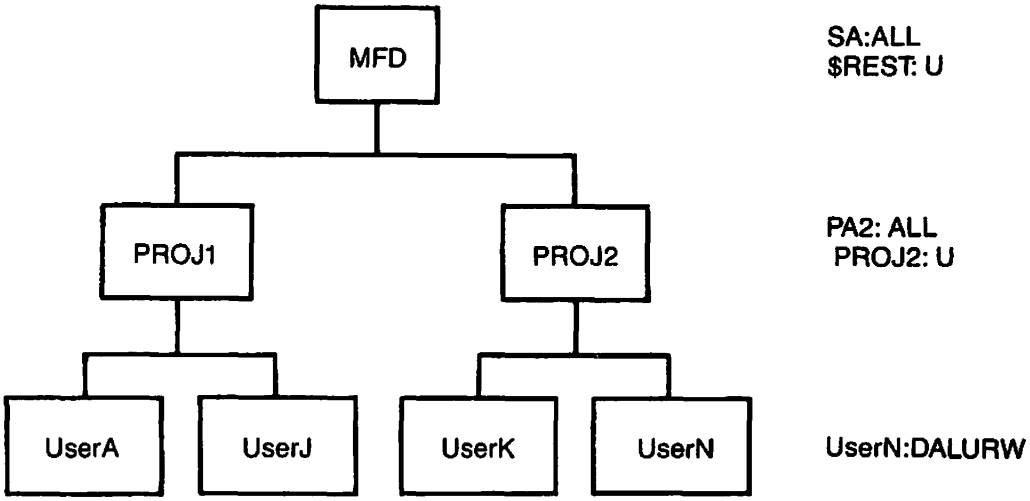
In a project-based system, you do not have to use project DEFAULT. However, if you think that you may use this project at any future time, you must set it up during initialization. Project DEFAULT is treated differently from other projects and cannot be added to the data base at a later time, as normal projects can. Project DEFAULT can be used to accommodate users who are not included in any formally established project. (Keep in mind that users cannot log in unless they belong to at least one project.)

Figures 5-3a and 5-3b diagram a tightly controlled or project-based system. In this example, project DEFAULT is not in use.



Tightly Controlled or Project-based System
Figure 5-3a

Project DEFAULT does not exist. Every user must log in to a specific project. Project-based ACL groups can be used to deny non-project members access to project files and directories. Users may belong to several projects, but their access rights during any session depend on the project ID they specified at login time. Project Administrators may perform limited security functions.



ACLs on a Tightly Controlled System
Figure 5-3b

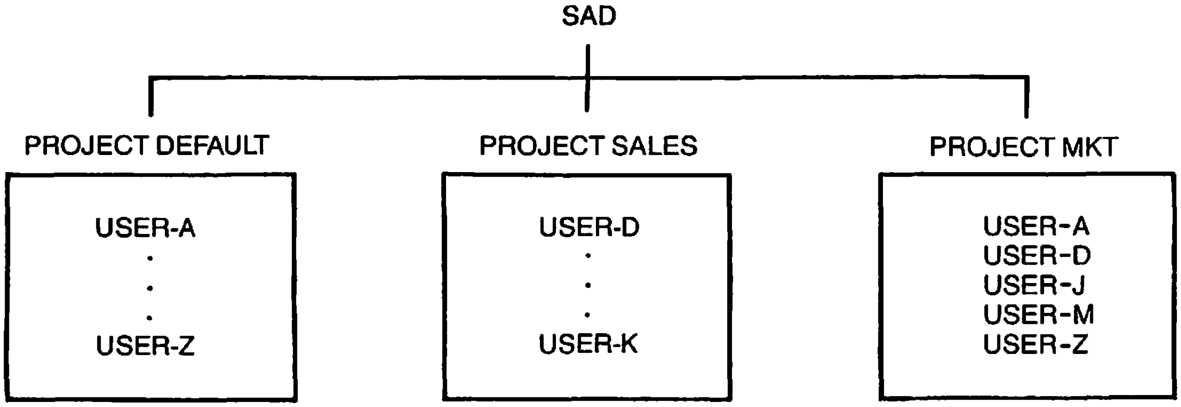
Mixed Systems

A mixed system provides different degrees of privilege for different users. You accomplish this by setting up one level of security as the system default, and then using specific projects to grant greater or lesser privileges to project members. A mixed system can be set up in one of two ways:

- Some users belong to project DEFAULT and are given wide rights throughout the system. All other users belong to specific projects and their rights are restricted to these projects. Under this scheme, members of project DEFAULT have wide rights while all others have limited rights.

- All users belong to project DEFAULT. Some users also belong to other projects. ACLs give members of these other projects sole access to project-specific directories. Under this scheme, membership in project DEFAULT confers standard rights. Other projects provide extra rights for their members.

Figure 5-4 diagrams a mixed system.



Mixed System
Figure 5-4

All users belong to project DEFAULT. Users with special privileges belong to other projects as well. Systemwide ACL groups tend to offer minimal access (such as LUR rights). Project-based ACL groups tend to offer wider rights.

6 Setting Up Subsystems

This chapter discusses the guidelines for setting up the Spooler subsystem and the Batch subsystem. Some of the other Prime subsystems available, and the documents that describe them, are the following:

● DBMS	DBMS Administrator's Guide (DOC6292-192, UPD6292-12A, and UPD6292-13A) and ROAM Administrator's Guide (DOC7345-2LA and UPD7345-21A)
● DPTX	Distributed Processing Terminal Executive Guide (DOC4035-193 and UPD4035-21A)
● FTR	Network Planning and Administration Guide (DOC7532-2LA)
● Prime INFORMATION™	Prime INFORMATION Administrator's Guide (DOC10065-1XA)
● OAS	OAS Administrator's Guide (DOC6781-040 for PT45 and PT65, DOC10017-1LA for PST 100 and PT200)
● PRIMENET	Network Planning and Administration Guide
● PRIME/SNA	PRIME/SNA Administrator's Guide (DOC8908-2LA)
● PRIMIX	Using PRIMIX on the Prime 50 Series (DOC9709-2LA)
● RJE	Remote Job Entry Phase II Guide (DOC6053-3LA)

THE SPOOLER SUBSYSTEM

The following sections outline some of the decisions you must make before setting up your Spooler subsystem. For detailed information, see the Operator's Guide to the Spooler Subsystem.

Security on the SPOOLQ Directory

One way to provide security for the SPOOLQ directory is to create it as a password directory. You can then change the password of the directory by using the CHG_PWD.CPL program in the directory SPOOL. When you invoke the program, it prompts you for the new password (which must be six characters in length), modifies the source program, and rebuilds the Spooler subsystem. The subsystem must then be reinstalled and reshared.

SPOOLQ Quota

The SPOOLQ directory not only stores the files that control the printers but also holds temporary copies of spooled files. (SPOOLQ can hold up to 200 spooled files, depending on the ordering of the PRT numbers.) If SPOOLQ exceeds its quota and runs out of records, files cannot be copied into it. Users, therefore, cannot spool files. If they attempt to do so, they receive error messages.

To ensure that SPOOLQ does not run out of space, place a high quota on it. Even if you do not place a quota on SPOOLQ itself, you may want to place quotas on other UFDs that share the partition with SPOOLQ, to ensure that SPOOLQ gets a certain minimum number of records.

Multiple Local SPOOLQ Directories

A system can have more than one SPOOLQ directory. Each SPOOLQ directory, however, must be on a separate local partition.

If you have only one spool queue and your users are spooling files frequently, they may fill the SPOOLQ limit of 200 files. When this limit is reached, files cannot be spooled until space is created. If users are complaining that this situation occurs regularly, you may want to create an extra SPOOLQ directory on another local partition. Users can then spool to this partition by using the -DISK option of the SPOOL command.

If you have two or more local spool queues, your printer environments must have the correct settings to access these extra queues. These settings are in the UPPER, LOWER, and QSCAN parameters.

Scanning Multiple Spool Queues

If your spool phantom scans multiple queues (both local and remote), you may want to change the QSCAN parameter of its environment. (The queues that the phantom scans are those in the range specified by the LOWER and UPPER parameters of its environment.) The default setting of the QSCAN parameter is 0 but can be changed to 1.

If the QSCAN parameter is 0, the spool phantom first prints all files in the local spool queue. When the queue is empty, the phantom checks the next queue and prints a file if one is found. (The second queue is on the next partition on the disks list that has a SPOOLQ directory. Use the STATUS DISKS command to display the disks list.) The phantom then checks the first queue and prints any files found. Only when both the first and second queues are empty does the phantom check any more queues.

To make the phantom access all queues equally, change the QSCAN parameter of the phantom's environment to 1 by using the QSCAN subcommand of the PROP utility.

Accessing Networked Spool Queues

If you have several computers networked, you must decide if they will read each other's queues. Such access is controlled for each printer by the UPPER and LOWER parameters in the printer's phantom environment. (For details, see the Operator's Guide to the Spooler Subsystem.)

This access also requires that each system allow its disks to be read by the others, as explained in the Network Planning and Administration Guide.

Access to Printer Environments

You can create one or more files in the SPOOLQ directory that specify which users (other than users logged in under the user ID SYSTEM) can do the following:

- Modify and control printer environments with the operator form of the PROP command

- Modify files in the spool queue with the SPOOL -MODIFY command

The files are the L.USER file and the L.environment files. If none of these files exist, only users logged in under the user ID SYSTEM have the above privileges.

For details on the PROP and SPOOL -MODIFY commands, see the Operator's Guide to the Spooler Subsystem.

L.USER File: The L.USER file contains the names of users who can control any printer environment on your system, except for those printers that have L.environment files. With the operator form of the PROP command, these privileged users can create printer environments. They can also modify, start, stop, and delete any existing environment in SPOOLQ, as long as that environment does not have a corresponding L.environment file.

Users in the L.USER file can also use the -MODIFY option of the SPOOL command to modify the characteristics of any spooled file, including other users' files. Only users in the L.USER file can use the -RUSH option of SPOOL.

To create the L.USER file, use a text editor, such as ED or EMACS, and place the file in the SPOOLQ directory. The file contains the user IDs of users who can modify any printer environment. You must enter each user ID on a separate line in uppercase.

L.environment Files: Each printer environment can have an associated file named L.environment, where environment is the name of the printer. This file contains the user IDs of users who can modify the environment for that printer.

If a printer has a corresponding L.environment file, the PROP command does not check the L.USER file, it checks only the L.environment file. If, however, the printer does not have an L.environment file, the PROP command searches the L.USER file.

To create an L.environment file, use a text editor such as ED or EMACS. Each user ID must be in uppercase and on a separate line.

Note

> If a user is listed in the L.USER file but not in a particular L.environment file, the user cannot modify that printer's environment because the PROP command does not read the L.USER file if an L.environment file exists.

Defining Paper Forms

The System Administrator defines the paper types and forms for the printer environments. Use the PAPER subcommand of the PROP command to define a paper type for a printer environment. To define synonyms for a paper type, use the FORM subcommand of PROP. The main difference between a paper type and a paper form is that an environment can have only one paper type defined (even if it is blank), but it can have up to eight forms.

The default paper type for your system should be the type that is used most frequently, so that users do not have to ask for it by name with the -FORM option of SPOOL. To define a default paper type for an environment, do not use the PAPER subcommand when you create the environment. The default paper type for that environment is blank.

You can also use the FORM subcommand to define other forms for that environment. The form names then become synonyms for the paper type and can be used with the -FORM option of the SPOOL command. If you define other forms, inform your users, operators, and spool phantoms of the form names.

If you have defined several paper forms for your environments, enter the form names in a file called L.FORM. The procedure for defining paper forms is as follows:

1. Using the PAPER and FORM subcommands of PROP, define the appropriate paper types and form names in the environment files that control the printers. (See the Operator's Guide to the Spooler Subsystem for details.)

2. Use a text editor (such as ED or EMACS) to create a file named L.FORM and place it in the SPOOLQ directory. Enter into this file all the names of the paper types and forms in the printer environments. Each name must be entered on a separate line, in uppercase.

3. Inform your users of the names of the paper types and forms. Users can use names with the -FORM option of the SPOOL command to specify a particular paper type or form. If the -FORM option is not used, the user receives the default form. (The SPOOL command is explained in the Prime User's Guide and in the PRIMOS Commands Reference Guide.)

When a file is spooled with the -FORM option, the spool phantom checks the L.FORM file for that form. If the form is not in the L.FORM file, the user receives a message warning that the form is not a valid form type for the system.

If you define a paper form for an environment but do not include the form name in the L.FORM file, users still receive the warning message. The spooled file, however, will print if the proper environment is active.

Note

Do not confuse the L.FORM file with the L.TYPE file. The L.TYPE file contains an environment's valid printer types for use with Prime's Office Automation System.

Scheduling Use of Forms

If you have created several environments for a printer, each with a different paper form, you must decide whether to schedule a specific time that each form is mounted. For example, if you have defined forms WIDE and NOTE for a printer, you may want to print WIDE forms from 9 A.M. to 1 P.M. each day and NOTE forms after 1 P.M. The advantage of this method is that the spool queue does not have to be monitored by your operators.

The alternative is to have your operators constantly monitor the spool queue. When they see a number of requests for one form waiting in the spool queue, they can stop the current environment, change the paper in the printer, and start the appropriate environment. The advantage of this method is that files on often requested forms are printed sooner.

Under either arrangement, users can submit jobs at any time. The jobs are held in the spool queue until a spool phantom recognizes and accepts the form name given with the -FORM option of SPOOL. Users can tell when their files are printed either by using the SPOOL -LIST command or by using the -NOTIFY option when spooling (-NOTIFY sends the user the message "Printing completed" when the file is printed).

If you decide to mount certain forms on a schedule, you should inform both users and operators of the schedule. Users can then submit jobs close to the scheduled print time, thus reducing the need for extra space in the spool queue.

Designating a Particular Printer

Before users can request output from a particular printer, they must know how to specify the printer with the SPOOL command. If you have several printers, you must decide whether they will be known to users by form names (specified with the -FORM option) or by destination names (specified with the -AT option).

For example, if you have several line printers and one letter quality printer (LQP) reading the same spool queue, you might want users to request letter quality jobs with the following command format:

```
    SPOOL filename -AT LQP
```

If, however, LQP itself sometimes uses cloth ribbon (for in-house work, perhaps) and sometimes uses carbon ribbon (for work going out to customers), users might request the LQP with either -FORM CLOTH or -FORM CARBON.

To use form names for printers, follow the steps in the previous section for defining form names and creating an L.FORM file.

Using Destination Names: To use destination names, you must create an L.DEST file in the SPOOLQ directory. The L.DEST file, which is similar to the L.FORM file, contains the environment names and destination names of your printers.

A destination name is a synonym for the name of an environment. Users can use either the environment name or the destination name with the -AT option of SPOOL. To define a destination name for a printer environment, use the DEST subcommand of PROP.

For details on the L.DEST file and on defining destination names, see the Operator's Guide to the Spooler Subsystem.

Note

> Whether you use form names or destination names for printers, you should have both the L.FORM and L.DEST files in SPOOLQ.

Delivery of Printouts: Another decision you must make is whether users should pick up their own printouts, or the operators should deliver printouts to some specified place or places outside the computer room. (You can also use both methods.)

If printouts are to be delivered to more than one place, you should create a separate destination name for each location. Users can then specify this location with the -AT option.

Using EVFU

Some printers define the length of forms with the Electronic Vertical Format Unit (EVFU), rather than a form length switch and/or a paper control tape loop. If your site has EVFU printers, you must determine whether the printers will use default EVFU forms or special EVFU forms.

If you use default forms, you must use the EVFU -ON subcommand of PROP in the environment so that the default form is properly invoked by the spool phantom.

If you need special forms, you must do the following:

1. Using a text editor, create a EVFU format file.

2. Modify the printer's environment with the PROP subcommand EVFU -NAME filename, where filename is the name of the EVFU file.

For details on creating EVFU files, see the Operator's Guide to the Spooler Subsystem.

THE BATCH SUBSYSTEM

The Batch subsystem makes phantom execution of jobs easier for the user, while giving the System Administrator and operators greater control of the environment and of job execution.

The Batch Administrator defines from one to sixteen Batch queues from which user jobs can run as phantoms. These phantoms should run at lower priorities than interactive jobs. Thus, the batch phantoms use smaller amounts of CPU time when interactive use is heavy, but larger amounts when interactive use is light or absent. Batch jobs may also be held in their queues by operators, and released to run at appropriate times. For example, time-consuming jobs (such as file updates and backups) can be set up as batch jobs during the day, then run under operator control at night.

System Administrator's Batch Responsibilities

As System Administrator, your responsibilities for the Batch subsystem are as follows:

- Designate someone as Batch Administrator to create and maintain the Batch subsystem. You may also serve as Batch Administrator.

- Help the Batch Administrator to decide on the number and the characteristics of the Batch queues.

- Ensure that Batch queues are created and added to the subsystem in the proper order.

- Assign (using EDIT_PROFILE) at least two command levels to any user who is going to use Batch.

- Set up enough phantoms for Batch to run (by specifying a sufficient number for the NPUSR directive in the system configuration file).

- Add the BATCH -START command to the PRIMOS.COMI file to bring up the Batch subsystem at cold start.

- Initialize the Batch subsystem (with the INIT program in the BATCHQ directory) after Rev. 20.2 software is installed.

- Decide whether to change the Batch data base password (and if so, run the Batch build and install program).

- Modify the Batch monitor startup file to remove old inactive jobs or to prevent job messages from displaying at the supervisor terminal.

- Ensure that either the FIXBAT or the INIT utility is run to repair or replace a damaged Batch data base.

The first five items, as well as a brief explanation of how the Batch subsystem works, are described below. For full details on the System Administrator's responsibilities for the Batch subsystem, see the Operator's Guide to the Batch Subsystem.

Prerequisites for a Batch Subsystem

To configure a Batch subsystem, your system must have the correct version of PRIMOS and enough phantoms and user file units.

PRIMOS: Rev. 20 or greater PRIMOS requires Rev. 20 Batch. Rev. 20 Batch cannot work correctly with a pre-Rev. 20 PRIMOS BATDEF file (which contains queue definition information) and queue files.

If you are converting to Rev. 20.2, you must initialize the Batch data base using the INIT program in the BATCHQ directory.

Note

> When Rev. 20 or greater is installed onto an existing pre-Rev. 20 system, the pre-Rev. 20 BATDEF file is overwritten. Therefore, save the old BATDEF file (which will help you in setting up the new one) before installing Rev. 20.2.

Phantoms: The Batch subsystem requires one phantom (which runs under the name BATCH_SERVICE) to control the Batch monitor exclusively.

User File Units: Because the Batch subsystem requires a minimum of 16 file units for each user, the FILUNT directive in the CONFIG file must be set at 16 or higher. If the FILUNT directive is set too low, Batch prints error messages at the supervisor terminal requesting that the directive be changed. For details on the FILUNT directive, see Chapter 10, CONFIGURATION DIRECTIVES.

Format of Batch Queues

Each Batch queue is a separate entity, defined by the Batch Administrator to be particularly hospitable to certain types of jobs. Each queue has a set of characteristics and a status.

A queue's characteristics consist of the following nine parameters:

- Name
- Default CPU time limit

- Maximum CPU time limit
- Default elapsed time limit
- Maximum elapsed time limit
- Default PRIMOS file unit for command input
- Default value for priority of job within queue
- Relative runtime priority
- Time slice

A queue's status is a combination of the following:

- Active or inactive (set by BATGEN's ACTIVE_WINDOW subcommand)
- Blocked or unblocked (set by BATGEN's BLOCK and UNBLOCK subcommands)
- Capped or uncapped (set by BATGEN's CAP and UNCAP subcommands)

The Batch Administrator creates queues and defines their characteristics by using the BATGEN command (explained in the Operator's Guide to the Batch Subsystem). The Batch Administrator (or operator) also uses the BATGEN command to activate, block, or cap queues.

Strategy for defining queues is explained in the section below, Planning a Batch Subsystem.

Submission of User Jobs: Users submitting jobs (with the JOB command) may specify the following queue parameters: a specific queue, maximum amount of CPU time for the job, elapsed time allowed before the job aborts, priority within the queue, and file unit (for COMINPUT files only). When a user does not specify queue parameters, the Batch monitor places the job in the first available queue and assigns the queue's default values to the job.

The Batch Administrator must either make that first available queue a reasonable default queue, or inform users which queues they should use and the default values of those queues.

By using the -STATUS and -DISPLAY options of the BATGEN command, users can find out which queues are available and what characteristics they have. They can then submit their jobs to the appropriate queues.

Note

> To use Batch, a user must have a command level depth of at least two levels. A user's batch jobs will fail if the user is set up with a command level depth of only one level. (Users can issue the LIST_LIMITS command to find out the limits of their command environment.) Use EDIT_PROFILE to change the command level depth for users who need more levels.

Planning a Batch Subsystem

The basic decisions the System Administrator and Batch Administrator must make in planning a Batch subsystem are the following:

- The number of queues to be defined
- The number of phantoms to be allocated to run Batch
- The time slice and scheduler priority of each queue
- The order in which queues are searched for job submission and job initiation

Some guidelines for making these decisions follow.

Number of Queues: A Batch queue contains default and maximum job parameters for jobs submitted to the queue and allows only one batch job to execute at a time. Therefore, in deciding how many queues to set up, you have to answer such questions as: How many batch jobs should be able to execute at a time? What are good default and maximum parameters for various kinds of batch jobs? How many queues do you need to support these various combinations?

A Batch subsystem can consist of a single queue with no limits (except for user-defined limits) placed on jobs running within it. In such a subsystem, all jobs run sequentially and have the same runtime priority (although users can request priorities from 9 to 0 for their jobs).

Alternatively, a Batch subsystem can contain from 2 to 16 queues. (The order in which the queues are numbered depends on the order in which they were added.) In a multiple-queue Batch subsystem, the Batch monitor checks each queue in turn, beginning with queue number one. If the monitor finds a job waiting to run and a phantom is available, it runs the job. If sixteen queues have jobs and sixteen phantoms are free, then one job from each queue is started. When the last of these jobs has been started, the monitor checks each queue again to see if any jobs have finished or aborted. If so, the monitor marks the job as completed or aborted, deletes temporary files, and then checks the queue for another waiting job.

If, however, there are fewer available phantoms than queues, the Batch monitor serves the queues differently. For example, if there are three queues but only one phantom available to run jobs, the monitor runs all waiting jobs from queue 1 before running a job from queue 2. Jobs from queue 3 are not run until queues 1 and 2 are both empty or until they contain only held jobs. (A held job is a job submitted with the -HOLD option of the JOB command.)

Number of Phantoms: The NPUSR directive in the system configuration file sets the number of phantoms for the system. The Batch subsystem requires one phantom for the Batch monitor. The Batch monitor (which runs under the user ID BATCH_SERVICE) runs Batch jobs on whatever other phantoms are available.

The number of batch jobs that can run simultaneously is limited by the number of queues and the number of available phantoms. Because only one job per queue can execute at one time, the number of jobs running simultaneously cannot be greater than the number of queues. On the other hand, the number of jobs running simultaneously cannot be greater than the number of available phantoms because no job can run without a phantom.

If you have many phantoms available and you expect Batch use to be heavy, you can define 10-16 queues to allow 10-16 jobs to run at once. If your system has only two or three available phantoms, you will probably not want to set up more than six queues.

You do not have to limit the number of queues to the number of phantoms. Setting up more queues than phantoms is a good strategy because the queues can separate jobs by priority. An example would be a Batch subsystem with three queues, each with different CPU and elapsed time limits. The first queue, with a stringent CPU time limit, accepts only very short jobs and gives them top priority. The second queue, with moderate (or no) CPU time limits and a moderate elapsed time limit, accepts average-length jobs and gives them medium priority. The third queue has no limits and is intended for large, slow-executing jobs. Jobs in the third queue do not run unless the other queues either are empty or have phantoms running them.

Time Slices and Scheduler Priorities: Every process on the system, including batch phantoms, has a time slice and a scheduler priority. By default, user phantoms run at the same priority level and with the same time slice as the user.

You can set up a Batch subsystem with different queues running batch jobs at different priority levels and with different time slices. You can set the priority level so that batch jobs receive more or less attention from the scheduler (in relation to other processes on the system). You can also use the time slice to control the length of time batch jobs are allowed to run before being rescheduled.

Because time slices and scheduler priorities are set individually for each queue, you can tailor queues for quick, average, or slow jobs. Following are some general guidelines:

- Queues for short jobs should have a limited CPU time, a relatively high priority, and a short or normal time slice. These queues can operate even when interactive use of the system is fairly heavy. To force users to set CPU time limits on their jobs, set the queue's default CPU time higher than its maximum CPU time. A job not submitted with the -CPTIME option cannot use such a queue.

- Queues for average jobs should have default time slices and priorities.

- Queues for large, slow jobs should have no CPU time limit, no elapsed time limit, a large time slice, and a relatively low priority. These queues cannot run jobs when interactive use is heavy, but can take advantage of free CPU time when interactive use is light. A queue with an IDLE priority runs jobs only when no other processes (batch and otherwise) are waiting for execution.

The PRIMOS scheduling mechanism has six levels: four numbered from 0 (lowest priority) through 3, IDLE, and SUSPEND. (Many systems use only levels 0 and 1 or 0, 1, and 2.)

To take full advantage of the scheduler, distribute processes on your system evenly across each of the priority levels you intend to use. For example, in a system with many users at priority 1 and only one process (such as a batch job) at priority 0, the batch job may run faster than any of the interactive users. Setting more processes (such as spool phantoms) to priority level 0 tends to alleviate this problem.

Use the CHAP command, described in the Operator's Guide to System Commands, to change priority and time slice levels of processes other than the Batch monitor and batch jobs.

Search Order of Batch Queues: The Batch monitor searches its list of queues either to find a queue in which to put a job or to find which queues have jobs that need to run. The monitor searches the queues in the order in which they were added to the system.

To find out the search order of your Batch subsystem, use the -DISPLAY or -STATUS option of the BATGEN command.

Use the following guidelines to establish the search order of your Batch subsystem:

- Queues for very short jobs should come first in the search order but should not accept jobs without the -CPTIME option.

- The default queue (the queue that accepts jobs submitted without options) should be the first queue into which a job can fall. This queue, therefore, must either be the first queue in the search order or must be preceded by queues that require an option (such as -CPTIME) supplied by the user.

- Queues for large, slow, background jobs should be at the bottom of the search list.

Designating a Batch Administrator

The Batch Administrator is responsible for most of the administrative duties involving the Batch subsystem. If you are not going to serve as Batch Administrator, you must designate another person (such as a senior system operator) as Batch Administrator. If you wish, you may have several Batch Administrators.

When you run the INIT program, you may designate one or more Batch Administrators. The INIT program uses this information to set up the access to the Batch data base. Batch Administrators receive ALL access to the BATCHQ UFD and to all its sub-UFDs and files. Users SYSTEM and BATCH_SERVICE also receive these privileges because they are automatically set up as Batch Administrators by the INIT program.

The responsibilities of the Batch Administrator include creating, monitoring, and maintaining the Batch queues. The Batch Administrator runs the FIXBAT and INIT programs to repair or replace a damaged Batch data base. These tasks are fully described in the Operator's Guide to the Batch Subsystem, which also details the system operator's responsibilities for Batch.

If you designate a Batch Administrator, you, as System Administrator, are still responsible for two ongoing tasks:

- Ensuring that the system configuration continues to be appropriate for the Batch usage on your system

- Updating the PRIMOS.COMI system startup file to reflect changes in the way Batch is started up (such as when the maximum time slice or the scheduler priority of batch jobs is changed)

Controlling the Batch Subsystem

After the Batch subsystem is set up and started, the System Administrator, the Batch Administrator, and the system operator can control it by performing the following operations:

- Pausing the monitor, temporarily preventing batch jobs from being initiated. (Jobs currently executing continue until they finish.)

- Blocking individual queues, thus keeping those queues from accepting new jobs while letting the rest of the subsystem continue running. (Jobs already in a blocked queue are not affected.)

- Capping individual queues, thus keeping those queues from executing new jobs while letting the rest of the subsystem continue running. (Jobs currently executing continue until they finish.)

- Adding new queues.

- Deleting queues, after allowing all jobs in the queues to finish. If an emergency requires an immediate cessation of all activity in a queue, first cap and block the queue. Then delete the queue after letting the jobs in the queue finish executing, or after canceling waiting jobs with the JOB -CANCEL command.

- Aborting, canceling, restarting, holding, or releasing individual jobs. (A held job remains in the queue but cannot execute until you release it.)

You can perform all of these operations from the supervisor terminal. If you are logged in as a Batch Administrator on a user terminal, you cannot start the Batch monitor, abort jobs, or restart jobs. Only users SYSTEM and BATCH_SERVICE can hold, release, and display status and submission information for batch jobs.

For details on performing these operations, see the Operator's Guide to the Batch Subsystem.

7
Allocating System Resources

This chapter covers the values and allocation of certain system resources. These resources are considered in the following order:

- System default configuration values and the corresponding directives to alter each value

- 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

- EPF libraries, including a description of the system entrypoint search list

- Shared static-mode libraries, including a table of the shared static-mode library package numbers

SYSTEM CONFIGURATION DEFAULTS

Default values for system parameters are established at cold start by the appropriate directives in the system configuration file. The parameters, their defaults, and the directives to alter these defaults are given in Table 7-1. For details on configuration directives, see Chapter 10, CONFIGURATION DIRECTIVES.

Table 7-1
Configuration Parameters and Directives

Parameter	Default	Meaning	Configuration Directive
ABBREV processor	YES	Enabled	ABBREV
AMLC DMC input buffer (tumble tables)	'60	48 chars	AMLIBL
AMLC/ICS programmable clock baud rate	'22600	9600 bps	AMLCLK
Assignable AMLC lines	0	None	NAMLC
ASR terminal input buffer	'200	128 chars	ASRBUF
ASR terminal output buffer	'300	192 chars	ASRBUF
Asynchronous line input buffer	'200	128 chars	AMLBUF
Asynchronous line output buffer	'300	192 chars	AMLBUF
Carrier check operations interval	2	.2 secs	AMLTIM
DMQ AMLC/ICS buffer	'40	32 chars	AMLBUF
DTR handling at logout	–	Not	DTRDRP
Event Logging (system)	0	Enabled	LOGREC
File system read/write lock	1	EXCL	RWLOCK
Halt on memory parity error	YES	Halt	MEMHLT
Inactivity before forced logout (seconds)	'1750	1000 min	LOUTQM
ICS input queue size	'77	63 chars	ICS INPQSZ
Interrupt rate for ICS lines	'12	10	ICS INTRPT
Line speeds for ICS controller	'113	75 bps	ASYNC JUMPER
	'226	150 bps	
	'3410	1800 bps	
Locate buffers, number of	'100	64 pages	NLBUF
Login inactivity timeout (minutes)	3	3 mins	LOTLIM
Login allowed while logged in	YES	Allowed	LOGLOG
Logout on asynchronous line disconnect	NO	No logout	DISLOG
Maximum main memory	0	All memory	MAXPAG
Maximum per-user file units	'77772	32762	FILUNT
Minimum grace time for terminal lines	0	Disabled	AMLTIM
Modem disconnect operations rate	'3410	1800	AMLTIM
Network event logging	0	Enabled	NETREC
Phantom users, number	1	1 phantom	NPUSR
Prepaged pages, number of	3	3 pages	PREPAG
Print configuration directives	NO	Disabled	TYPOUT
Print LOGIN/LOGOUT messages	YES	Enabled	LOGMSG
Ratio of ALTDEV to PAGDEV use	5	5 of 10	PRATIO
Record unsuccessful login attempts	NO	Disabled	LOGBAD
Remote users, number of	0	None	NRUSR
Remote users' input buffer	'201	260 chars	REMBUF
Remote users' output buffer	'101	130 chars	REMBUF
Restart after power failure	-	Disabled	UPS
Slave users, number	'177777	None	NSLUSR
Supervisor terminal baud rate	'1010	300 bps	ASRATE
Synchronous lines	OFF	Disabled	SMLC
System erase character	'242	"	ERASE
System kill character	'277	?	KILL
Terminal users, number	0	None	NTUSR
Total virtual address space (segments)	'1776	1022 segs	NSEG
VMFA segments for EPFs, number of	'144	100	NVMFS
Wired memory size printout	NO	Disabled	WIRMEM

SHARED SEGMENTS

Shared subsystems normally are incorporated into the PRIMOS operating system at cold start. To incorporate a shared subsystem after startup, use the following command sequence from the supervisor terminal:

```
OPRPRI 1
SHARE pathname segment-number [access-rights]
OPRPRI 0
```

pathname is the pathname of the file to be restored into segment segment-number.

segment-number is the octal number of the segment to be shared. See Table 7-2 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. The System Administrator assigns and coordinates the use of shared segments for customer use.

Caution

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, which are reserved for PRIMOS. Other segments that may contain system utilities are listed in Table 7-2.

Table 7-2
Contents of Shared Segments at Rev. 20.2

Segment	Product
2000	ED
2001-2003	DBMS
2004-2011	SPSS (note 1)
2012	DBMS
2013	BASIC/VM
2014	Reserved for Prime
2015	DPTX
2016	COBOL
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

Table 7-2 (continued)
Contents of Shared Segments at Rev. 20.2

Segment	Product
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
2577-2599	Reserved for Prime
2600-2601	ROAM/DDM
2602-2665	Reserved for Prime
2666-2765	Reserved for EPFs
6001	Per-user linkage segment (note 1)
6006	Per-user linkage segment (note 2)
6007	Per-user linkage segment (note 3)
6010	ORACLE, EMACS, PRIMEWAY
6011	ROAM

Notes to Table 7-2

1. 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

2. Segment 6006

Allocated	Product
0-37777	FTS
40000-70000	MIDASPLUS
70001-9999	Reserved for Prime
10000-17777	ROAM/DDM

3. Segment 6007

Allocated	Product
0-47777	ROAM
50000-122777	PRISAM
140000-177777	MAGLIB

EPF LIBRARIES

An EPF library 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 EPF Libraries

EPF libraries share the following advantages 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 users with many 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, EPF libraries provide the following advantages over shared static-mode libraries:

- EPF libraries are not shared into static segments, but instead are brought into memory only 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 on the system.
- EPF libraries can be protected against modification with ACLs.
- EPF libraries 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.)
- EPF libraries are not loaded into the segments at cold start.
- Users can create their own EPF libraries and use ACLs to restrict their use as desired.

For more information on EPF libraries, see the Programmer's Guide to BIND and EPFs and the Advanced Programmer's Guide, Volume 1: BIND and EPFs.

Installation of EPF Libraries

The directory LIBRARIES* contains the Prime-supplied EPF libraries that are available when Rev. 20.2 is installed. These EPF libraries are not shared with the SHARE command as are static-mode libraries.

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

1. Use the COPY command to copy the EPF into LIBRARIES*. For details on using the COPY command to replace existing EPFs, see the Programmer's Guide to BIND and EPFs or Chapter 16, ADDING AND MODIFYING SYSTEM SOFTWARE, in this guide.

2. Use a text editor (such as ED or EMACS) and add the name of the library to the system entrypoint search list. (The entrypoint search list, named ENTRY$.SR in the SEARCH_RULES* directory, is described in the next section.)

If an EPF library is mapped into a user's address space when a new library is copied into the LIBRARIES* directory, the user continues, uninterrupted, using the original library. A user who invokes the library after the new file has been copied gets the new library.

If users modify libraries frequently, you should periodically clean up the LIBRARIES* directory by deleting old replace files (with the .RPn suffix) that are no longer used.

SYSTEM ENTRYPOINT SEARCH LIST

At Rev. 20.2 ENTRY$.SR, the default entrypoint search list, is kept in a new directory called SEARCH_RULES*. (The search list determines the order in which PRIMOS searches libraries to find a match to a subroutine entrypoint in a program.) The installation program SYSTEM>INSTALL.STD.COMI automatically creates the new directory and copies SYSTEM>ENTRY$.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
```

Unlike the ENTRY$.SR file, which can be modified, you should not modify the ADMIN$.ENTRY$.SR file.

Format of SEARCH_RULES*>ENTRY$.SR

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

Because SEARCH_RULES*>ENTRY$.SR is a text file, you can modify it with a text editor or display its contents with the SLIST command, as in the following example:

```
OK, SLIST SEARCH_RULES*>ENTRY$.SR
/* ENTRY$.SR, SYSTEM, RDW, 05/21/85
/* ENTRY Search list
/* Copyright (c) 1985, Prime Computer, Inc., Natick, MA 01760
LIBRARIES*>SYSTEM_LIBRARY.RUN
LIBRARIES*>FORTRAN_IO_LIBRARY.RUN
LIBRARIES*>APPLICATION_LIBRARY.RUN
LIBRARIES*>COMMON_ENVELOPE.RUN
LIBRARIES*>PRIMOS_LIBRARY.RUN
LIBRARIES*>FTN_LIBRARY.RUN
LIBRARIES*>MAGTAPE_LIBRARY.RUN
LIBRARIES*>TTYIN$
LIBRARIES*>CC_LIBRARY.RUN
LIBRARIES*>PRIMIX_SYSTEM_LIBRARY.RUN
-STATIC_MODE_LIBRARIES
```

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* UFD) are to be searched first (after internal PRIMOS entrypoints, which are always searched before any libraries listed in the entrypoint search list). These libraries include SYSTEM_LIBRARY (the system library), FORTRAN_IO_LIBRARY (the FORTRAN I/O library), and APPLICATION_LIBRARY (the application library).

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 library.

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. A frequently called subroutine (such as one in SYSTEM_LIBRARY) 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 libraries appear determines which copy of a subroutine is actually invoked.

Access Rights on SEARCH_RULES*>ENTRY$.SR

Set access rights on the system entrypoint search list 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. When a user entrypoint search list is in effect, SEARCH_RULES*>ENTRY$.SR is not used automatically if the user's command line includes the -NO_SYSTEM keyword and the list does not include -SYSTEM.

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

WARNING

You should 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.

If a user maintains a private copy of ENTRY$.SR, sets the rules with the -NO_SYSTEM option of the SET_SEARCH_RULES command, and does not use -SYSTEM as a rule in the list to get the system search rules, that user is responsible for making any changes to the private copy that may be necessary because of the changes to the system copy of ENTRY$.SR.

Linkage Faults

If the end of the search list is reached and the target subroutine is not found or the ENTRY$ list has been improperly installed or altered, 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!
```

Perform the following steps to remedy the condition:

1. Enter the following command to reinitialize the system default search list:

   ```
   SET_SEARCH_RULES -DEFAULT ENTRY$
   ```

   If you can perform the operation that caused the linkage fault without generating an error message, you may have been using a private entrypoint search list that contains an error. If you repeat the operation and the linkage fault occurs again, perform Step 2.

2. Enter the following command to display the entrypoint search list:

   ```
   LIST_SEARCH_RULES ENTRY$
   ```

   Check that the keyword -PRIMOS_DIRECT_ENTRIES is at the top of the ENTRY$ list that PRIMOS displays after you enter the command. If it is not, check that the SEARCH_RULES*>ADMIN$.ENTRY$.SR file contains the rule -PRIMOS_DIRECT_ENTRIES. If the file or the entry or both are missing, create them and try the operation again.

   Note

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

3. 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.)

SHARED STATIC-MODE LIBRARIES

A system can have a maximum of 32 shared static-mode libraries. The SPOOL libraries are shipped with all systems. Any other library is supplied if the customer has purchased that particular software product. See Table 7-3 for a list of shared static-mode libraries.

Table 7-3
Shared Library Package Numbers

Package Number	Shared Library
'1	Reserved for Prime
'2	VKDALB and MPLUSLB
'3	Reserved for Prime
'4	VFORMS
'5	DBMSLB
'6	OAS
'7	EMACS
'10	Reserved for Prime
'11	FTS
'12	SPOOL
'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

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 impure 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. To reduce the number of segments, use the MIX subcommand of the SEG loader. For details on SEG, see the SEG and LOAD Reference Guide.

Like EPF libraries, the use of shared static-mode libraries means that user runfiles are smaller and execute faster, that system load is reduced, and that programs do not have to be reloaded when a library is updated.

Installation of Shared Libraries

Shared static-mode libraries must be installed each time the system is cold started. The runfiles are resident in UFD SYSTEM of the U1 Master Disk. Copy these runfiles to UFD SYSTEM on the system disk.

The SHARE commands that install the runfiles at cold start may be incorporated into the PRIMOS.COMI file or called from PRIMOS.COMI by the COMINPUT command. SHARE commands are included in the PRIMOS.COMI.TEMPLATE file in the UFD PRIRUN. (See Chapter 8 for the PRIMOS.COMI.TEMPLATE file.)

The PRIMOS.COMI file installs memory image files in the proper segments (see Table 7-2) and runs the programs required to inform PRIMOS that shared libraries are activated. After the libraries are installed, users with programs loaded using the special shared library object files may run V-mode and I-mode programs accessing these shared libraries. If the shared libraries are not installed, programs that expect the shared libraries to be resident receive an error message from PRIMOS whenever an attempt is made 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 installing a shared library, shut down PRIMOS and then reboot it.

Shared Library Usage

If one of the shared libraries is to be used, all appropriate shared libraries must also be used. If the user wishes to use the shared PL/1G library and also requires MIDASPLUS or COBOL, the shared MIDASPLUS and COBOL libraries must also be used. After the new V-mode or I-mode runfile has been created and the shared libraries have been installed, the user's programs may be run as before.

The spool (VSPOO$) library (in segment '2167) should always be shared. Other libraries may be shared as desired.

Administration

The shared static-mode library files are in UFD LIB. For LOAD to operate properly, UFD LIB must be on the logical partition 0. If LIB is not on the logical partition 0, LOAD returns a "Not found" message to subcommands such as LIBRARY and SPLIT.

If static-mode libraries are not to be shared for systemwide use, users planning to use them must modify their command files to use the unshared library files.

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 UFD SYSTEM so that installation is easily accomplished by running the appropriate command file.

Caution

A shared static-mode library should not be reshared while being used. 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 in UFD SYSTEM at any time because these files are loaded into memory only when the explicit SHARE commands are given (such as during system startup).

PART II

Creating the System

8 Installation

This chapter describes how to install Revision 20.2 software in the following situations:

- Initial installation of Rev. 20.2 PRIMOS on a new system
- Upgrading from Rev. 19.4 PRIMOS to Rev. 20.2
- Upgrading from Rev. 20 PRIMOS to Rev. 20.2

The chapter also describes the PRIMOS.COMI.TEMPLATE file.

HARDWARE AND SOFTWARE REQUIREMENTS

Before you install Rev. 20.2, your system must meet the following hardware requirements:

- The system must have at least 512 kilobytes of physical memory. Even if the system has more than 512 kilobytes, the first 512 kilobytes must be contiguous starting at location 0.
- The system must have a partition onto which you will load the Rev. 20.2 software.
- You cannot use a diskette (floppy disk) for the command device (COMDEV), the primary paging partition (PAGDEV), or the alternate paging partition (ALTDEV).

- The new system boot cannot boot from Option B (4001) and Option B' (B prime or 4002) disk controllers, diskette controllers, 7-track tape drives, 4020 9-track tape controllers, and paper tape.

The controllers and drives mentioned above can be used under PRIMOS after it is running.

Whether you are performing an initial installation or are upgrading, the following Rev. 20.2 software must be installed together:

- MFD>BOOT
- PRIRUN directory, which contains Rev. 20.2 PRIMOS
- CMDNC0>COPY_DISK.SAVE
- CMDNC0>FIX_DISK.SAVE
- CMDNC0>MAGRST
- CMDNC0>MAGSAV
- CMDNC0>MAKE.SAVE
- DOS>DOS.SAVE
- DOWN_LINE_LOAD* directory, if you have ICS controllers

After Rev. 20, COPY_DISK.SAVE, MAKE.SAVE, and DOS>DOS.SAVE have the .SAVE suffix. You should delete older versions of these programs (COPY_DISK, MAKE, and DOS>*DOS64) to prevent invocation of these older versions.

INITIAL INSTALLATION

All Prime software products shipped with new Prime computer systems are stored on several magnetic tapes by using the MAGSAV utility. The tapes contain the operating system (PRIMOS), the utilities, the nonchargeable software products, and any separately priced software products that you have ordered.

If you have problems with the tapes, call your Customer Support Center.

Initial Installation Procedure

Use the following procedure to install the Rev. 20.2 software from the tapes:

1. Turn on power to equipment.

2. Mount the tape containing the MAKE utility.

3. Boot MAKE from tape and format the command partition (COMDEV).

4. Boot MAKE from tape and format the paging partition (PAGDEV).

5. Boot PRIMOS from tape with the '100000 bit set in the BOOT option word. If necessary, use the SETIME command to set the system date and time.

6. Use the MTRESUME command to execute the MAGRST utility from tape and restore the M202U1 master disk partition to the command partition.

7. Use NSED (the nonshared Editor) to create a configuration file named CONFIG in CMDNC0. At Rev. 20.2, the NPUSR directive must have a value of at least 1 because the Login Server runs as a phantom.

8. Use the COPY command to copy the file PRIMOS.COMI.TEMPLATE from UFD PRIRUN into CMDNC0 and rename it PRIMOS.COMI. The PRIMOS.COMI.TEMPLATE file is listed at the end of this chapter.

9. Use NSED to modify PRIMOS.COMI for your system needs.

10. Use NSED to modify the SEARCH_RULES*>ENTRY$.SR file for your system needs.

11. Use the SHUTDN ALL command to shut down PRIMOS.

12. Boot PRIMOS from disk with the autoboot option word.

13. Set up the User Profile Data Base with EDIT_PROFILE. At Rev. 20.2, you can configure attributes for the supervisor terminal under the entry for SYSTEM.

14. Set security on your system with Access Control Lists (ACLs). The ACLs on SERVERS* and on SEARCH_RULES*, two new directories at Rev. 20.2, should be $REST:LUR. You can execute the INIT_ACLS.CPL program in the TOOLS directory.

15. Set up the files in the spool queue. For details, see the Operator's Guide to the Spooler Subsystem.

16. Set up the Batch subsystem and define the Batch queues. For details, see the Operator's Guide to the Batch Subsystem.

17. If you want to set quotas on top-level user UFDs, use the SET_QUOTA command.

UPGRADE INSTALLATION

The next two sections describe how to upgrade from Rev. 19.4 to Rev. 20.2 and from Rev. 20.0 to Rev 20.2.

When upgrading to Rev. 20.2, you receive a tape of the M202U1 partition of the Master Disk. The M202U1 partition contains nonchargeable software. Although the tape is a MAGSAV tape, you cannot boot PRIMOS from it.

Depending on which separately chargeable software you ordered (if any), you may also receive tapes of the M202C1 and M202D1 partitions. Your Customer Support Center will help you install these partitions.

Upgrade Procedure From Rev. 19.4 to 20.2

Before you perform the upgrade, use the following guidelines:

- If your system uses ROAM-based products, save the ROAM files before installing Rev. 20.2. See the Rev. 20.2 update package to the ROAM Administrator's Guide for details.

- After you install Rev.. 20.2 PRIMOS, it is recommended that you use Rev. 20.2 MAKE to convert your disks to Rev. 20-format disks. Rev. 20 disks can be created only on a system running Rev. 20 or later PRIMOS. After you convert your disks to Rev. 20, you cannot run Rev. 19.4 PRIMOS on them. If, however, you are connected with PRIMENET to a pre-Rev. 20.2 system, your Rev. 20 disks can be added remotely to that system.

- Because only ACL directories can be converted to Rev. 20 hashed directories, you should convert password directories to ACL directories before converting the disks to Rev 20.2.

Perform the following steps to upgrade your Rev. 19.4 system to Rev. 20.2. The procedure includes running Rev. 20.2 MAKE on the old command partition to convert it to a Rev. 20.2 partition. The description of the procedure refers to the spare partition onto which you load the Rev. 20.2 software from tape.

1. Use the MESSAGE command to warn users well in advance that they are going to be logged out.

2. Use the LOGOUT ALL command to clear the system of users. This sets MAXUSR to 0.

3. Make sure that all top-level UFDs are ACL directories so that they will be hashed when they are restored.

4. Save the command partition to tape or disk.

5. Make sure the Batch queues are empty because otherwise waiting jobs will be lost.

6. Attach to the MFD of the spare partition onto which you will restore the Rev. 20.2 software.

7. Use the ASSIGN command to assign the tape drive that holds the M202U1 tape.

8. Use the MAGRST command to restore the nonchargeable Rev. 20.2 software from the M202U1 tape to the spare partition.

9. Attach to the BOOTRUN top-level directory of the Rev. 20.2 spare partition.

10. Use a text editor to edit the BOOT.INSTALL.COMI file that installs the BOOT file on the spare partition. Use the editor to change "mfd xxxxxx" to include the name of the spare partition. The BOOT.INSTALL.COMI file describes how to change the BOOT file's supervisor terminal baud rate from the default of 300 baud.

11. Use the COMINPUT command to execute the BOOT.INSTALL.COMI file. The file installs the BOOT file on the MFD of the spare partition.

12. Boot Rev. 20.2 PRIMOS from the spare partition by setting the '100000 bit in the BOOT option word (for example, BOOT 100114). For the COMDEV, use the physical device number of the spare partition. Give the name PRIRUN>PRIMOS.SAVE for the runfile treename. (The purpose of this step is to run Rev. 20.2 MAKE on the old command partition, as described in Step 14.)

13. Add the old command partition to the Assignable Disks Table (using the DISKS command) and assign it (using the ASSIGN DISK command).

14. Run Rev. 20.2 MAKE (in CMDNC0 of the spare partition) to remake the old Rev. 19.4 command partition. MAKE also installs the Rev. 20.2 BOOT file in the MFD. An example of the command line follows. In the example, the name of the old partition is SYSCMD and its physical device number is 6162.

```
MAKE -DISK 6162 -PART SYSCMD -NEWDSK -DSKREV 20 -FMT
```

15. Unassign the disk partition with the UNASSIGN DISK command.

16. Remove the old command partition from the Assignable Disks Table (by using the DISKS NOT command).

17. Make the disk available to users of the system (by using the ADDISK command).

18. Restore the old command partition from the saved tape or disk.

19. Attach to the SYSTEM top-level directory of the spare partition.

20. Use a text editor to edit the INSTALL.STD.COMI file. Be sure to perform the following tasks: change the <MXXXU1> names to the name of the spare partition, add the name of the old command partition to the destination names, and delete commands that will overwrite existing software that you want to save.

21. Attach to the BOOTRUN top-level directory of the Rev. 20.2 spare partition.

22. Use a text editor to edit the BOOT.INSTALL.COMI file. Change the name of the partition to the name that you specified in Step 14.

23. Use the COMINPUT command to run the SYSTEM>INSTALL.STD.COMI file in the Rev. 20.2 partition. The INSTALL.STD.COMI program uses FUTIL to copy the appropriate files from the spare partition to the directories on the old command partition. INSTALL.STD.COMI also executes the BOOTRUN>BOOT.INSTALL.COMI file that installs the BOOT program on the MFD of the old command partition. After this step, your old command partition will be a Rev. 20 partition with Rev. 20.2 software on it.

24. Check your CONFIG and PRIMOS.COMI files to make sure they require no changes. At Rev. 20.2, the NPUSR directive must have a value of at least 1 because the Login Server runs as a phantom. Also check that the printer environment files in the SPOOLQ UFD are correct.

25. Shut down PRIMOS and reboot it on the old command partition.

26. Save the other partitions (to tape or disk) and run Rev. 20.2 MAKE to convert those partitions to Rev. 20. As with Step 3, make sure that all top-level UFDs are ACL directories so that they will be hashed when they are restored. After running MAKE, restore the files from the backup tapes or disks with a utility (for example, MAGRST) that corresponds to the utility with which you saved the files (for example, MAGSAV).

27. Check that the ACL $REST:LUR is on SERVERS* and on SEARCH_RULES*, two new directories at Rev. 20.2.

28. If you have a Batch subsystem, redefine your Batch queues.

> WARNING
>
> Do not boot pre-Rev. 20 PRIMOS on a system with Rev. 20 partitions. This will result in file system errors and possibly loss of user or system data.

Upgrade Procedure From Rev. 20 to Rev 20.2

1. Use the MESSAGE command to warn users well in advance that they are going to be logged out.

2. Enter the LOGOUT ALL command to clear the system of users. This sets MAXUSR to 0.

3. Save the command partition to tape or disk.

4. Use the ASSIGN command to assign the tape drive that holds the M202U1 tape.

5. Use the MAGRST command to restore the nonchargeable Rev. 20.2 software from the M202U1 tape.

6. Restore the TOOLS directory from your tape as a top-level UFD in the command partition.

7. Restore Rev. 20.2 Master Disk software (nonchargeable) from your tape(s), using LOAD_20.2, a program in the TOOLS directory. This step automatically installs almost all nonchargeable software.

8. Check your CONFIG and PRIMOS.COMI files to make sure they require no changes. At Rev. 20.2, the NPUSR directive must have a value of at least 1 because the Login Server runs as a phantom. Also check that the printer environment files in the SPOOLQ UFD are correct.

9. Check that the ACL $REST:LUR is on SERVERS* and on SEARCH_RULES*, two new directories at Rev. 20.2.

10. Shut down PRIMOS and reboot it on the old command partition.

11. If you have a Batch subsystem, redefine your Batch queues.

PRIMOS.COMI TEMPLATE FILE

Rev. 20.2 software is shipped with a PRIMOS.COMI.TEMPLATE file in the PRIRUN directory. If you do not already have a PRIMOS.COMI file in CMDNC0, copy PRIMOS.COMI.TEMPLATE into CMDNC0 and use a text editor to modify the file to suit your system needs.

Below is the PRIMOS.COMI.TEMPLATE file as shipped from Prime.

```
/* PRIMOS.COMI.TEMPLATE, PRIRUN, JK-JNS-RJR, 04/26/84
/* TEMPLATE FOR MAKING C_PRMO FILE FOR BRINGING UP PRIMOS
/* Copyright (C) 1980, Prime Computer, Inc., Wellesley, MA  02181
/*
/* Shared libraries are released as part of the unchargeable software
/* as of Rev 19.1.  The commands for sharing the following libraries
/* are included as part of this file and also as part of the
/* PRODUCT.SHARE.COMI file:  COBOL, FORMS, MIDASPLUS.  This
/* may cause these libraries to be shared twice at system startup.
/*
/* This is not a problem, but if you wish you may remove the
/* duplicate commands.
/*
CONFIG -DATA                /* specify CONFIG file after -DATA
ADDISK                      /* specify local disks to be added
AMLC TTY                    /* specify AMLC lines
OPR 1                       /* SHARE REQUIRES OPR 1
SHARE SYSTEM>ED2000 2000         /* SHARE the editor  -  ED
/*
SHARE SYSTEM>MP2122 2122 700       /* SHARE MIDASPLUS LIBRARY
SHARE SYSTEM>MP2123 2123 700
SHARE 2124 700
SHARE 2125 700
R SYSTEM>MP4000 1/2
SHARE 2122
R SYSTEM>IMIDASPLUS SYSTEM>MPLUS.CONFIG
/*
SHARE SYSTEM>F2021A 2021 700         /* SHARE FORMS LIBRARY
SHARE SYSTEM>F2021B 2021 700
R SYSTEM>F4000 1/4
SHARE 2021
SHARE SYSTEM>S$2167 2167          /* SHARE SPOOL LIBRARIES
R SYSTEM>S$4000 1/12
SHARE 2020 700
/* MAGLIB is a shared library as of Rev 19.2
SHARE SYSTEM>ML2222 2222           /* SHARE MAGLIB
R SYSTEM>ML4000 1/16
OPR 0
PROP PRO -START                  /* START SPOOLER PHANTOM
CO SYSTEM>BASICV.SHARE.COMI 7    /* SHARE BASICV COMPILER
CO SYSTEM>COBOL.SHARE.COMI 7     /* SHARE COBOL COMPILER AND LIBRARY
CO SYSTEM>ROAM.SHARE.COMI 7      /* SHARE ROAM before DBMS
CO SYSTEM>DBMS.SHARE.COMI 7      /* SHARE DBMS
CO SYSTEM>DPTX.SHARE.COMI 7      /* SHARE DPTX
CO SYSTEM>EMACS.SHARE.COMI 7     /* SHARE EMACS
CO SYSTEM>FED.SHARE.COMI 7       /* SHARE FED
CO SYSTEM>FORMS.SHARE.COMI 7     /* SHARE FORMS LIBRARY
CO SYSTEM>FTS.SHARE.COMI 7       /* SHARE FTS
CO SYSTEM>MIDASPLUS.SHARE.COMI 7   /* SHARE MIDASPLUS LIBRARY
CO SYSTEM>POWERPLUS.SHARE.COMI 7   /* SHARE POWERPLUS
CO SYSTEM>VRPG.SHARE.COMI 7        /* SHARE VRPG
CO SYSTEM>DISCOVER_DBMS.SHARE.COMI 7    /* SHARE DISCOVER_DBMS
CO SYSTEM>DISCOVER_PRISAM.SHARE.COMI 7  /* SHARE DISCOVER_PRISAM
CO SYSTEM>CC.SHARE.COMI 7               /* SHARE CC
CO SYSTEM>CBL.SHARE.COMI 7              /* SHARE CBL
CO SYSTEM>DBG.SHARE.COMI 7              /* SHARE DBG
CO SYSTEM>PRISAM.SHARE.COMI 7           /* SHARE PRISAM
CLOSE 7
/* SET THE DATE AND TIME **********
/* TYPE MAXUSR TO ALLOW USERS TO LOG IN
CO -END
```

9 Setting System Access

The System Administrator is responsible for setting access rights on MFDs, system directories, and top-level user directories. Proper system access gives users sufficient scope to accomplish their tasks, while minimizing the danger of interference with files used in common. (Such common files may include user files, as well as system files and directories.)

Your responsibility for setting system access also includes setting a priority ACL on a partition when a user needs special access to the entire partition.

The first part of this chapter discusses what protection to set on MFDs, system directories, and top-level user directories. The second part explains how ACLs affect the operation of the ATTACH command. The third part of the chapter discusses priority ACLs.

PROTECTING SYSTEM AND USER DIRECTORIES

Although you can set system access by using directory passwords, the use of Access Control Lists (ACLs) is recommended because they provide better security and more flexibility. (See Chapter 5, SECURITY, for a comparison of ACLs and passwords.)

For details on ACLs, see the Prime User's Guide and Chapter 3 of this guide, PLANNING THE USER ENVIRONMENT.

Protecting MFDs

As a rule, you should restrict access to MFDs to users who perform administrative or operations tasks.

Other users, however, need rights to MFDs in the following situations:

- Users need Use rights to access a partition at all. (See the section below, ACLS AND THE ATTACH COMMAND.)
- Users need List rights to list the partition contents or to protect top-level directories.
- Users may need Read rights on an open system. At a minimum, you should grant users Read rights to the DSKRAT file, so that they can use the AVAIL command.

If a user has no rights to an MFD, the user cannot attach to the partition (using the ATTACH command) and cannot get any information about its contents. Granting no rights to the MFD to users is an effective way of protecting sensitive data on a partition or of limiting access to the partition to as few users as possible.

Protecting Users' Top-level Directories

Only users who have Add rights to the MFD can create top-level directories, and only users who have Protect and List access to the MFD can set protection on the top-level directories. (On many systems, only the System Administrator and operators have Protect and Add rights to the MFD. Thus, if users do not have Protect rights to the MFD and they accidentally lock themselves out of their top-level directories by destroying their ACLs, you have to create new ACLs for them in order to restore their access.)

It is your responsibility to decide what rights to give users to top-level directories. Generally, you should give at least one person (perhaps a project leader or Project Administrator) ALL rights to a top-level directory. That person can then create subdirectories and set protection as necessary.

Combinations of Access Rights: The following list suggests useful combinations of rights for users.

U Users can only attach. Use (U) access is essential if users are to do anything, anywhere in the tree below. A user without U access to the MFD cannot search the partition for attaches or for information.

LU Users can attach and list the contents of the directory.

LUR Users can attach to the directory, list directory contents, read files, and execute runfiles. Users can read all the information they want and can copy files and subdirectories from the directory (assuming they have proper rights to another directory). They cannot, however, alter the contents of the directory.

Note

U, LU, and LUR are often granted as rights to $REST.

LUX Users can attach, list directory contents, and execute local EPF runfiles. If the attach is to a local partition, the X access allows the user to execute an EPF runfile but not to read or copy it with a standard file utility, such as the COPY command. If the attach is to a remote partition, the user cannot execute the EPF. Alternatively, you can grant users U or LU rights to the directory and set X access on individual EPFs.

ALUR Users can attach, list directory contents, read files, and add files and subdirectories. Users cannot modify files or delete any of the contents of the directory. ALUR is a useful combination for users who have to trade information, but who cannot alter each other's work.

DALURW Users can do almost everything (including modifying files and deleting entries) except change the protection on the directory itself or any of its objects. DALURW is used when an administrator wants to give users all working rights to a directory, but wants to keep a firm hold on the access control to the directory.

ALL Users can do anything to the directory (or to any directories beneath it in the tree structure), including changing ACLs. In setting protection, ALL rights are dependent on the rights granted in the directory above the directory to which ALL rights are given. ALL access is generally given to the following individuals or groups:

- Administrators
- Operations personnel

- A project leader, supervisor, or instructor who needs full rights to a directory and to the disk space it commands
- A user who has full and sole responsibility for a directory and the disk space it commands
- A group of users who work closely together and share responsibility for their files, directories, and disk space

A group that shares ALL rights to a directory can meet needs more rapidly and flexibly. However, group members must be trustworthy and they must keep each other informed of changes they make in a directory. Sharing ALL rights in a group is useful in situations like the following:

- A troubleshooter joins the group for a few days. Any group member can immediately grant the troubleshooter access to the directory.
- A key file is identified. Any group member can set delete protection on it.
- A concurrency problem is being studied. Group members can alter the read/write locks on various files and study the results thus obtained.
- The group suspects that someone outside the group is using a group member's user ID. They temporarily deny access rights to that ID and observe the results.

Protecting System Directories

System directories contain Prime-supplied software that is used by some or all users of a system. Users (or certain system processes) may not be able to work if they have insufficient rights to system directories and files.

Table 9-1 lists the minimum protection required for standard system directories. Table 9-2 lists the minimum access required for special products. (You may have all, some, or none of these products on your system.)

Table 9-1
Access Rights for System Directories

Directory	Minimum Access Needed
BATCHQ	(protection set by Batch subsystem)
CMDNC0	$REST:LUR System Administrator:ALL recommended
DOS	SYSTEM:LUR
HELP*	$REST:LUR
INFO	$REST:LUR recommended
LIB	$REST:LUR DALURW recommended for users modifying the libraries
LIBRARIES*	$REST:LUR DALURW recommended for users modifying the libraries
LOGREC*	SYSTEM:DALURW Operators:ALL recommended
MFD (on command device)	$REST:LU
PRIRUN	SYSTEM:LUR
SAD	(protection maintained by EDIT_PROFILE)
SEARCH_RULES*	$REST:LUR
SEGRUN*	$REST:LUR
SERVERS*	$REST:LUR
SPOOLQ	Should be password directory
SYSCOM	$REST:LUR
SYSOVL	$REST:LUR
SYSTEM	SYSTEM:LUR $REST:LUR (for SYSTEM>DISCS)

Table 9-2
Access Rights for Special Products

Product	Directory	Minimum Access Needed
DISCOVER	DISCOVER*	Should be a password UFD
	DISCOVER_DBMS	System Administrator:ALL SYSTEM:LUR $REST:NONE
	DISCOVER_PRISAM	System Administrator:ALL SYSTEM:LUR $REST:NONE
	DISCOVER_TOOLS	System Administrator:ALL $REST:ALUR
FED	FED*	$REST:RU Installer:ALL
FORMS	FORMS*	$REST:ALL
FTS	FTS	System Administrator:ALL
	FTSSRC	System Administrator:ALL
	FTSQ*	SYSTEM, YTSMAN, FTP, RT_FTP, and FTS Servers:ALL $REST:DALURW
POWERPLUS	POWER*	$REST:ALL
	POWRCM	$REST:ALL
PRIMENET	PRIMENET*	NETMAN:ALL RT_SERVER:AURW Network Administrator:ALL SYSTEM:ALURWX SLAVE$:LUR $REST:NONE
MFD containing PRIMENET*		NETMAN, RT_SERVER, and SLAVE$:U

Table 9-2 (continued)
Access Rights for Special Products

Product	Directory	Minimum Access Needed
PRIME/SNA	PRIME/SNA*	See PRIME/SNA Administrator's Guide.
PRIMIX		See Using PRIMIX on the Prime 50 Series.
RJE	RJSPLQ*	Operator:ALL User:ALL
	RJSPLQ*>CMDHELP RJSPLQ*>ERRHELP	Operator:LUR User:LUR
	RJSPLQ*>BINARY RJSPLQ*>PUNCH RJSPLQ*>Qnnn RJSPLQ*>SAVE RJSPLQ*>SDRF RJSPLQ*>TO_ROUTE	Operator:DALURW User:NONE
	RJSPLQ*>CMDNCO SYSCOM	Operator:NONE User:NONE

ACLS AND THE ATTACH COMMAND

During its operation, the ATTACH command checks access rights on MFDs and UFDs to determine if a user may be attached to a directory.

Following are the general rules for attaches:

- A user who has no rights to a directory cannot attach to that directory.
- A user who has no Use rights to the MFD of a partition cannot attach to any directory on that partition nor can the user gain any information about a directory (using the LD command, for example).

- If a user supplies a relative pathname in an ATTACH command, only the current directory tree is searched. A relative pathname begins with the *> symbol (for example, *>LETTERS).
- If a user specifies a partition name or a logical device number in an ATTACH command, only the specified partition is searched.
- If a user supplies a pathname that begins with a top-level directory, the search is carried out as shown in the flow chart in Figure 9-1. The only partitions that are searched are those to which the user has Use (U) rights. The order in which partitions are searched is as follows:

    1. All local partitions are searched first, in logical device order.
    2. Remote partitions (if any) are searched next, in logical device order.

Search Finish

An ATTACH search finishes when one of the following conditions is met:

- A top-level directory of the right name is found.
- All available partitions have been searched.

If the directory is found and the user has Use rights to it (and to any subdirectories specified in the pathname), the user is attached.

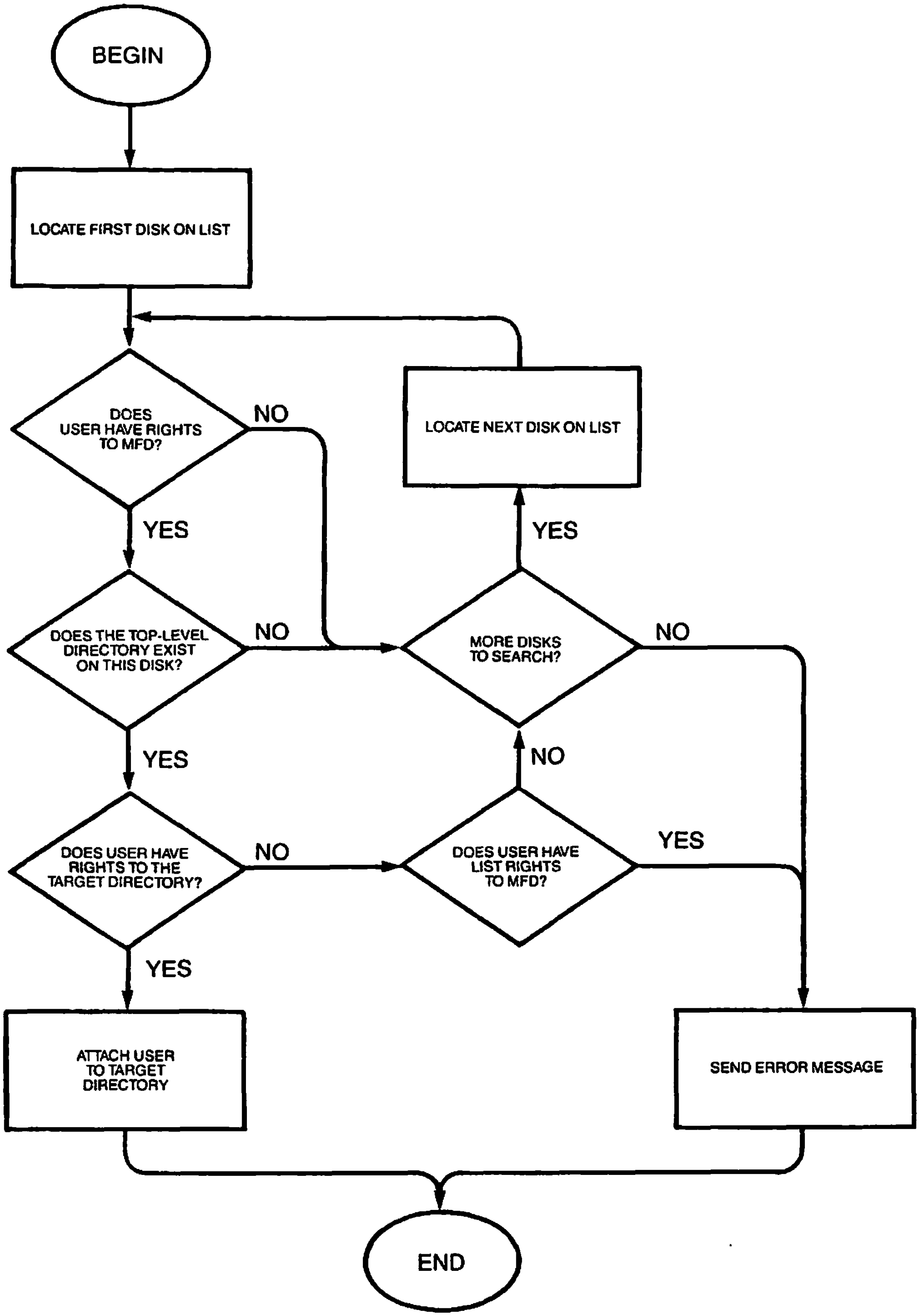
If a top-level directory of the right name is found but the user does not have Use rights to it, the following occurs:

- If the user has List rights to the MFD, the search ends and the user receives the error message "Insufficient access rights."
- If the user does not have List rights to the MFD, the search continues with the next partition on the list.

If ATTACH finishes its scan of the MFDs without being able to attach the user anywhere, one of three situations occurred:

- The specified directory does not exist.
- The specified directory was found on a partition to which the user has no rights.
- The specified directory is on a remote partition that is temporarily unavailable.

In these cases, ATTACH returns the message "Top-level directory not found or inaccessible."


BEGIN
LOCATE FIRST DISK ON LIST
DOES USER HAVE RIGHTS TO MFD?
NO
YES
LOCATE NEXT DISK ON LIST
YES
DOES THE TOP-LEVEL DIRECTORY EXIST ON THIS DISK?
NO
MORE DISKS TO SEARCH?
NO
YES
NO
DOES USER HAVE RIGHTS TO THE TARGET DIRECTORY?
NO
DOES USER HAVE LIST RIGHTS TO MFD?
YES
YES
ATTACH USER TO TARGET DIRECTORY
SEND ERROR MESSAGE
END

Search Order for ATTACH
Figure 9-1

Remote Searches

Remote searches are done in the most efficient manner when partitions from a single system are grouped together in the logical device order. You can ensure this order with the proper use of the ADDISK command (usually in the PRIMOS.COMI file).

Figure 9-2 shows good and poor orderings of a list. (In Figure 9-2, the names of local partitions begin with LOCL; the names of remote partitions begin with SYS.)

Good Order		Poor Order	
PARTITION	LDEV	PARTITION	LDEV
LOCL-1	0	LOCL-1	0
LOCL-2	1	LOCL-2	1
LOCL-3	2	LOCL-3	2
SYSA-1	3	SYSA-1	3
SYSA-2	4	SYSB-1	4
SYSA-3	5	SYSC-1	5
SYSB-1	6	SYSA-2	6
SYSB-2	7	SYSB-2	7
SYSB-3	10	SYSC-2	10
SYSC-1	11	SYSA-3	11
SYSC-2	12	SYSB-3	12
SYSC-3	13	SYSC-3	13
(3 remote calls needed to search all partitions)		(9 remote calls needed to search all partitions)	

Good and Poor Ordering of LDEV Numbers
Figure 9-2

Note

If a remote system is down or if no slaves are available to search remote partitions, those partitions are not searched and the search continues with the next partition on the list.

Specifying Partition Names for Remote Searches: Attaches to remote directories are faster and the messages received are more informative if users specify partition names in pathnames. When a partition name is included, only that partition is searched. Because partition names must be unique on Rev. 20 systems, there is no possibility of ambiguity.

If the systems within your network tend to use the same directory names, you should encourage users to supply partition names when the users are attaching to remote directories.

Attaches to remote directories are unsuccessful for the following reasons:

- The directory does not exist on the specified partition. The user receives the error message "Not found."
- The partition exists, but ATTACH cannot search it because the remote system is down. The user gets the message "Remote system down."
- The remote system is up, but no slaves are available to search for the directory. The user receives the message "No NPX slaves available."

Unexpected Attaches

The attach-scan algorithm may cause unexpected attaches, especially if two or more top-level directories (on different partitions) have the same name. The following examples illustrate two types of unexpected attaches.

Example 1: A system has two top-level directories named BLUE on local partitions, one on partition COLOR1 and another on COLOR2. COLOR1's logical device number is 1 and COLOR2's is 2. You have at least LU rights to both. You type ATTACH BLUE in an attempt to attach to <COLOR2>BLUE, but you are attached to <COLOR1>BLUE instead.

The reason for this attach is that, if the partition name is not specified, ATTACH first searches local partitions in the order of their logical device numbers and then searches remote partitions, also in the order of their logical device numbers. In this case, ATTACH searched COLOR1 first and stopped because it found a directory named BLUE.

To prevent this type of misattach, specify the partition name.

Example 2: Users KATHY and MARK, both attached to the directory <HOME>ARMCHAIR, issue the identical command ATTACH BALLPARK because they both want to attach to <BOSTON>BALLPARK. MARK, who has rights to the local partition BOSTON, is attached to <BOSTON>BALLPARK. KATHY, who has no rights to BOSTON, is attached to the remote partition <DALLAS>BALLPARK. KATHY was unaware that she had no rights to BOSTON and that it would therefore not be searched by ATTACH.

A user who encounters this type of attach should use the LIST_ACCESS command to check the ACLs on the partition's MFD and on the target directory.

PRIORITY ACLS

System Administrators and operators occasionally need special access to all files and directories on a partition. For example, they need Read access to all files to perform a backup. Special access is created by setting a priority ACL on the partition.

A priority ACL is a list of users and their access rights to a partition. Priority ACLs use the same identifiers, access rights, and formats as regular ACLs. The differences between priority ACLs and regular ACLs are as follows:

- Priority ACLs are set only on entire partitions, not on individual directories or files. (Regular ACLs cannot be set on partitions, only on directories and files.) Priority ACLs can be set both on ACL-protected and on password-protected partitions.

- Priority ACLs can be set or removed by the System Administrator from any terminal or by anyone (administrator, operator, or user) from the supervisor terminal.

- Priority ACLs take precedence over other regular ACLs on the partition.

- Priority ACLs, unlike regular ACLs, do not contain an implied $REST:NONE. To exclude all users not mentioned in the priority ACL, you must explicitly include $REST:NONE in the command line. ($REST:NONE denies $REST all access to the partition.)

- Priority ACLs are either inclusive or exclusive. An inclusive priority ACL adds some special access to the access rights that already exist on the partition. An exclusive priority ACL entirely replaces the current access rights on the partition.

Setting Priority ACLs

To set a priority ACL on a partition, use the following command format:

```
{ SET_PRIORITY_ACCESS }  partition-name   access-control-list
{ SPAC                }
```

The values for access-control-list use the same identifiers, access rights, and general formats as the SET_ACCESS and EDIT_ACCESS commands.

Inclusive ACL: As an example of setting an inclusive priority ACL, assume that a pair of analysts must do some troubleshooting on a partition named LONDON. The System Administrator issues the following command:

```
SET_PRIORITY_ACCESS LONDON HOLMES:ALL WATSON:ALL
```

This command gives the troubleshooters HOLMES and WATSON ALL rights to all directories on the partition LONDON. The rights of other users to the files and directories on LONDON are not disturbed.

Exclusive ACL: As an example of setting an exclusive priority ACL, assume that an operator has to back up the partition STAFF. Because he wants no other activity to take place on the disk at this time, he gives the following command from the supervisor terminal:

```
SET_PRIORITY_ACCESS  STAFF SYSTEM:LUR $REST:NONE
```

Only SYSTEM has any rights at all to the partition until SYSTEM removes the priority ACL with the REMOVE_PRIORITY_ACCESS command. No one else can access the partition in the meantime.

> **Caution**
>
> Use the $REST identifier carefully when setting a priority ACL, because you may unintentionally grant users more rights than they normally have on the partition.

Listing Priority ACLs

When a priority ACL is in effect for a partition, the contents of the ACL are displayed in a LIST_ACCESS command. However, because the priority ACL may prevent users from accessing any part of the partition, users can issue the LIST_PRIORITY_ACCESS command at any time to list the priority ACL on a specific partition.

The LIST_PRIORITY_ACCESS command has the following format:

```
{ LIST_PRIORITY_ACCESS } partition-name
{ LPAC                 }
```

Removing Priority ACLs

The REMOVE_PRIORITY_ACCESS command removes a priority ACL from a partition. The format of this command is as follows:

```
{ REMOVE_PRIORITY_ACCESS }  partition-name
{ RPAC                   }
```

The REMOVE_PRIORITY_ACCESS command may be given by the System Administrator from any terminal or by any user (usually an administrator or operator) from the supervisor terminal.

10 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. This chapter discusses this file and the PRIMOS CONFIG command that processes it. The chapter also explains in detail the configuration directives and the arguments that they take.

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 the same set of configuration directives for most cold starts. For this reason, 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.

Processing the System Configuration File

At cold start, the system configuration file is processed by the CONFIG command. The CONFIG command must be the first command in the system startup file that brings up PRIMOS.

The name of the system startup file is PRIMOS.COMI (C_PRMO is also used). The file is kept in the CMDNC0 directory. A template of the PRIMOS.COMI system startup file is shipped with each Prime system. (This template is shown in Chapter 8, INSTALLATION.) Use a text editor to tailor the template to suit your installation.

The CONFIG Command: The CONFIG command is recognized only during system startup. The format of the CONFIG command is as follows:

```
CONFIG -DATA config-filename
```

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

```
CONFIG -DATA CONFIG
```

At Rev. 20, two changes were made to the way the CONFIG command is processed:

- The obsolete single-line form of the CONFIG command is no longer accepted. If your system startup file contains the single-line form, the system ignores the CONFIG command. Instead, the system queries the operator for the COMDEV, PAGDEV, and NTUSR parameters, and uses the default values of the other directives.

- A CPL file named CONFIG.CPL has been added to the CMDNC0 directory. If you reexecute the PRIMOS.COMI startup file (by issuing a COMINPUT PRIMOS.COMI command at the supervisor terminal), the CONFIG.CPL file prevents the generation of an error when the system encounters the CONFIG command in the startup file. Instead, CONFIG.CPL displays the warning

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

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

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:

- Use a text editor (such as ED or EMACS) to create or modify the file.
- Do not attempt to create or modify the configuration file under PRIMOS II if the file is on a Rev. 20 disk. At Rev. 20, PRIMOS II cannot write on Rev. 20 disks. (If the file is on a pre-Rev. 20 disk, you can use NSED under Rev. 20 PRIMOS II.)
- 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. A comment line must begin with the /* characters.
- The COMDEV, NTUSR, PAGDEV, 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 below in Booting PRIMOS Without a Configuration File.

Network Information

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

The global network configuration file is created by the System Administrator with the CONFIG_NET utility. The file is usually stored in the PRIMENET* directory.

For general network information, see the PRIMENET Guide. For information on configuring and administering the PRIMENET system, see the Network Planning and Administration 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 the following:

- Out-of-range values for parameters (for example, a value of 12 in the SMLC directive, which only takes values from 00 to 07).
- Values that are correct in themselves, but conflict with other values. For example, values for NPUSR+NSLUSR+NRUSR+NTUSR must be <= '377 (decimal 255). Values of '177 for NPUSR and NTUSR, plus '20 for NRUSR and NSLUSR, though each correct in isolation, produce a sum greater than '377 and prevent the system from starting up.
- Decimal numbers used by mistake for octal numbers.

Appendix B, PRIMOS COLD START ERROR MESSAGES, contains a list of error messages produced by erroneous directives.

Booting PRIMOS Without a Configuration File

At Rev. 20, the procedure for booting PRIMOS was simplified. Your CPU handbook describes the various boot methods, including autoboot and manual boot.

Manual boot is used when you cannot use your configuration file to bring up PRIMOS. (Two frequent reasons are an incorrect configuration directive and a move of the COMDEV disk pack to another disk drive or disk controller.)

Prior to Rev. 20, if you could not modify the configuration file under PRIMOS, you used NSED under PRIMOS II to edit the file. You can still use NSED under PRIMOS II at Rev. 20 if the file is on a pre-Rev. 20 disk. However, if the file is on a Rev. 20-format disk, you cannot use PRIMOS II because it cannot write on a Rev. 20 disk.

Use the following procedure to change the configuration file using the manual boot:

1. At the CP> prompt, use the VCP 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 114134).

3. Enter the COMDEV, PAGDEV, and NTUSR parameters at the appropriate prompts. PRIMOS comes up after this step, but the PRIMOS.COMI file is not executed.

4. Use NSED to edit the configuration file.

5. To reconfigure the system with the new configuration parameters, shut down PRIMOS and reboot it without the '100000 boot option.

Adding the '100000 value to the BOOT option word (at Step 2) instructs PRIMOS to do the following:

- Ignore the configuration file.
- Query the operator for the COMDEV, PAGDEV, and NTUSR parameters.
- Use default values for the other configuration parameters.
- Do not execute the PRIMOS.COMI system startup file.
- Do not initialize the communication controllers.

For details and examples of how to boot PRIMOS manually, see your CPU handbook.

CONFIGURATION DIRECTIVES

This section describes the directives used in configuration files. See also Chapter 2, PLANNING THE SYSTEM CONFIGURATION, for more information and guidelines on these directives.

▶ ABBREV $\begin{Bmatrix} \text{YES} \\ \text{NO} \end{Bmatrix}$

Controls user abbreviations.

YES enables the abbreviation processor, thus allowing users to create abbreviations (using the ABBREV command) and store them in abbreviation files. YES is the default.

NO prohibits the creation of user abbreviations.

▶ ALTDEV pdev [records]

Specifies the alternate paging partition and, optionally, its size.

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 <u>records</u> 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 I/O buffer.

The arguments have the following values and meanings:

line — The number of the asynchronous line for which buffer sizes are to be set. For terminal users, this value is the physical line number. For assignable lines, this value is NTUSR + NRUSR - 1 for the first assignable line, NTUSR + NRUSR for the second, and so on up to NTUSR + NRUSR + NAMLC - 2 for the last assignable line. Use the actual line number to change the size of the DMQ buffer on an assignable line. An invalid line number produces the message "BAD LINE # IN AMLBUF COMMAND."

in-buff-size — The size of the input buffer, in halfwords (two characters per halfword). The minimum value is 1 and the maximum is '7777 (4095 decimal). The default is '200 (128 decimal). If 0 is specified, the buffer size remains at its previously set value (which is either the default size or the size set by a previous AMLBUF directive).

out-buff-size — The size of the output buffer, in halfwords (two characters per halfword). The minimum value is '62 (50 decimal) and the maximum is '7777 (4095 decimal). The default is '300 (192 decimal). If 0 is specified, the buffer size remains at its previously set value (which is either the default size or the size set by a previous AMLBUF directive).

dmq-size The size in halfwords (one character per halfword) of the DMQ AMLC or ICS buffer (only meaningful if the corresponding line is on a DMQ AMLC or ICS controller). Valid values, which must be a power of 2, are '20 (16 decimal; the minimum value), '40 (32 decimal; the default value), '100 (64 decimal), '200 (128 decimal), '400 (256 decimal), '1000 (512 decimal), and '2000 (1024 decimal; the maximum value). If 0 is specified, the buffer size remains at its previously set value (which is either the default size or the size set by a previous AMLBUF directive).

The AMLBUF directive cannot modify input or output buffer sizes for remote users. Use the REMBUF directive instead.

Total size of input buffers plus output buffers cannot exceed 768,000 (decimal) halfwords. Exceeding this limit produces the message "No room. AMLBUF (TFLADJ)".

No individual buffer can exceed '7777 (4095 decimal) halfwords. The total size of the DMQ buffers cannot exceed 64,000 (decimal) halfwords. Failure to meet these specifications produces error messages.

See Chapter 11, CONFIGURING ASYNCHRONOUS LINES, for details on configuring AMLC buffers.

▶ AMLCLK baudrate

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

The value specified for baudrate must be no less than '35 (29 decimal) and no greater than '45400 (19200 decimal). The default is '22600 (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 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. 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 (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 Chapter 11, CONFIGURING ASYNCHRONOUS LINES, for more information on configuring asynchronous lines.

► AMLTIM [ticks [disctime [gracetime]]]

Sets time intervals for the three variable event timers.

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 period, PRIMOS checks each line for carrier loss. If a loss has occurred and the DISLOG directive is set, the process is logged out. The value for ticks must be greater than 0. The default is 2 (0.2 seconds).

disctime   The time period (in tenths of a second) when the DTR signal is again raised on all lines. A modem that was forced to be inactive because the DTR signal was dropped will again be able to establish an active carrier. (The most common reason for a carrier to be inactive occurs when a dialup line has been hung up.) 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 that value. The default is '3410 (1800 decimal, which is 3 minutes).

gracetime The minimum grace period (in tenths of a second) for terminal lines that have active carriers but are not connected to processes that are logged in. gracetime in effect defines the minimum time for a caller to establish itself with a logged-in process. (The actual grace period varies from gracetime to twice gracetime.) The default value of 0 disables the grace period. The specified value (if not 0) must be greater than ticks and is truncated to the nearest multiple of ticks.

Note

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. Set the value for disctime to at least twice that of ticks, preferably larger than '400. Set a value of '1000 for gracetime. ASD allows a portion of gracetime to elapse before a user logs in. 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.

▶ ASRATE rate

Sets the baud rate of the supervisor terminal.

rate is one of the following four octal integers:

rate	Baud rate (decimal)
110	110
1010	300 (default)
2010	1200
3410	9600

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.

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 (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 (192 decimal). The minimum value (other than 0) is '100 (64 decimal). If 0 is specified, the buffer size remains at its previously set value (which is usually the default size).

You may want to enlarge out-buff-size if you have the following situation: a 300-1200 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.

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 AMLC lines and are not using ASD, do not use the ASYNC JUMPER directive because it does not affect the lines.
- 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.

For a more detailed discussion of the choice of line speeds, see the description of this directive in Chapter 2, PLANNING THE SYSTEM CONFIGURATION.

Note

> The ASYNC JUMPER directive replaces the obsolete ICS JUMPER directive. If the configuration file contains the ICS JUMPER directive, the directive is obeyed but a warning message is displayed. (See the ICS JUMPER directive below for the content of this message.) If the configuration file contains both ASYNC JUMPER and ICS JUMPER directives, only the last one in the file is used to configure the system. It is recommended that you use only the ASYNC JUMPER directive in the configuration file.

Setting Speeds With the SET_ASYNC Command: Set speeds for ICS lines with the SET_ASYNC -LINE n -SPEED option. For assignable lines, use SET_ASYNC -LINE n -SPEED -ASGN. The AMLCLK directive defines program speed and defaults to 9600 baud.

Valid speeds are as follows:

Speed	
110	
134.5	
300	
1200	
program	(defined by AMLCLK directive, default is 9600)
speeda	*
speedb	*
speedc	*

* These jumper speeds are used either to set the baud rate for ICS lines or to indicate the hardware speeds set on the AMLC boards for AMLC lines.

▶ COMDEV pdev

Specifies pdev as the physical device number of the partition on which the system UFD CMDNC0 resides.

The partition specified by this directive becomes the system command device. The COMDEV directive must be specified in the configuration file.

▶ $\text{DISLOG} \left\{ \begin{array}{l} \text{YES} \\ \text{NO} \\ \text{line\_number} \end{array} \right\}$

Enables or disables automatic logout when a line is disconnected. DISLOG followed by a line number enables DISLOG on the selected line.

A line is defined as disconnected when the carrier detect signal goes logically low.

YES logs out any user whose line is disconnected. line_number logs out only the user of that line if that line is disconnected. (You can specify only one line number per each DISLOG directive, but you can specify as many DISLOG directives as you need.) NO does not log out the user of any line if the user's line is disconnected. NO is the default.

The DISLOG directive is useful for installations with port selectors or dialup modems. Specify DISLOG YES or DISLOG line_number if you are using Auto Speed Detect (ASD).

The following restrictions apply to the DISLOG directive. One DISLOG directive that follows another cannot override that first directive.

For example, DISLOG YES followed by DISLOG NO causes every line to be set for DISLOG. Also a DISLOG directive that sets all lines takes precedence over a directive that sets a single line. For example, when the DISLOG directive consists of several per-line directives and a DISLOG YES directive, DISLOG is enabled on every line.

▶ 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 $\left\{\begin{array}{l}\text{character}\\\text{octal-value}\end{array}\right\}$

Sets the system default erase character.

character is any printing ASCII character.

octal-value is the octal value of any ASCII character.

The default erase character is the double quotation mark ("), which is '242. Both of the following examples set the system default erase character to the exclamation mark (!):

```
ERASE !
ERASE 241
```

Use ERASE 210 to set the erase character to the BACKSPACE key.

▶ 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 (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 (decimal 32,762).

If the FILUNT directive is not specified in the configuration file, the default value is used.

▶ 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 ICS2/3 controller.

The ICS CARDS directive causes PRIMOS to check the actual configuration of an ICS2/3 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 Appendix B of this book, PRIMOS COLD START ERROR MESSAGES.) The differences are due to unexpected, faulty, or missing LACs.

Whether or not you use the ICS CARDS directive, ICS2/3 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 ICS2/3 asynchronous configurations have not changed.

If the ICS CARDS directive is omitted for any ICS2/3 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 ICS2/3 controller. Valid values are '10, '11, '36, and '37.

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.

Use the following two-step process to calculate config-word:

1. Compose a sixteen-bit binary integer where each bit represents a slot in the ICS2/3. If a slot contains an asynchronous LAC, enter 1 in that position. If no LAC or a synchronous LAC is present, enter zero.

2. Convert this binary number to octal.

Consider, for example, the following directive:

```
ICS CARDS  10  114764
```

This example defines the asynchronous LAC configuration for the ICS2/3 controller with device-address of '10.

Derive the config-word of 114764 as follows:

config-word bit number	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
slot number on card cage	6	7	8	9	10	11	12	13	19	20	21	22	23	24	25	26
1 if async LAC in slot	1	0	0	1	1	0	0	1	1	1	1	1	0	1	0	0
config-word	1			1			4			7			6			4

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

▶ ICS INPQSZ queuesize

Sets the size of all the ICS input queues.

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 less than '1777 and 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 Chapter 11, CONFIGURING ASYNCHRONOUS LINES, for further details on configuring ICS lines.

▶ ICS INTRPT rate

Sets the async interrupt rate for ICS controllers.

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 AMLC Baud Rate column in the table shows what the last AMLC line would be set at for equivalent performance.

Number of Interrupts/Second		Interrupt Rate (ms)	Equivalent AMLC 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.

▶ ICS JUMPER speeda speedb speedc

The ICS JUMPER directive is obsolete at Rev. 20, having been replaced by the ASYNC JUMPER directive, discussed elsewhere in this chapter. The use and values for ICS JUMPER are the same as for ASYNC JUMPER.

If the configuration file contains the ICS JUMPER directive, the directive is obeyed but the following warning message is displayed:

```
Warning:  ICS JUMPER directive will be phased out.
Please use ASYNC JUMPER instead.  (CINIT)
```

If the configuration file contains both ASYNC JUMPER and ICS JUMPER directives, only the last one in the file is used to configure the system.

It is recommended that you use only the ASYNC JUMPER directive in the configuration file.

```
▶ 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. For example, KILL 237 sets the system default kill character to the DELETE key.

The default kill character is the question mark (?), which is '277.

```
▶ 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.

▶ LOGREC value

Enables or disables system event logging.

If *value* is set either to 0 (the default) or to a positive number, event logging is enabled. If *value* is set to a negative number (177777), event logging is disabled. (Disable system event logging when running a write-protected disk.)

Prior to Rev. 19, a positive argument set the maximum size of the event logging file. At Revs. 19 and later, a positive value enables event logging (the size of the event logging file can be controlled by the more general quota system) and causes the following message to be displayed at the supervisor terminal:

```
LOGREC CONFIG DIRECTIVE NO LONGER SETS A QUOTA ON THE
SYSTEM EVENT LOGGING FILE.
PLEASE USE 'SET_QUOTA'.  (CINIT)
```

See Chapter 17, SYSTEM MONITORING, for more information on system event logging and the EVENT_LOG command. See the NETREC directive later in this chapter for controlling network event logging.

▶ LOTLIM minutes

Specifies login time limit in minutes.

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.

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 (1000 decimal minutes, which is 16 hours and 40 minutes).

▶ MAXPAG number-of-pages

Specifies the number of pages of physical memory (starting from page 0) to be used after cold start.

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 (256 decimal) and '40000 (16384 decimal). If number-of-pages is not specified or if MAXPAG is not in the configuration file, all available memory is used.

Use the following table as a reference for setting MAXPAG.

Memory (MBytes)	Memory (pages)	MAXPAG argument	Memory (MBytes)	Memory (pages)	MAXPAG argument
10	5120	'12000	32	16384	'40000
8	4096	'10000	28	14336	'34000
6	3072	'6000	24	12288	'30000
4	2048	'4000	20	10240	'24000
3	1536	'3000	16	8192	'20000
2	1024	'2000	14	7168	'16000
1	512	'1000	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.

▶ MEMHLT $\begin{Bmatrix} \text{YES} \\ \text{NO} \end{Bmatrix}$

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 what user process caused 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 user process to be logged out.

- The process that is running 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.

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.

▶ NAMLC number-of-buffers

Sets the number of buffers to be reserved for assigned asynchronous lines.

number-of-buffers is the number of buffers to be reserved for assigned asynchronous lines. The default is 0. NAMLC + NTUSR must be less than or equal to '377 (256 decimal).

number-of-buffers equals the maximum number of lines that may be assigned simultaneously on your system. The AMLC command, described in Chapter 11, CONFIGURING ASYNCHRONOUS LINES, specifies which lines are assignable.

▶ NETREC value

Enables/disables network event logging.

If value is set to either 0 (the default) or to a positive number, network event logging is enabled. If value is set to a negative number (177777), network event logging is disabled. (Disable network event logging when running a write-protected disk.)

Prior to Rev. 19, a positive argument set the maximum size of the network event logging file. At Revs. 19 and later, a positive value enables event logging (the size of the file can be controlled by the more general quota system) and causes the following message to be displayed at the supervisor terminal:

```
NETREC CONFIG DIRECTIVE NO LONGER SETS A QUOTA ON THE
NETWORK EVENT LOGGING FILE.
PLEASE USE 'SET_QUOTA'.  (CINIT)
```

See Chapter 17, SYSTEM MONITORING, for more information on network event logging and the EVENT_LOG command. See the LOGREC directive earlier in this chapter for more information about controlling system event logging.

▶ NLBUF buffers

Specifies the number of LOCATE buffers to be configured.

buffers is the number of LOCATE buffers for which the system is to be configured. Minimum value is '10 (8 decimal) and maximum is '400 (256 decimal). The default is '100 (64 decimal).

▶ NPUSR number

Sets the number of phantom users.

number is the number of phantom users for which the system is to be configured. (number must be a nonnegative octal integer.) The default is 1. The maximum is '377 (255 decimal) minus the number of terminal, slave, and remote users (NTUSR, NSLUSR, and NRUSR).

You must configure a phantom for the Login Server. The Login Server cannot run when NPUSR is 0. 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 spoolers, the Batch subsystem, the File Transfer Service (FTS), and other purposes.

▶ NRUSR number

Specifies the number of processes to be reserved for remote logins across the PRIMENET network.

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 '77 (63 decimal). NTUSR + NPUSR + NRUSR + NSLUSR cannot exceed '377 (255 decimal). NRUSR need not be used for terminals connected through modems to asynchronous AMLC or ICS lines.

Note

> Although you can configure up to 126 remote and slave users (63 of each), only a maximum of 63 remote and/or slave users can be active at one time. An attempt to exceed that limit produces an error message stating that the resource is temporarily unavailable.

▶ NSEG number

Specifies the total virtual address space for a system.

number specifies the number of page maps to be allocated during system initialization. The maximum value is '20000 (8192 decimal). The default is '1776 (1022 decimal).

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 signalled 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 signalled.

▶ NSLUSR number

Sets the number of slave processes (users).

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 255 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 '77 (63 decimal). NTUSR + NPUSR + NRUSR + NSLUSR must be less than or equal to '377 (255 decimal).

Note

> Although you can configure up to 126 remote and slave users (63 of each), only a maximum of 63 remote and/or slave users can be active at one time. An attempt to exceed that limit produces an error message stating that the resource is temporarily unavailable.

▶ NTUSR number

Specifies number of terminal users, including the supervisor terminal.

number is the number of terminal (local system) users for which the system is to be configured. The minimum value of number is '2 and the maximum is '377 (255 decimal). NTUSR, which has no default value, must be included in the configuration file.

NTUSR is added to NPUSR, NSLUSR, and NRUSR to determine the total number of users configured on the system. NTUSR + NPUSR + NRUSR + NSLUSR must be less than or equal to '377 (255 decimal). NTUSR + NAMLC must also be less than or equal to '377 (255 decimal).

▶ NVMFS number

Sets the number of VMFA (Virtual Memory File Access) dynamic segments available in virtual address space for the system.

The maximum value for number is '2000 (1024 decimal). The default is '144 (100 decimal).

The total number of segments available for both NSEG and NVMFS is '20000 (8192 decimal). If you specify values for NSEG and NVMFS that total higher than the '20000 (8192 decimal) maximum, NVMFS takes precedence over NSEG. For example, if you specify NSEG as '17500 (8000 decimal) and specify NVMFS as '454 (300 decimal), the system is configured with '454 (300 decimal) NVMFS segments available and '17324 (7892 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."

▶ PAGDEV pdev [records]

Specifies the paging partition and, optionally, its size.

PAGDEV must be included in the configuration file. 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 signalled for that user.

See Chapter 4, DISKS AND TAPE DRIVES, for details on paging partitions.

▶ PRATIO n

Specifies the ratio of use of the alternate paging partition in relation to the primary paging partition.

n is an octal integer ranging from 0 to '12 (10 decimal). n represents the approximate number of times the alternate paging partition is to be used for each 10 times that paging space is allocated. For example, a setting of 2 causes the alternate paging partition to be used approximately 20% of the time and the primary paging partition to be used approximately 80% of the time.

The default for n is 5, which that means the alternate paging partition is used approximately 50% of the time.

If n is set to 0, the alternate paging partition is used only when the primary paging partition becomes full. If n is set to '12 (10 decimal), the primary paging partition is used only when the alternate paging partition becomes full. (However, certain portions of PRIMOS are always allocated on the primary paging partition when there is sufficient room at system startup.)

The PRATIO directive is ignored if the configuration file does not contain the ALTDEV directive. Use the PRATIO directive only to improve system performance.

▶ PREPAG pages

Specifies the number of pages to prepage out when a page fault occurs.

pages cannot be less than '1 or more than the number of memory pages available for paging. The default of pages is '3.

The PREPAGE directive is used only to improve system performance.

▶ REMBUF in-buff-size out-buff-size

Sets the sizes of the terminal input and output buffers for remote users.

The arguments have the following values and meanings:

in-buff-size	The terminal input buffer size in halfwords (two characters per halfword). The default and minimum values are both '202 halfwords (260 bytes decimal). If 0 is specified, the buffer size remains at its previously set value (which is usually the default size).
out-buff-size	The terminal output buffer size in halfwords (two characters per halfword). The default and minimum values are both '101 halfwords (130 bytes decimal). If 0 is specified, the buffer size remains at its previously set value (which is usually the default size).

Total size of all input buffers plus all output buffers cannot exceed '2734000 (768,000 decimal) halfwords.

On systems using block-mode terminals, you may improve throughput by increasing the size of the input buffer to '404 halfwords (520 bytes decimal). You may also improve throughput on systems where users are logging in remotely across a ring network by increasing the size of the output buffer to '402 halfwords (516 bytes decimal).

For remote login over public data networks (PDNs), you must set in-buff-size and out-buff-size to the octal value of the packet size that you configured with CONFIG_NET. For example, a packet size of '200 (128 decimal) bytes requires that both in-buff-size and out-buff-size be set to '200.

▶ RWLOCK value

Specifies the setting of the file system read/write lock.

value is 0, 1, or 3. The default is 1. value has the following meanings:

0	1 reader or 1 writer (writer has exclusive control)
1	N readers or 1 writer (writer has exclusive control)
3	N readers and 1 writer

Caution

Many subsystems and utilities will not work correctly if the system default RWLOCK is not set to 1. To change the read/write lock of a file or set of files, use the PRIMOS RWLOCK command instead of this directive.

▶ SMLC ON

Configures synchronous communication drivers for all synchronous communication controllers, including the ICS2/3.

SMLC ON must be specified when synchronous lines are used for products other than PRIMENET. At Rev. 20, SYNC ON is a synonym for SMLC ON. The default configuration for SMLC ON is given below.

Logical Line Number	Logical Controller	Controller Address	Controller Physical Line Number
00	0	'50	0
01	0	'50	1
02	0	'50	2
03	0	'50	3
04	1	'100000	0
05	1	'100000	1
06	1	'100000	2
07	1	'100000	3

Logical lines 00 to 03 are mapped to logical controller 0. The physical device address is '50, with physical line numbers 0 to 3. (Note that the MDLC device address is usually '50 or '51.)

Logical lines 04 to 07 are mapped to physical lines 0 to 3 on logical controller 1. Controller 1's physical address is '100000, indicating that the controller is disabled. To enable controller 1, set its address to a valid device address ('50 or '51) with the SMLC CNTRLR directive.

You can change the default configuration with the SMLC CNTRLR and SMLC SMLCnn directives, which are described below. SMLC CNTRLR changes the mapping of logical controller to physical address. SMLC SMLCnn changes the mapping of a single logical line number.

You may use either directive separately. If you need both directives, you must give the SMLC CNTRLR directive(s) first. In other words, you must assign the controller its correct physical address before you assign any new SMLC lines to it.

Always specify synchronous line configuration explicitly. Do not rely on the default configuration for lines to be used or for lines to be left unused.

Be sure to override the default configuration when synchronous lines are configured on two controllers, one of which is not device address '50.

▶ SMLC CNTRLR controller-number [device-address] [protocol]

Assigns a physical controller address to a logical controller number with a particular protocol. This directive must be given before any SMLC SMLCnn directive. It enables an ICS2/3 to handle synchronous communications. At Rev. 20, SYNC CNTRLR is a synonym for SMLC CNTRLR.

The arguments have the following values and meanings:

controller-number — The logical controller. Valid values are 0, 1, or 7. Any other number produces the error message "BAD SMLC CONTROLLER MAPPING COMMAND." The value 7 is for SDLC or ASYNC_SDLC only and allows multiple controllers (that only support SDLC) to be configured. You can have more than one controller-number 7.

device-address — The physical device address of the specified controller given in octal. Usually '10 or '11 is given for an ICS2/3, and '36 or '37 for an ICS1. The default values are address '50 for controller 0 and '100000 (disabled address) for controller 1. Note that device address '50 is usually an MDLC. If controller 1 is used, its address should not conflict with the address of any other peripheral controller.

protocol	This field selects the correct downline load file for the specified ICS2/3 and protocols. Use this field only with an ICS2/3 controller. See Tables 10-1 and 10-2 for valid protocol combinations.

Note

Entering a protocol token for a controller other than an ICS2/3 controller causes the following error message to be displayed:

```
Error:  Controller xx does not support sync protocols
        (COMINI)
```

xx is the device address of the controller.

If the protocol token is omitted, the following message appears:

```
Error:  protocol combination not supported on ICS2
        device address dd (BTPOC).
```

If the SMLC CNRTLR directive is omitted entirely, the default protocol (ASYNC) is loaded.

If you map one logical controller to a previously mapped address, SMLC automatically disables the previously mapped controller (without warning), setting its address to '100000. A new mapping directive enables the disabled controller. For example, the following directives map controller 1 to address '10 and controller 0 to address '11:

```
SMLC CNTRLR 1 10 ASYNC_HDLC
SMLC CNTRLR 0 11 BSCRJE_BSCX25
```

Appropriate downline load files to support the specified protocol combination are selected with the protocol field. Note that the second example above disables the default configuration for '50.

You can also disable a logical controller by setting its address to blank or to '100000. For example, either of the following directives can disable controller 0:

```
SMLC CNTRLR 0 100000
SMLC CNTRLR 0
```

Valid Protocol Token Combinations

Valid protocol tokens are ASYNC, SDLC, HDLC, BSCX25, and BSCRJE. You need not enter a token combination in the order shown in Tables 10-1 and 10-2, but the parts must be separated by an underscore.

ICS2: Table 10-1 lists valid protocol token combinations, microcode image names, and downline load file numbers for the ICS2. The ICS2 cannot support all protocols concurrently because of limitations to memory size.

Note

> Asynchronous Reverse Flow Control is not enabled for an ICS2 controller when a combination of BSC and ASYNC protocols is selected (as in BSCRJE_BSCX25_ASYNC, BSCRJE_ASYNC, or ASYNC_BSCX25). If you use the AMLC command to attempt to enable Reverse Flow Control in this situation, the ICS2 ignores the request and PRIMOS returns an error message.

ICS3: The ICS3 is available with 256KB, 512KB, and 1024KB of RAM. In the 512KB ICS3, any combination of protocol tokens is allowed. When all protocols are required, the downline file ICS3_08.DL is loaded. The 256KB ICS3 can handle all protocol combinations except the two largest combinations, ASYNC_SDLC_HDLC_BSCX25_BSCRJE and SDLC_BSCX25_BSCRJE.

Any combination not containing SDLC causes downline load file ICS3_09.DL to be loaded. This file contains all protocols except SDLC. For any combination containing SDLC, the loaded file contains only SDLC and the other protocol(s), resulting in more economical running than when loading the "all protocols" file, ICS3_08.DL.

Table 10-2 lists the protocol tokens, the microcode image names and the downline load file numbers used for the ICS3.

To conserve memory space, you can delete unused downline load files and maps from the DOWN_LINE_LOAD* directory.

Table 10-1
ICS2 Valid Protocol Token Combinations

Protocol Token	Microcode Image*	DLL-File Number
ASYNC	HSAXX	ICS2_01.DL
SDLC	HSAXX	ICS2_02.DL
HDLC	HSAXX	ICS2_03.DL
BSCRJE_BSCX25	HSBXX	ICS2_04.DL
BSCX25	HSBXX	ICS2_04.DL
ASYNC_SDLC	HSAXX	ICS2_05.DL
ASYNC_HDLC	HSAXX	ICS2_06.DL
HDLC_SDLC	HSAXX	ICS2_07.DL
BSCRJE_BSCX25_ASYNC (Note 1)	ABXX	ICS2_08.DL
ASYNC_BSCX25 (Note 1)	ABXX	ICS2_08.DL
BSCRJE_BSCX25_SDLC	HSBXX	ICS2_09.DL
BSCRJE_SDLC	HSBXX	ICS2_09.DL
BSCX25_SDLC	HSBXX	ICS2_09.DL
BSCRJE_BSCX25_HDLC	HSBXX	ICS2_10.DL
BSCRJE_HDLC	HSBXX	ICS2_10.DL
ASYNC_HDLC_SDLC	HSAXX	ICS2_11.DL
BSCRJE	HSBXX	ICS2_12.DL
BSCRJE_ASYNC (Note 1)	ABXX	ICS2_13.DL
BSCX25_HDLC	HSAXX	ICS2_14.DL
BSCRJE_BSCX25_HDLC_SDLC	HSAXX	ICS2_15.DL
BSCX25_HDLC_SDLC	HSAXX	ICS2_15.DL
BSCRJE_HDLC_SDLC	HSAXX	ICS2_15.DL

* XX is the version number.

Table 10-2
ICS3 Valid Protocol Token Combinations

Protocol Token	Microcode Image*	DLL-File Number
SDLC	ABHSXX	ICS3_01.DLL
ASYNC_SDLC	ABHSXX	ICS3_02.DLL
SDLC_HDLC	ABHSXX	ICS3_03.DLL
SDLC_BSCX25	ABHSXX	ICS3_03.DLL
SDLC_HDLC_BSCX25	ABHSXX	ICS3_03.DLL
ASYNC_SDLC_HDLC	ABHSXX	ICS3_04.DLL
ASYNC_SDLC_BSCX25	ABHSXX	ICS3_04.DLL
ASYNC_SDLC_BSCX25_HDLC	ABHSXX	ICS3_04.DLL
SDLC_BSCRJE	ABHSXX	ICS3_05.DLL
ASYNC_SDLC_BSCRJE	ABHSXX	ICS3_06.DLL
SDLC_HDLC_BSCRJE	ABHSXX	ICS3_07.DLL
SDLC_BSCX25_BSCRJE	ABHSXX	ICS3_07.DLL
SDLC_HDLC_BSCX25_BSCRJE	ABHSXX	ICS3_07.DLL
ASYNC_SDLC_HDLC_BSCRJE_BSCX25	ABHSXX	ICS3_08.DLL
ASYNC_SDLC_BSCRJE_BSCX25	ABHSXX	ICS3_08.DLL
ASYNC_SDLC_HDLC_BSCRJE	ABHSXX	ICS3_08.DLL
ASYNC_HDLC_BSCX25_BSCRJE	ABHSXX	ICS3_09.DLL

* XX is the version number.

▶ SMLC SMLCnn [controller-number [line-number]]

Maps a logical line number to a physical line number on a specified logical controller.

SMLC SMLCnn is used for X.25/RJE but, not for SDLC. At Rev. 20, SYNC SYNCnn is a synonym for SMLC SMLCnn. The arguments have the following values and meanings:

nn — The logical line number. The values are 00 to 07.

controller-number — The logical controller set by a SMLC CNTRLR directive. The values are 0, 1, or '100000. Use 0 or 1 to identify a controller. Use '100000 when the specified line is not to be configured or allocated memory. (7 is not accepted.) The default is '100000.

line-number — The physical line number of the specified controller onto which the logical line number is mapped. (For ICS2/3 sync LAC physical line numbering, see the *ICS User's Guide*.) If the controller is an ICS1, *line-number* must be 0. The default values map SMLC00 to SMLC03 to physical lines 0 to 3 on the first controller, and SMLC04 to SMLC07 to physical lines 0 to 3 on the second controller. This value must be specified unless *controller-number* is '100000 or is unspecified.

For example, the following directive assigns logical line 4 to physical line 3 on controller 1:

```
SMLC SMLC04 1 3
```

Setting the controller number either to a blank or to '100000 disables a logical line number. For example, either of the following directives disables logical line 07:

```
SMLC SMLC07
SMLC SMLC07 100000
```

Giving any value for controller-number other than 0, 1, blank, or 100000, produces the following error message:

```
BAD SMLC LINE MAPPING COMMAND
```

▶ SMLC DSC line strap proc recv

Specifies data set control (DSC) information used by DPTX/BSCMAN for a logical line provided by an SMLC, HSSMLC, or MDLC controller.

Use SMLC DSC for DPTX/BSCMAN only. If specified, SMLC DSC must appear after any SMLC SMLCnn directives.

At Rev. 20, SYNC DSC is a synonym for SMLC DSC.

The arguments have the following values and meanings:

line — The logical line number (00-07) represented by nn in the SMLC SMLCnn directive.

strap — A bit pattern that indicates specific data set signals to be strapped on by the software. The bit patterns are as follows:

1 Data Terminal Ready (DTR). The default is 1.

2 Request to Send (RTS)

In addition, speed select in Europe is specified with the '10 bit:

Set to 1 - select fast data set

Reset to 0 - select slow data set

proc — The data set control procedure to be used for transmitting data. The procedures are as follows:

1 No data set orders. Usually used with DTR and RTS strapped on, with modems used for four-wire full-duplex service.

2 Use data set orders as follows: Issue RTS, wait for CLEAR to send (CTS), send, drop RTS. Usually used with most half-duplex modems. The default is 2.

3 Use data set orders as follows: Wait for .NOT. Carrier Detect (CD), issue RTS, wait for CTS, send, drop RTS. Rarely used, but may be necessary with 201-series modems only if lines are very noisy. Try 2 first.

recv Indicates whether the receiver is to be turned on before or after transmitting. The settings are as follows:

0 Turn on receiver before transmitting. This setting provides a faster response and should be used if possible. The default is 0.

1 Turn on receiver after transmitting. This setting must be used with two-wire 201-series modems. This setting may be tried on other two-wire systems only if problems appear that cannot be solved by other means.

The default setup, if no DSC is specified, is the equivalent of including the following line in the configuration file:

```
SMLC DSC line 1 2 0
```

▶
```
SYNC ON
SYNC CNTRLR controller-number [device-address] [protocol]
SYNC SYNCnn [controller-number line-number]]
SYNC DSC line strap proc recv
```

At Rev. 20, SYNC is a synonym for SMLC. For details on the SYNC directives, see the preceding discussions of the SMLC directives.

▶ TYPOUT $\left\{\begin{array}{l}\text{YES}\\\text{NO}\end{array}\right\}$

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 use any number of TYPOUT directives in the configuration file to display selected directives.

▶ UPS number

Controls restart after a power failure.

An Uninterruptible Power Supply (UPS) maintains power to the CPU and memory during a power failure and then 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 automatic 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 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 (64 decimal seconds) is recommended for a storage module.

▶ VPSD

Wires the kernel VPSD debugger into PRIMOS at system startup time.

The VPSD debugger is used for debugging the operating system, but is not used in a normal production environment. Most installations will not want this debugger enabled during normal use.

▶ 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.

11
Configuring Asynchronous Lines

Asynchronous communication permits the transfer of data to and from devices. This chapter discusses the hardware and software that support asynchronous communication. The first part of the chapter describes Prime's AMLC and ICS controllers and the asynchronous lines attached to them. The second part introduces two new commands, SET_ASYNC and CONVERT_AMLC_COMMANDS, and shows how to use them to configure your lines. The final part discusses allocation of buffers during system configuration.

For information on configuring synchronous lines for devices that use half-duplex or packet data exhanges, see the ICS User's Guide and the explanation of the SMLC directives in Chapter 10 of this book, CONFIGURATION DIRECTIVES.

Note

> At Rev. 20.2, the SET_ASYNC command replaces the AMLC command. SET_ASYNC provides a more straightforward way of configuring asynchronous lines. Although the AMLC command is still supported, its use is no longer recommended. For information on the AMLC command, see Appendix D, OBSOLETE AND RARELY USED COMMANDS AND DIRECTIVES.

ASYNCHRONOUS LINES

Asynchronous lines connect terminals, modems, and peripheral devices, such as printers and plotters, to your system. Figure 11-1 shows a typical system and the types of devices that communicate via asynchronous lines.

Controllers and Interface Standards

The following Prime communications controllers support asynchronous lines:

- AMLC (Asynchronous Multi-Line Controller) all models
- ICS1 (Intelligent Communications Subsystem 1)
- ICS2 (Intelligent Communications Subsystem 2)
- ICS3 (Intelligent Communications Subsystem 3)

Prime asynchronous communications controllers support two communications interface standards: RS-232-C (also called CCITT V.24 in Europe) and 20 milliamp current loop for AMLC controllers.

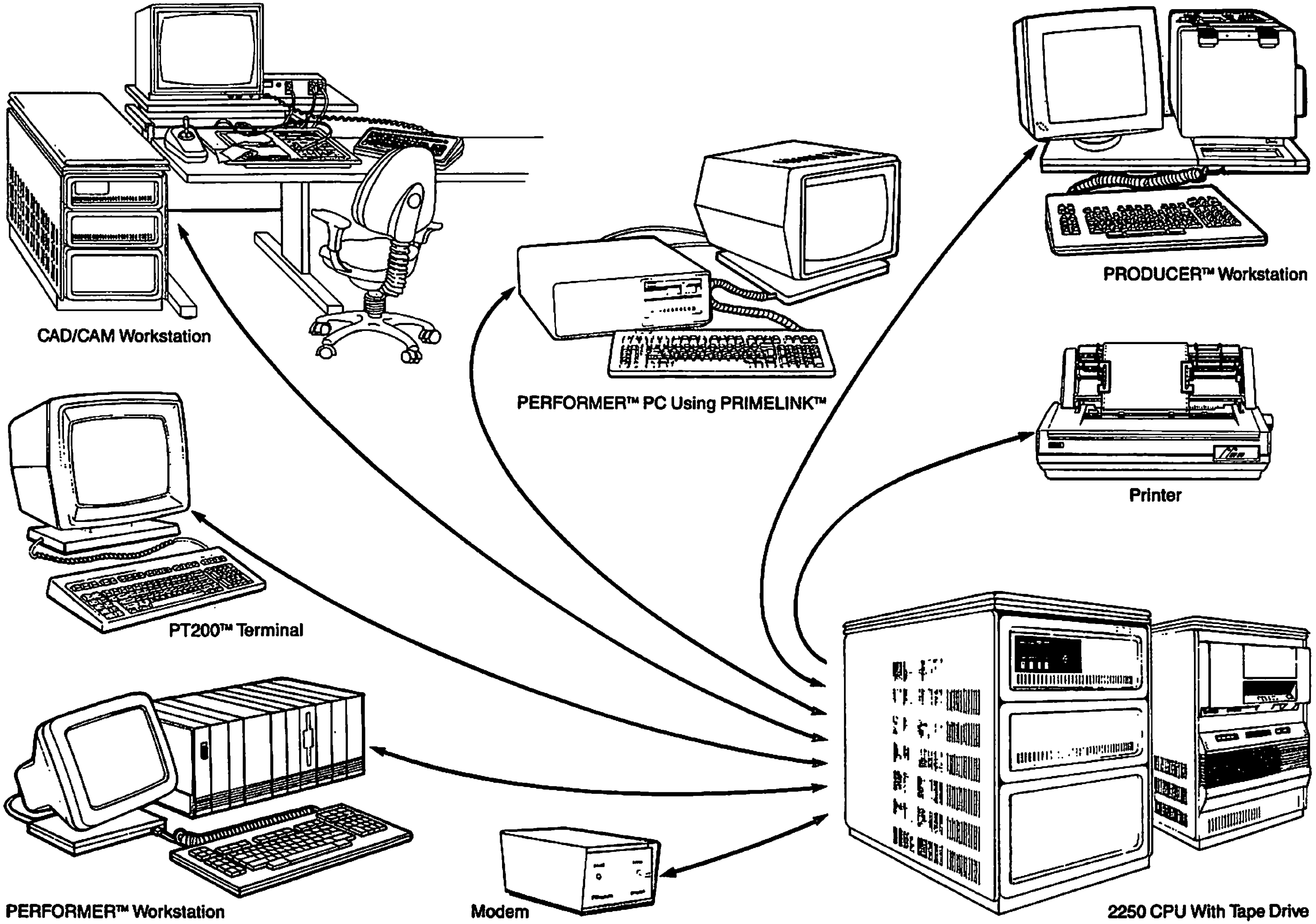
RS-232-C/CCITT V.24: RS-232-C/CCITT V.24 is an industrywide communications standard that describes a communications interface. It is used to support many terminal and modem types, including all Prime products that use asynchronous lines. RS-232-C is the standard adopted by manufacturers in the U.S. and Canada; CCITT V.24 is the equivalent European standard.

20 Milliamp Current Loop: The 20 milliamp current loop is a communications interface that supports other slower or older terminals and devices. Some Prime terminals (such as the PT25™, PT45, and PT65™ terminals) are supported on RS-232-C/CCITT V.24 and 20 milliamp current loop lines.

Number of Lines Supported

The maximum number of asynchronous lines that can exist on a system is currently 256. One of these lines is reserved for the supervisor terminal. The remaining 255 lines can be distributed between AMLC and ICS controllers and connected to terminals and other devices.

Current models of AMLC controllers support a maximum of 16 lines each. A maximum of 8 AMLC controllers can exist on a system, for a total of 128 lines.

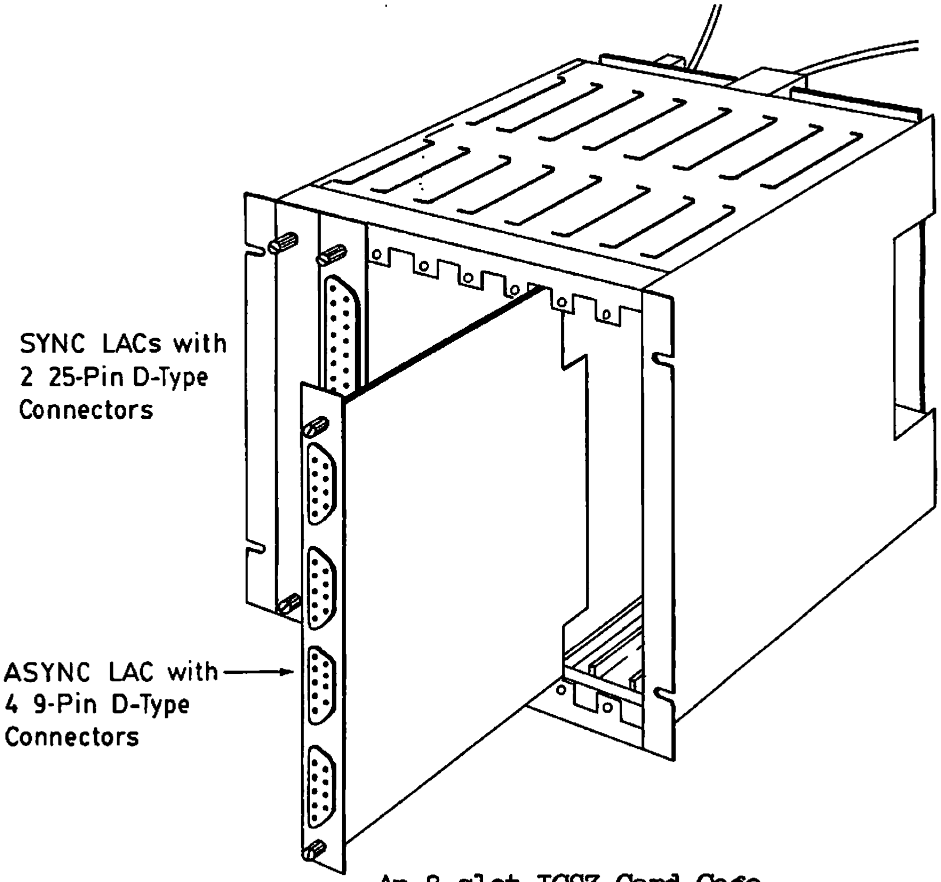


Asynchronous Lines Connect Terminals and Peripherals to Your System
Figure 11-1

Like AMLC controllers, ICS1 controllers are located in the CPU cabinet. Each ICS1 controller supports 1 synchronous and a maximum of 8 asynchronous lines.

ICS2 and ICS3 controllers have two basic parts: the controller board, which is located in the mainframe cabinet, and an external card cage, which is located in a peripheral cabinet. Card cages come in 8-slot models for 2350s and 2450s, and 16-slot models for all other systems. Each slot can hold one 4-line asynchronous Line Adapter Card (LAC) or one 2-line synchronous LAC to add more lines as your need for lines increases. Thus, an 8-slot cage can support a maximum of 32 asynchronous lines, and a 16-slot cage can support a maximum of 64 asynchronous lines. You can connect any combination of 8-slot and 16-slot card cages to your system provided that the total number of lines conected to your system does not exceed 256.

LACs are usually distributed amoung card cages. You do not have to load them into adjacent slots. An empty slot does not affect operation. ICS controllers recognize the presence of a LAC and allocate logical line numbers based on the order in which the cards appear. Logical line numbers are allocated first to lines attached to AMLC controllers, then to lines attached to ICS controllers based on the controller's address. Figure 11-2 shows an 8-slot ICS3 LAC card cage loaded with a 2-line synchronous LAC and a 4-line asynchronous LAC.



An 8-slot ICS3 Card Cage
Figure 11-2

Controller Characteristics

Table 11-1 summarizes the characteristics of AMLC and ICS controllers.

Table 11-1
Characteristics of Asynchronous Controllers

Controller	Lines	Comments
AMLC 5152	8 async	No longer marketed, but still supported by Prime
AMLC 5154	16 async	RS-232-C/CCITT V.24
AMLC 5174	16 async	20 milliamp current loop
AMLC 5175	16 async	Lines 0-7 20 milliamp Lines 8-15 RS-232-C/V.24
ICS1 5181	8 async 1 sync	RS-232-C/CCITT V.24 Synchronous line supports bisync RJE (HASP/2780,3780) and HDLC PRIMENET
ICS2 5242	64 async maximum	RS-232-C/CCITT V.24 Basic model supports 16 slots for 2-line sync or 4-line async line adapter cards
ICS2 5720	32 async maximum	RS-232-C/CCITT V.24 Basic model provides async and sync SDLC SNA support V.35/DDS
ICS3 5725 ICS3 5730 ICS3 5755	64 async maximum	RS-232-C/CCITT v.24 Basic model supports 8 slots for 2-line sync or 4-line async line adapter cards for 2350s and 2450s, 16-slot models for all other systems

MONITORING YOUR CONTROLLERS

The STATUS COMM command provides an overview of your communication lines and controllers. Use the STATUS COMM command to display the line count, device name, and device address (in octal) for all the controllers connected to your system.

If any of these devices are identified by type (AMLC), downline load file number (ICS), or a PROM set ID number (MDLC), this information is displayed. Otherwise, the Type field is left blank.

Each controller's total line count is calculated separately for synchronous and asynchronous lines. Line counts for inoperable lines are calculated and displayed in the same manner. An entry in the Bad-Lines field indicates that a line has failed. This is a minor hardware problem and other lines attached to the same LAC continue to operate.

There are five controllers in the example of the STATUS COMM command shown below. One is an ICS2 controller, using downline load file ICS2_01.DL, at device address '10. It currently supports two synchronous lines and 60 asynchronous lines, one of which is inoperable.

```
OK, STAT COMM

                        Device    Total-Lines     Bad-Lines
Controller   Type       Address   Async   Sync    Async    Sync

 ICS2        F-01         10       60      2      1        0
 ICS1                     37        8      1      0        0
 MDLC        5646         50        0      4      No Information
 AMLC        DMQ          52       16      0      No Information
 AMLC        DMQ          53       16      0      No Information
 AMLC        DMQ          54       16      0      No Information
```

Line Number Allocation

Line numbers are allocated first to AMLC controllers and then to ICS controllers. Line number assignment to AMLC controllers is based on the octal device address, as shown in the following list. This assignment is fixed and cannot be changed.

AMLC Controller	Octal Device Address	Octal Line Numbers	Decimal Line Numbers
1	'54	'0 to '17	0 to 15
2	'53	'20 to '37	16 to 31
3	'53	'40 to '57	32 to 47
4	'35	'60 to '77	48 to 63
5	'32	'100 to '117	64 to 79
6	'17	'120 to '137	80 to 95
7	'16	'140 to '157	96 to 113
8	'15	'160 to '177	112 to 127

Lines attached to ICS controllers are assigned line numbers based on device address in the following order: '10, '11, '36, and '37.

ICS2 and ICS3 controllers are given a device address of '10 or '11 by the manufacturer. ICS1 controllers are usually assigned a device address of '36 or '37. However, if you have only ICS1 controllers on your system, all ICS device addresses ('10 , '11, '36, and '37) can be used.

In the following example there are three controllers. Blocks of line numbers have been allocated to each controller.

Controller	Octal Device Address	Line Count	Octal Line Numbers	Decimal Line Numbers
AMLC	'54	16 Lines	'0 to '17	0 to 15
ICS3	'10	32 Lines	'20 to '57	16 to 47
ICS1	'36	8 Lines	'60 to '67	48 to 55

The first line number on each ICS controller is always a multiple of eight. Line numbers are allocated for all the lines connected to an ICS controller and then skip to the next multiple of eight for the next ICS controller. This can result in line numbers being allocated for lines that are not currently in use. In the example below, gaps occur from line numbers 44 to 47 and from 108 to 111. The block of line numbers from 0 to 15 is reserved for an AMLC controller at device address '54.

Controller	Octal Device Address	Line Count	Octal Line Numbers	Decimal Line Numbers
AMLC	'53	16 Lines	'20 to '37	16 to 31
ICS3	'10	12 Lines	'40 to '53	32 to 43
ICS2	'11	60 Lines	'60 to '153	48 to 107
ICS1	'36	8 Lines	'160 to '167	112 to 119

Maintaining ICS2 Integrity

ICS2 and ICS3 controllers contain a maximum of 16 line adapter cards, each supporting four lines. Use the optional ICS CARDS directive to have PRIMOS report, at cold start, any inconsistencies between the expected LAC configuration and the actual configuration. (The ICS CARDS directive is explained in Chapter 10 of this guide, CONFIGURATION DIRECTIVES, and in the ICS User's Guide.)

If, at any other time, you suspect that one or more LACs have failed, try the following procedure.

1. Warm start the system.

2. If the warm start does not identify or resolve the problem, use the STATUS COMM command and record the results, making note of the number of lines attached to each ICS2 or ICS3 controller.

3. Cold start the system. If the ICS2 or ICS3 subsystem can identify a bad LAC, one of the following messages is displayed at the supervisor terminal:

   ICS device dd: async line ee (Jn) on line card in slot y is inoperable

   ICS device dd: line card in slot y is inoperable.

   ICS device dd: line card in slot y is unrecognizable.

   dd identifies the device address, usually 10 or 11
   ee identifies the physical line number
   n identifies one of four jacks on the LAC
   y identifies the affected slot on the controller

4. If no message appears but the problem has not been resolved, use the STATUS COMM command again. Compare its output with that in Step 2. If the output differs, a problem with at least one LAC is causing configuration discrepancies. Contact your Customer Service Representative for help.

Changing Line Cards on ICS2 or ICS3 Controllers

If it becomes necessary for your Customer Service Representative to change the line adapter cards on an ICS2 or ICS3 (for example, to replace or remove a card or to change the total number of cards on the controller), shut down PRIMOS and power off the system before doing so. After the changes are complete, cold start the system.

> **Caution**
>
> Do not connect or disconnect LACs to or from LAC card cages, or LAC cables to or from LACs, when the power is on. Damage may result.

CONFIGURING ASYNCHRONOUS LINES

To configure asynchronous lines, use the following procedure:

1. Calculate the octal values for the following directives:

Terminal Users	(NTUSR)
Assigned Lines	(NAMLC)

   For instructions on calculating these values, see Chapter 10, CONFIGURATION DIRECTIVES.

2. Use EMACS or ED to include these values in your PRIMOS configuration file.

3. Use SET_ASYNC to create the commands that define assignability, protocols, baud rates, and other parameters for your asynchronous lines. The total number of terminal lines you configure should be one less than the value of NTUSR. (The system console is included in NTUSR.) The directives you generate are stored in your PRIMOS configuration file and will take effect the next time you cold start your system.

4. Optionally, use the AMLBUF directive to change buffer sizes to fine tune the system and increase performance. The AMLBUF directive is described later in this chapter. For more information see Chapter 10, CONFIGURATION DIRECTIVES.

Configuration Directives

The following configuration directives establish asynchronous communication at cold start. They are stored in the system configuration file.

Directive	Meaning	Controller
NTUSR n	Sets number of local terminals (including supervisor terminal)	ALL
NAMLC n	Specifies number of assigned lines	ALL
AMLIBL	Sets size of DMC input tumble tables	AMLC
AMLBUF	Sets size of input, output, and DMQ buffers	ALL
REMBUF	Sets size of input and output buffers for remote users	ALL
AMLCLK	Sets the software programmable clock	ALL
ASYNC JUMPER	Sets asynchronous line speeds	ALL
ICS INPQSZ	Sets size of input queue	ICS
ICS CARDS	Specifies LAC (Line Adapter Card) configuration	ICS2 and ICS3
ICS INTRPT	Sets value for I/O clock interrupt rate	ICS
AMLTIM	Sets time intervals for data set signal management.	ALL
DTRDRP	Drops the DTR signal when users log out	ALL
DISLOG	Automatic logout for disconnected users	ALL

These are communications directives. Other directives are needed to operate your system. Refer to Chapter 10, CONFIGURATION DIRECTIVES, for a complete listing and further instructions.

Determining Line Numbers

A maximum of 256 asynchronous communication lines can be attached to your system, including one line for the supervisor terminal. These lines are attached to your AMLC and ICS controllers and are identified by number. Line numbers begin at zero and cannot exceed 255. The decimal line number required to configure an asynchronous line with SET_ASYNC can be determined by using the STATUS USERS command, as shown below. This command displays the user ID, the user number in decimal, the line number in octal and decimal, and the devices currently in use.

```
    OK, STATUS USERS

                        Line
User             No   oct(  dec) Devices
SYSTEM            1      asr    <OZSYS1> AL77
DOROTHY           2    0(    0) <MUNCH8> <KANSAS> <OZSYS1>
TOTO              6    4(    4) <MUNCH8> <KANSAS> <OZSYS1>
RAINBOW           7    5(    5) <KANSAS> <MUNCH8> <OZSYS1>
TINMAN           11   11(    9) <OZSYS1>
SCARECROW        14   14(   12) <OZSYS1> <KANSAS>
WIZARD           17   17(   15) <OZSYS1>
Y.B.ROAD         22   24(   20) <MUNCH8> <EMCITY>
AUNTI.EM         24   26(   22) <KANSAS> (to SHELTER)
UNCLE.H          25   27(   23) <KANSAS> (to SHELTER)
LION             30   34(   28) <FOREST> <OZSYS1> <CASTLE>
WIZARD           38   44(   36) <OZSYS1> <EMCITY>
W.WITCH          40   46(   38) <BROOM1> <OZSYS1> <CASTLE>
MONKEY           41    slave    <CASTLE>
BALLOON          43    slave    <EMCITY>
G.WITCH          65      rem    <MUNCH8> (from NORTH )
WIZARD           88    phant    <EMCITY>
YTSMAN           93    phant    <OZSYS1>
BACKUP_SERVICE   96    phant    <OZSYS1>
LOGIN_SERVER     97     LSr     <OZSYS1> (3)
NETMAN           98     nsp     <OZSYS1>
TAPE_PHANTOM    100    phant    <OZSYS1>
LQ.PRINTER      101    phant    <OZSYS1> PRO
BATCH_SERVICE   103    phant    <OZSYS1> (2)
PRINT_SERVER    111    phant    <OZSYS1>
RAINBOW         117    child    <KANSAS> <MUNCH8> <OZSYS1>
```

In the above example, W.WITCH, user number 40, is attached to physical line number 38. G.WITCH, user number 65, is a remote user and is assigned a virtual line from a pool reserved for remote users with the NRUSR directive. This is indicated by the abbreviation "rem" in the line number field. WIZARD, user numbers 17, 38, and 88, is logged in at two devices. Although this user name appears three times in this display, WIZARD is attached to only two physical line numbers (lines 15 and 36). WIZARD, user number 88, is a phantom process. It is not associated with a terminal and does not require a line.

You can generally determine the physical line number from the user number that is displayed after login or logout by using the following formula:

```
    Physical Line Number= User Number - 2
```

The following examples show how to apply this formula.

```
    Login please.
    LOGIN TINMAN
    Password?

    TINMAN (User 11) Logged in Monday, 28 Apr 86 19:01:20.
    Welcome to PRIMOS version 20.2
    Copyright (c) Prime Computer, Inc. 1985.
    Last Login Friday, 25 Apr 86 10:54:40.
```

In this example, TINMAN is User 11 on line number 9 (11 - 2 = 9). Zero is the first valid line number on all Prime systems. In the following example DOROTHY is User 2 on line number 0 (2 - 2 = 0).

```
    OK, LO

    DOROTHY (User 2) Logged out Tuesday, 15 Apr 86 17:59:12.
    Time used:  08h 41m connect, 04m 20s CPU, 01m 06s I/O.
```

The supervisor terminal is connected to the virtual control panel on the CPU and uses a separate line that is configured with the ASRATE directive. This is indicated by the abbreviation "asr" in the line number field. For more information on the ASRATE directive, see Chapter 10, CONFIGURATION DIRECTIVES.

These methods apply when the default line number assignments are used. If you cannot use either of these two methods to determine a line number, use the procedure shown in Appendix E, DETERMINING PHYSICAL LINE NUMBERS.

AMLC I/O Interrupt Rate

Each AMLC controller reads incoming characters from all the lines attached to it and holds them a two-part storage area called a tumble table. When the current tumble table buffer is full, the AMLC controller switches the incoming stream of characters to the other tumble table buffer and interrupts the AMLDIM process to execute. When the AMLDIM process executes, it reads each character in the tumble table, identifies the line that it came from, and transfers it to the line's input ring buffer. If the line is full-duplex, the characters are also sent to the line's output ring buffer. The AMLC I/O interrupt rate determines how frequently the AMLC controllers interrupt the AMLDIM process. This is also known as the character-time interrupt (CTI) rate.

The AMLC I/O interrupt rate is calculated by dividing the baud rate of the system's highest-numbered AMLC line (that is, the last line of the last AMLC board) by the character size (11 bits for 110 baud, 10 bits for other speeds). Increasing the interrupt rate greatly increases the amount of CPU time spent servicing the AMLC controllers.

> Caution
>
> The default AMLC I/O interrupt rate is 10 interrupts per second. This requires that the last AMLC line be configured for 110 baud. Increasing this line's baud rate beyond 300 baud (30 interrupts per second) results in a severe performance degradation. For this reason, do not assign this line or connect it to a high speed device. Reserve this line for a very slow terminal or a printer. If your hardware resources permit, configure this line at 110 baud but do not connect it.

On a system with both AMLC and ICS controllers, the baud rate of the last physical AMLC line controls the rate for processing AMLC interrupts. The ICS INTRPT directive controls the rate for ICS interrupts.

ICS I/O Interrupt Rate

The I/O interrupt rate for ICS controllers is set by the ICS INTRPT directive. It is independent of the speed or use of any ICS lines. The ICS I/O interrupt rate, like the AMLC I/O interrupt rate, operates at the character level.

Characters are transferred from the ICS controller to the DMQ input queue via the PRIME I/O bus. The ASYNDM process is interrupted by the ICS controllers and transfers characters to the appropriate input ring buffer. The frequency of the interrupts is governed by the ICS INTRPT directive, which specifies the number of interrupts per second.

The ICS INTRPT rate must be set fast enough to clear the incoming data without interrupting the ASYNDM process too frequently. The default and minimum value is 10 (octal '12) interrupts per second. The maximum is 100 (octal '144) interrupts per second. If the value selected is out of range (too high or too low), the ICS controller defaults to the maximum or minimum value as appropriate.

The procedure for calculating the ICS interrupt rate is explained in Chapter 10, CONFIGURATION DIRECTIVES, and in the ICS User's Guide.

THE SET_ASYNC COMMAND

The SET_ASYNC command sets terminal line characteristics for asynchronous lines connected to AMLC and ICS controllers. Usually, you will store the SET_ASYNC commands in your PRIMOS startup command input file, so that all asynchronous lines will be configured on your system at cold start. However, you can also use the SET_ASYNC command while the system is running to alter the characteristics of a particular line. SET_ASYNC can be used only at the supervisor terminal.

SET_ASYNC is a functional replacement, with extensions, for the AMLC command. It accepts data in the form of command line options that are straightforward and easy to understand. The SET_ASYNC command performs the following operations:

- Defines the protocol
- Enables Auto Speed Detect (ASD)
- Sets line speed and parity
- Defines the initial terminal line characteristics
- Associates a user number with a line number or defines the line as assignable
- Provides the default values shown in Table 11-2 when the -DEFAULT option is specified
- Associates a set of identical line characteristics with a range of consecutively numbered lines

Users can customize their own terminals by issuing the TERM command at the keyboard or in a LOGIN.CPL or LOGIN.COMI file. The TERM command allows the user to specify full-duplex or half-duplex mode and disable the break character (CONTROL-P). For a detailed description of the TERM command, see the PRIMOS Commands Reference Guide.

Note

SET_ASYNC accepts decimal-based input only. Octal-based input is not supported.

The format of the SET_ASYNC command is as follows:

```
SET_ASYNC { -LINE n [-TO m] [options] }
          { -HELP                     }
```

Table 11-2

Default Settings for Asynchronous Lines
Provided by the SET_ASYNC -DEFAULT Option

Option	Setting
-ASSIGNABLE NO	Line is a login line.
-CHAR_LENGTH 8	Character length is 8 bits.
-DATA_SET_CONTROL	Enables modems and port selectors to recognize when information is being transmitted.
-ECHO	Full-duplex line. (PRIMOS echoes characters.)
-NO_DATA_SENSE_ENABLE	Data Set Sense or reverse channel protocol is disabled.
-NO_ERROR_DETECTION	NAK character is not placed in the input buffer when an input parity or input buffer overflow is sensed.
-NO_LINE_FEED	Do not echo LINE FEED for RETURN.
-NO_LOOP_LINE	Line is not in loopback mode.
-NO_REVERSE_XOFF	Reverse Flow Control is not enabled.
-PARITY NONE	Line parity is neither generated nor checked.
-PROTOCOL TRAN	Line uses the transparent protocol, TRAN.
-SPEED 1200	Line speed is 1200 bits per second.
-STOP_BITS 1	One stop bit.
-USER_NUMBER (n + 2)	Associates the buffers for USER_NUMBER (n + 2) with the physical line number n specified in the -LINE n option. For example, the default buffer associated with -LINE 5 is buffer 7.
-XOFF	CONTROL-S stops and CONTROL-Q resumes the flow of data from the system to the terminal.

The options to the SET_ASYNC command are as follows:

Option — Description

-HELP

Displays the format of the command and a complete list of available options. When you select -HELP, all other options on the command line are ignored.

-LINE n

Where n is the required DECIMAL line number to be configured or, when used with the -TO option, the first line number in a series to be configured with identical options.

-TO m

Where m specifies the last line number in a series beginning at the line number given in -LINE. The -TO option allows the System Administrator to configure a range of consecutively numbered lines with identical options. This value must be greater than the line number specified by the -LINE option.

-DEFAULT

Sets all options to their default setting, shown in Table 11-2, with the exception of any options further specified in the command line.

-ASSIGNABLE YES
-ASGN NO

Indicates whether the line is an assignable line. Assigned lines are required for serial devices. NO is the default value. The user number for assigned lines must be zero. The use of an async line can be changed after system startup from assignable to login and vice versa with the SET_ASYNC -ASGN command. However, unless you edit your PRIMOS.COMI file to include the command line, the assignment will apply only until the next system shutdown. The -ASGN NO option is required to convert a previously assigned line to a regular login line. For more information, see the section Assignable Asynchronous Lines later in this chapter.

-USER_NUMBER n
-USER

Where n is the decimal user number. This option associates a buffer with a physical line. The default value is the line number specified in the -LINE option plus two. The user number for assigned lines must be zero. Do not use this option with the -TO option unless you set a range of assigned lines to user number zero. For example,

```
SET_ASYNC -LINE 5 -TO 9 -ASGN YES -USER 0
```

Option — Description

-PROTOCOL name

Where name may be any of the following:

TTY (System default)
TRAN (-DEF option default)
TTYUPC
TTYNOP
TT8BIT
ASD

For more information, see the sections Data Communication Protocols and Enabling Auto Speed Detect later in this chapter.

The following obsolete protocols are also supported:

TTYHS
TRANHS
TTYHUP

-SPEED value

value may be any of the following baud rates:

50
75
110
134.5
150
200
300
600
1200 (-DEF option default)
1800
2400
3600
4800
7200
9600
19200
CLOCK (Speed set with AMLCLK directive)
J1 (Jumper speeds J1, J2, and J3 are
J2 set with ASYNC JUMPER directive.)
J3

For further information on setting the ASYNC jumper speeds and the AMLCLK, see Chapter 10 of this book, CONFIGURATION DIRECTIVES.

Option	Description
-STOP_BITS n -SB	Where n defines the number of stop bits to use per character either 1 (default) or 2. Stop bits signal the receiving device to wait for the next character. All characters have 1 start bit, 1 parity bit, 7 information bits, and either 1 (default) or 2 stop bits. PRIMOS recognizes both 10-bit and 11-bit character lengths. Two stop bits are used for devices that operate at 110 baud.
-PARITY NONE ODD EVEN	Either sets the line parity to the desired setting or disables parity. The default is NONE.
-CHAR_LENGTH n -CL	Where n can be 5, 6, 7, or 8. 8 is the default. Sets the number of information and parity bits per character. Character length can be adjusted for nonstandard character sizes required by Arabic terminals, telex lines, or foreign devices. PRIMOS right-justifies the bits in a byte and sets the leftmost bits to zero. The example below uses one stop bit.

```
-CL 5   STOP 000XXXXX START
-CL 6   STOP 00XXXXXX START
-CL 7   STOP 0XXXXXXX START
-CL 8   STOP XXXXXXXX START
```

Option	Description
-ECHO	Sets the line to be echoed by the host (full-duplex). This is the default.
-NO_ECHO	Sets the line so characters do not echo on the screen (half-duplex). -ECHO is the default.
-LINE_FEED -LF	Echos a line feed character for the RETURN key. This option is valid only when -NO_ECHO (half-duplex) is specified. -NO_LINE_FEED is the default.
-NO_LINE_FEED -NOLF	Does not echo a line feed character for RETURN. This is the default.

Option — Description

-XOFF

The line will recognize CONTROL-S and CONTROL-Q (-XON and -XOFF) to stop and start the flow of data on the line from the host to the terminal. This is the SET_ASYNC default.

-NO_XOFF

Disables CONTROL-S and CONTROL-Q. This option is used for devices that cannot recognize these control key sequences. -XOFF is the default.

-REVERSE_XOFF
-REVXOFF

Enables Reverse Flow Control (RFC) for asynchronous lines. -NO_REVERSE_XOFF is the default. RFC sends XOFF characters to a device when the PRIMOS input ring buffer is 60% full. When the input ring buffer drops to 20% full, an XON character is sent to the device to indicate that transmission can resume. Choose this option only for lines connected to devices that can interpret XON and XOFF characters (such as PT45, PST 100, and PT200). RFC also attempts to prevent DMQ input queue overruns for ICS3 controllers and ICS2 controllers that are not using BSC and ASYNC protocols.

-NO_REVERSE_XOFF
-NOREVXOFF

Disables Reverse Flow Control for the line. This is the default.

-DATA_SENSE_ENABLE
-DSE

Enables the DATA_SET_SENSE protocol, also known as reverse channel protocol. This option is used for transmitting control information and for devices that do not recognize XON/XOFF. NO_DATA_SENSE_ENABLE is the default.

-NO_DATA_SENSE_ENABLE
-NODSE

Disables the DSS protocol (reverse channel). This is the default.

Option	Description
-DATA_SET_SENSE { HIGH -DSS { LOW	Supports devices that toggle an RS-232-C pin (usually pin 8) to indicate when they are busy/ready instead of using XON/XOFF. The DSS protocol sets the ready value as either high (pin signal raised) or low (pin signal lowered). The default is HIGH. You must use the -DSS option with -DATA_SENSE_ENABLE. Some devices use pins other than pin 8. If this is the case, ask your Customer Service Representative to arrange your cables so that the data set sense signal is wired into the pin used for carrier detect. Data set sense is also referred to as buffered protocol or reverse channel protocol.
-DATA_SET_CONTROL -DSC	Required for modems and port selectors to recognize when a block of information is transmitted. -DSC is ignored by terminals. This is the default.
-NO_DATA_SET_CONTROL -NODSC	Disables the -DSC option.
-LOOP_LINE -LOOP	This is used only for testing purposes to verify the accuracy of data transmitted by returning it back to the receiving line. This is a software-enabled hardware loopback.
-NO_LOOP_LINE -NO_LOOP	Disables the -LOOP_LINE option. This is the default.
-ERROR_DETECTION -ERRDET	This is used only for testing. When an input buffer overflows or when a parity error is detected, the incoming character is replaced with an ASCII NAK. -NO_ERROR_DETECTION is the default.
-NO_ERROR_DETECTION -NOERRDET	Prevents the line from sending an ASCII NAK character if an input parity or input buffer overflow error is sensed. This is the default.

Here are some examples of the use of the SET_ASYNC command.

The following SET_ASYNC command line enables reverse flow control for a group of 10 PT200 terminals attached to consecutive lines on an ICS3 controller.

```
    SET_ASYNC -LINE 129 -TO 138 -DEF -PRO TTY -SPEED 9600 -REVXOFF
```

To run diagnostics on a line without changing any existing parameters, issue the following SET_ASYNC command from the supervisor terminal.

```
    SET_ASYNC -LINE 71 -LOOP -ERRDET
```

Use the following command line for a 110-baud modem attached to the last line of the eighth AMLC controller.

```
    SET_ASYNC -LINE 127 -DEF -SPEED 110 -SB 2 -DSC
```

This command line also sets the AMLC I/O interrupt rate at 10 interrupts per second.

The following SET_ASYNC command line creates a group of six assignable lines. In this case the corresponding value of NAMLC is six.

```
    SET_ASYNC -LINE 194 -TO 199 -ASGN YES -PRO TTY
```

Use the following command line for a 9600-baud Arabic terminal.

```
    SET_ASYNC -LINE 87 -PRO TT8BIT -PAR ODD -SPEED 9600 -SB 2
```

Use the following SET_ASYNC command line for 110-baud hard copy terminal that cannot print lowercase characters.

```
    SET_ASYNC -LINE 17 -PRO TTYUPC -SPEED 110 -CL 7 -SB 2
```

To find out the SET_ASYNC equivalent of a given AMLC command, you can use the interactive form of the CONVERT_AMLC_COMMANDS utility described later in this chapter.

SET_ASYNC Error Messages

Most errors are generated by SET_ASYNC when you enter contradictory, ambiguous, or incorrect values in the command line options. If any error occurs that is termed FATAL in the error messages below, then the command is aborted and no action is taken. If any error occurs that is termed WARNING in the error messages below, the command is processed.

All error messages display the line number that you specified, if possible. This is especially helpful when you specify multiple line numbers. Note that no error messages are displayed when you select the -HELP option.

- Incompatible USER_NUMBER and ASSIGNABLE options specified (line n).

This message is displayed when a user number of zero is associated with a non-assignable line or a non-zero user number is associated with an assignable line. FATAL

- Invalid <argument> value x (line n).

This message is displayed when an invalid, unrecognized, or missing argument value is specified for argument on line n. FATAL.

- Invalid line number(s) specified.

This message is displayed when the line number or -TO number specified is out of range or otherwise invalid. FATAL.

- Invalid protocol value x (line n).

This message is displayed when the argument entered is not one of the expected values. See the -PROTOCOL option described previously for a list of valid protocols. FATAL.

- Invalid speed value x (line n).

This message is displayed when the argument entered is not one of the expected values. See the -SPEED option described previously for a list of valid speeds. FATAL.

- Line number MUST be specified.

This message is displayed when the line number to be configured is omitted. FATAL.

- Line number specified with the -TO option must be greater than the line number specified with the -LINE option.

This message is displayed when the -TO line number is less than or equal to the -LINE argument. FATAL.

- -<option> and -NO_<option> cannot both be specified at the same time (line n)

This message is displayed when an option and its converse are specified at the same time. For example, specifying both -XOFF and -NO_XOFF for the same line produces this error message. FATAL.

- Speed specified x can not be set (line n).

This message is displayed when the speed value selected is valid but is not available in the current physical terminal configuration. FATAL.

- System User command only.

This message is displayed if the SET_ASYNC command is issued from anywhere but the supervisor terminal. Although any user can issue the command SET_ASYNC -HELP, only the person at the supervisor terminal, usually the System Administrator, is permitted to configure lines. FATAL.

- Warning: Line_Feed option not meaningful unless -NO_ECHO is specified (line n)

This message is intended to notify the operator that -LINE_FEED was specified, which is only meaningful if the line is half duplex. This combination is ignored.

CONVERT_AMLC_COMMANDS

CONVERT_AMLC_COMMANDS is a utility that translates AMLC commands to their equivalent SET_ASYNC commands. By using this utility, you can easily translate the AMLC commands in your existing PRIMOS startup command input file without altering the other commands or corrupting your original file. Interactive mode is particularly useful when you are learning how to use SET_ASYNC.

CONVERT_AMLC_COMMANDS is not an installed PRIMOS command. It is supplied in the UFD TOOLS as CONVERT_AMLC_COMMANDS.RUN. You can copy this file into the UFD CMDNC0 and invoke it as a regular PRIMOS command. Use the following format to invoke this utility:

```
RESUME TOOLS>CONVERT_AMLC_COMMANDS  { input-file output-file }
                                    { -INTERACTIVE           }
                                    { -HELP                  }
```

Option — Description

input-file output-file
Translates AMLC commands contained in the input file to their SET_ASYNC equivalent, and stores them in the output file. The input file can be your current PRIMOS.COMI, or any other file that contains AMLC commands. The output file is created by this utility. Do not choose the input file or any other existing file as your output file.

-INTERACTIVE
Queries you for an AMLC command, translates the command, and displays the SET_ASYNC equivalent on the screen. Type QUIT to exit from interactive mode and return to PRIMOS.

-HELP
Shows the syntax of the command line.

The following message is displayed when the -HELP option is used.

```
OK, RESUME TOOLS>CONVERT_AMLC_COMMANDS -HELP

[Convert_Amlc_Commands  Rev 20.2 (c) Prime Computer 1985]
USAGE:
CONVERT_AMLC_COMMANDS  { inputpathname outputpathname | -Help |
 -INTERactive }
```

The following example demonstrates interactive mode:

```
OK, RESUME TOOLS>CONVERT_AMLC_COMMANDS -INTER
[Convert_Amlc_Commands  Rev 20.2 (c) Prime Computer 1985]

>AMLC 0 TTY 2413 020002

SET_ASYNC -LINE 0 -PRO TTY -DEFAULT -SPEED CLOCK -LF -BUFFER 2

>AMLC 1 TTY 2413 020003

SET_ASYNC -LINE 1 -PRO TTY -DEFAULT -SPEED CLOCK -LF -BUFFER 3

>AMLC 4 TTY 2413 020006

SET_ASYNC -LINE 4 -PRO TTY -DEFAULT -SPEED CLOCK -LF -BUFFER 6

>AMLC 5 TTY 2413 020007

SET_ASYNC -LINE 5 -PRO TTY -DEFAULT -SPEED CLOCK -LF -BUFFER 7

>AMLC 6 TTY 2413 020010

SET_ASYNC -LINE 6 -PRO TTY -DEFAULT -SPEED CLOCK -LF -BUFFER 8

>AMLC 7 TTY 2213 020020

SET_ASYNC -LINE 7 -PRO TTY -DEFAULT -SPEED 300 -LF -BUFFER 16

>AMLC 10 TRAN 2313 020000

SET_ASYNC -LINE 8 -PRO TRAN -DEFAULT -LF

>AMLC 11 TRAN 2313

SET_ASYNC -LINE 9 -PRO TRAN -DSC -NO_LOOP -SPEED 1200 -NO_REV_XOFF
-SB 1 -PAR NONE -CL 8

>QUIT

OK,
```

Note

CONVERT_AMLC_COMMANDS generates a separate SET_ASYNC command for each AMLC command it processes. This utility cannot recognize when consecutive lines have identical directives and cannot use the SET_ASYNC -TO option to translate a logical group of directives into a single directive.

In the following example the CONVERT_AMLC_COMMANDS utility is given an input file called PRIMOS.COMI. An abbreviated listing of this file is provided on the following page.

```
CONFIG -DATA CONFIG                   /* Name of config file for start-up
COMO PRIMOS.COMO                      /* Open como file to record this startup
ADDISK 3660 14061 3462 71261          /* Add partitions to the system
COMO -NTTY                            /* Turn off echo to system console
AMLC  TTY      0  2413  20002         /* User terminal set to 9600 baud
AMLC  TTY      1  2413  20003         /* User terminal set to 9600 baud
AMLC  TTY      2  2413  20004         /* User terminal set to 9600 baud
AMLC  TTY      3  2413  20005         /* User terminal set to 9600 baud
AMLC  TTY      4  2413  20006         /* Set TTY Protocol and 9600 baud
AMLC  ASD      4                      /* Enable ASD, Retain TTY and baud
AMLC  TTY      5  2313  20007         /* Dial-in line set to 1200 baud
AMLC  TTY      6  2413  20010         /* ONTEL line set at 9600 baud
AMLC  TTY      7  2413  20011         /* OAS terminal set to 9600 baud
AMLC  TTY     10  2413  20012         /* Forms terminal set to 9600 baud
AMLC  TTY     11  2413  20013         /* Unused line
AMLC  TTY     12  2413  20014         /* Unused line
AMLC  TTY     13  2413  20015         /* Unused line
AMLC  TTYNOP  14  2413  20000         /* Laser Printer set at 9600 baud
AMLC  TRAN    15  2413  20000         /* Assigned Line for 9600 baud Printer
AMLC  TTYNOP  16  2313  20000         /* OASTAT Printer set at 1200 baud
AMLC  TTYNOP  17  2213  20000         /* Letter Quality Printer set at 300 baud
OPR 1                                 /* Share requires OPR 1 (opens memory)
SHARE SYSTEM>ED2000 2000              /* Share the Editor
SHARE SYSTEM>S2050 2050 700
R SYSTEM>S4000 1/1
SHARE 2050
R SYSTEM>SP4000 1/10
SHARE SYSTEM>S$2167 2167              /* Share Spool libraries
R SYSTEM>S$4000 1/12
OPR 0                                 /* Close memory
PROP PRO -START                       /* Start printer PRO
/*
CO SYSTEM>BASICV.SHARE.COMI 7             /* Share BASICV
CO SYSTEM>DBG.SHARE.COMI 7                /* Share DBG
CO SYSTEM>EMACS.SHARE.COMI 7              /* Share EMACS
CO SYSTEM>FORMS.SHARE.COMI 7              /* Share FORMS
CO SYSTEM>MIDASPLUS.SHARE.COMI 7          /* Share MIDASPLUS
CO SYSTEM>OAS.SHARE.COMI 7                /* Share OAS
CO SYSTEM>INFORMATION.SHARE.COMI 7        /* Share Prime INFORMATION
CLOSE 7                                   /* Close file unit 7
COMO -END -TTY                            /* Close como file, turn on echo
CO -END                                   /* End of Primos.comi startup
```

The source filename, PRIMOS.COMI, and the target filename, CONVERT.COMI, are supplied in the command line, as shown below.

```
OK, RESUME TOOLS>CONVERT_AMLC_COMMANDS PRIMOS.COMI CONVERT.COMI
[Convert_Amlc_Commands  Rev 20.2 (c) Prime Computer 1985]
OK,
```

CONVERT_AMLC_COMMANDS processes the input file, creates the output file in your current directory, and returns to PRIMOS. When you list the contents of your directory with an LD command, the output filename appears. Here is an abbreviated listing of the target file CONVERT.COMI. Notice that CONVERT_AMLC_COMMANDS has translated only AMLC commands. All the other commands are the same as before.

```
CONFIG -DATA CONFIG               /* Name of config file for start-up
COMO PRIMOS.COMO                  /* Open como file to record this startup
ADDISK 3660 14061 3462 71261      /* Add partitions to the system
COMO -NTTY                        /* Turn off echo to system console
SET_ASYNC -LINE 0  -PRO TTY    -DEFAULT -SPEED CLOCK -LF -USER 2  /* User terminal set to 9600 baud
SET_ASYNC -LINE 1  -PRO TTY    -DEFAULT -SPEED CLOCK -LF -USER 3  /* User terminal set to 9600 baud
SET_ASYNC -LINE 2  -PRO TTY    -DEFAULT -SPEED CLOCK -LF -USER 4  /* User terminal set to 9600 baud
SET_ASYNC -LINE 3  -PRO TTY    -DEFAULT -SPEED CLOCK -LF -USER 5  /* User terminal set to 9600 baud
SET_ASYNC -LINE 4  -PRO TTY    -DEFAULT -SPEED CLOCK -LF -USER 6  /* Set TTY Protocol and 9600 baud
SET_ASYNC -LINE 4  -PRO ASD                                       /* Enable ASD, Retain TTY and baud
SET_ASYNC -LINE 5  -PRO TTY    -DEFAULT -LF -USER 7               /* Dial-in line set to 1200 baud
SET_ASYNC -LINE 6  -PRO TTY    -DEFAULT -SPEED CLOCK -LF -USER 8  /* CNTEL line set at 9600 baud
SET_ASYNC -LINE 7  -PRO TTY    -DEFAULT -SPEED CLOCK -LF -USER 9  /* OAS terminal set to 9600 baud
SET_ASYNC -LINE 8  -PRO TTY    -DEFAULT -SPEED CLOCK -LF -USER 10  /* Forms terminal set to 9600 baud
SET_ASYNC -LINE 9  -PRO TTY    -DEFAULT -SPEED CLOCK -LF -USER 11  /* Unused line
SET_ASYNC -LINE 10 -PRO TTY    -DEFAULT -SPEED CLOCK -LF -USER 12  /* Unused line
SET_ASYNC -LINE 11 -PRO TTY    -DEFAULT -SPEED CLOCK -LF -USER 13  /* Unused line
SET_ASYNC -LINE 12 -PRO TTYNOP -DEFAULT -SPEED CLOCK -LF           /* Laser Printer set at 9600 baud
SET_ASYNC -LINE 13 -PRO TRAN   -DEFAULT -SPEED CLOCK -LF           /* Assigned Line for 9600 baud Printer
SET_ASYNC -LINE 14 -PRO TTYNOP -DEFAULT -LF                        /* OASTAT Printer set at 1200 baud
SET_ASYNC -LINE 15 -PRO TTYNOP -DEFAULT -SPEED 300 -LF             /* Letter Quality Printer set at 300 baud
OPR 1                             /* Share requires OPR 1 (opens memory)
SHARE SYSTEM>ED2000 2000          /* Share the Editor
SHARE SYSTEM>S2050 2050 700
R SYSTEM>S4000 1/1
SHARE 2050
R SYSTEM>SP4000 1/10
SHARE SYSTEM>S$2167 2167          /* Share Spool libraries
R SYSTEM>S$4000 1/12
OPR 0                             /* Close memory
PROP PRO -START                       /* Start printer PRO
/*
CO SYSTEM>BASICV.SHARE.COMI 7         /* Share BASICV
CO SYSTEM>DBG.SHARE.COMI 7            /* Share DBG
CO SYSTEM>EMACS.SHARE.COMI 7          /* Share EMACS
CO SYSTEM>FORMS.SHARE.COMI 7          /* Share FORMS
CO SYSTEM>MIDASPLUS.SHARE.COMI 7      /* Share MIDASPLUS
CO SYSTEM>OAS.SHARE.COMI 7            /* Share OAS
CO SYSTEM>INFORMATION.SHARE.COMI 7    /* Share Prime INFORMATION
CLOSE 7                               /* Close file unit 7
COMO -END -TTY                        /* Close como file, turn on echo
CO -END                               /* End of Primos.comi startup
```

OPTIONS FOR ASSIGNABLE LINES AND TERMINAL USERS

The following sections contain detailed information on assignable asynchronous lines, data communication protocols, and enabling Auto Speed Detect (ASD).

Assignable Asynchronous Lines

Assignable asynchronous lines are a class of asynchronous lines that are used for serial devices such as serial printers, plotters, card readers, punches, microcomputers, and certain hard-copy terminals when these terminals are used as printers. These serial devices include the following Prime products:

- 30/60 cps hard-copy terminal (Product Number 3115F)
- 300 lpm, matrix lineprinter/plotter (Product Number 3126F)
- 200 cps bidirectional matrix printer (Product Number 3350F)
- 200 cps bidirectional hard-copy terminal (Product Number 3351F)
- 55 cps bidirectional letter quality printer (Product Number 3185)

To assign an asynchronous line to a specific device use the following procedure.

1. Use the NAMLC directive to reserve buffers for the total number of assigned lines.
2. Use the AMLBUF directive to change the sizes of these buffers from their default settings (optional).
3. Use SET_ASYNC -LINE n -ASGN to indicate an assignable line.
4. Use the ASSIGN AMLC command to assign the line.

Note

> Printers on the Spooler subsystem are assigned lines when the PROP -START command is issued. PROP -START only assigns lines for printers. You must assign lines for any other devices.

For more information on the AMLBUF directive, see the section on allocating buffers, later in this chapter. For more information about assigning lines, see Appendix D of this book, the Operator's Guide to System Commands, and the PRIMOS Commands Reference Guide.

Data Communication Protocols

Data communication protocols allow the orderly transfer of data. These protocols usually define the format and relative timing of the information exchanged for a specific device. SET_ASYNC supports several asynchronous protocols that govern the actions taken by the PRIMOS Asynchronous Device Interface Modules (ASYNC DIMs). The acceptable values for the -PROTOCOL option are TRAN (the default), TTY, TT8BIT, TTYUPC, TTYNOP, and ASD. The basis for selection of the protocol is discussed next.

Protocols for older model AMLC boards (model 5054) are discussed in Appendix D of this guide, OBSOLETE AND RARELY USED COMMANDS AND DIRECTIVES.

TRAN: TRAN, the transparent protocol, is often used by lines connected to peripheral devices or to other computers. TRAN is chosen when it is not necessary to echo input, convert carriage returns to line feeds, or specifically acknowledge carriage returns and line feeds. The key sequence CONTROL-P has no special meaning under this protocol and is passed through to the program.

If no protocol is given in the SET_ASYNC command line, TRAN is assigned by the operating system.

TTY: TTY, the terminal protocol, is the protocol assigned at cold start to lines controlling interactive terminals. When the TTY protocol is chosen, all input from the terminal is echoed if the line is set for full duplex; a carriage return and a line feed is echoed following carriage return. The high order bit (the ASCII code parity bit) of each character input from the terminal is forced on. CONTROL-P and BREAK are interpreted as a QUIT command if the terminal is connected to the system as a user terminal.

If the line is an assigned line, CONTROL-P, BREAK, and line feed input from the terminal are ignored and discarded. A carriage return entered at the terminal is transmitted as a new line (or line feed) to the program requesting input. If the associated DMC tumble table or DMQ input queue becomes full, input is no longer echoed and any additional characters that are typed are lost.

Most terminals use this protocol and separate SET_ASYNC commands can be issued for each of your terminals. However, when terminals are attached to a series of consecutive lines, this protocol can be assigned to the entire group with only one command line as follows:

```
SET_ASYNC -LINE 14 -TO 35 -DEF -PRO TTY -SPEED 9600
```

TT8BIT: TT8BIT behaves in the same manner as the TTY protocol except that the high order bit (ASCII parity bit) is not forced on for each character entered at the terminal. All control characters are handled in the same manner as the TTY protocol.

This protocol may not be used for remote users and is recommended only for local Arabic DM5E/PLUS terminals.

To use this protocol, you must set the line parity to ODD, as shown in the example below:

```
    SET_ASYNC -LINE 5 -PRO TT8BIT -PAR ODD -SPEED 9600 -SB 2

    /* Line 5:  odd parity enabled, 9600 baud
```

TTYUPC: TTYUPC, the uppercase translating protocol, is used to avoid sending lowercase output to terminals or peripheral devices that cannot print lowercase characters. This is the only difference between TTY and TTYUPC protocols.

TTYNOP: TTYNOP configures the asynchronous DIM to ignore all traffic on the line. If you have a line that is not connected to a terminal, use this protocol to avoid wasting CPU time in attempts to interpret random noise on this line as valid commands.

```
    SET_ASYNC -LINE 51 -PRO TTYNOP
```

ASD: ASD allows PRIMOS to detect the line speed of a user terminal automatically. Depending on how you configure the line, PRIMOS can detect baud rates of 110, 300, 600, 1200, 2400, 4800, 9600, and 19200 bits per second. ASD can be used on lines attached to AMLC and ICS controllers provided that the line is set up for user login. The system rejects any attempt to issue the ASD protocol for an assignable line. For further information on ASD, see the following section, Enabling Auto Speed Detect.

Enabling Auto Speed Detect

Auto Speed Detect (ASD) is particularily useful if your system has dialup lines. A user who makes the telephone connection presses the carriage return key a few times. PRIMOS uses these carriage returns to determine the line's baud rate, and changes the rate at which it sends data out on the line to match the baud rate. After PRIMOS has determined the line speed, the "Login please." message is displayed and the user is allowed to log in. When the user logs out, or after a forced logout, the line returns to ASD.

First configure the line as you normally would with SET_ASYNC -PRO to specify the appropriate protocol for the line. Then explicitly set ASD for the line with another SET_ASYNC command.

```
SET_ASYNC -LINE 5 -PRO TTY -SPEED 9600

SET_ASYNC -LINE 5 -PRO ASD
```

The three configuration directives that affect ASD are ASYNC JUMPER, DISLOG, and DTRDRP. Use these directives in the configuration file as follows:

- Specify the valid jumper speeds with ASYNC JUMPER.
- Set DISLOG systemwide or for the specific line. This directive logs out the user if the user disconnects without logging out.
- Include DTRDRP. This directive drops the DTR signal and returns the line to ASD when the user logs out or is logged out because of DISLOG.

Note

> If you do not have the DTRDRP and DISLOG directives in your configuration file, you should set the AMLTIM disctime and gracetime parameters to values greater than 0.

How you use the ASYNC JUMPER directive depends on whether your system uses AMLC boards, ICS controllers, or both:

- For AMLC boards, the ASYNC JUMPER directive tells PRIMOS how the hardware jumpers are set. To set the jumpers to other non-default speeds, have a representative from your Customer Support Center change the hardware settings. If you do not use the ASYNC JUMPER directive with AMLC lines, only baud rates of 110, 300, 1200, and 9600 are detected.
- All AMLC controllers must be jumpered identically for ASD to detect all speeds on all AMLC controllers.

- For ICS controllers, the ASYNC JUMPER directive assigns the appropriate speeds to the lines.
- For a mixture of both AMLC boards and ICS controllers, the speeds in the ASYNC JUMPER directive must match the hardware-jumpered speeds on the AMLC board.

For further details on configuration directives, see Chapter 10, CONFIGURATION directives.

ALLOCATION OF INPUT AND OUTPUT RING BUFFERS

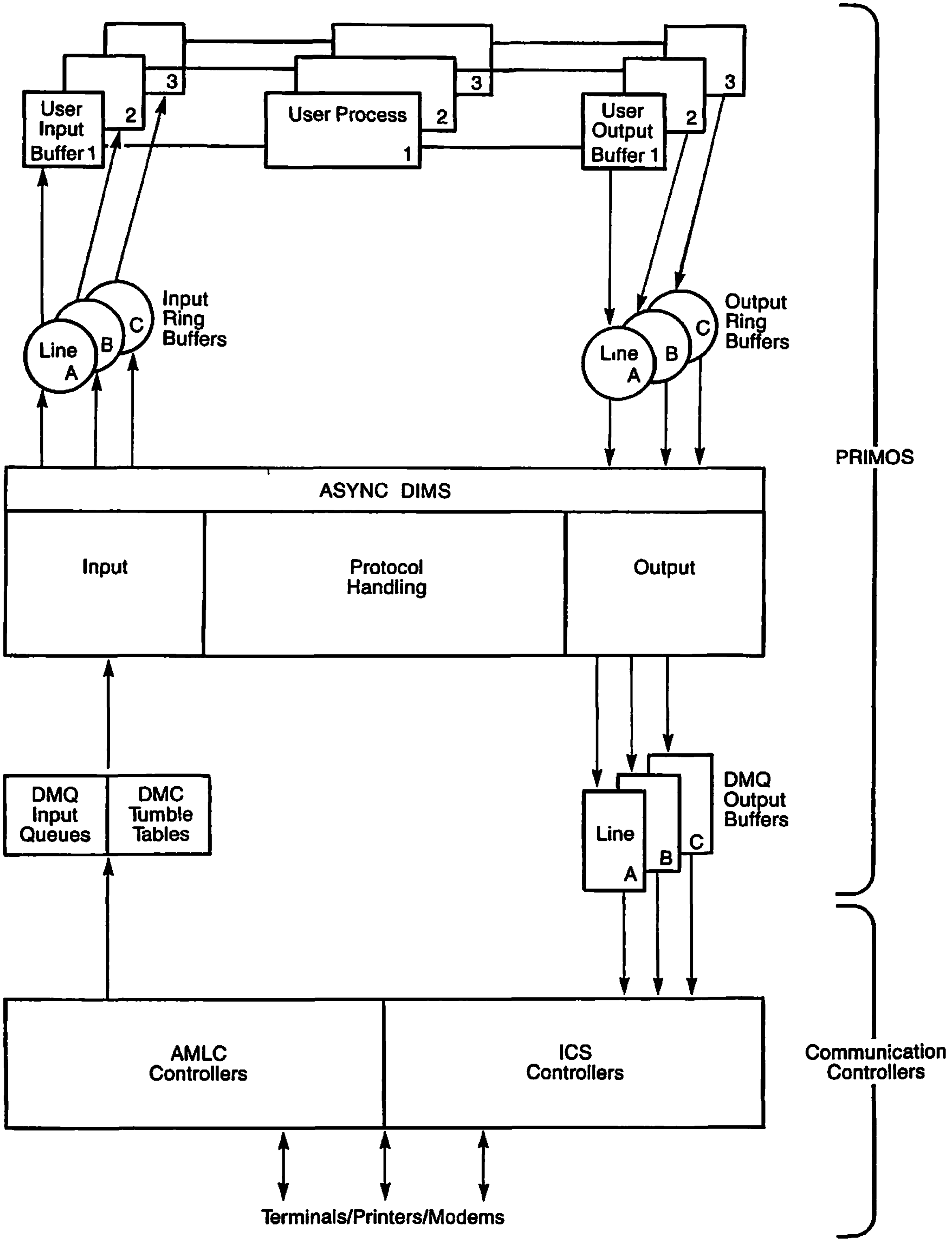
PRIMOS reserves memory during system configuration for all input and output ring buffers. These buffers are allocated at cold start first to terminal users, then to PRIMENET remote login users, and finally to assigned lines.

Figure 11-3 illustrates the flow of data from the input device to PRIMOS. The communication controllers receive characters and store them in either the Direct Memory Queue (DMQ) input queue or the DMC tumble table. Periodic I/O interrupts trigger the Asynchronous Device Interface Modules (ASYNC DIMs) to read the data. The ASYNC DIMs decode the bits that designate line number and protocol before transferring the character to the appropriate Input Ring Buffer (IRB). Processing of the characters from an IRB is performed on a per-user basis. Output characters are passed to the appropriate Output Ring Buffer (ORB). The ASYNC DIMs retrieve the data from the ORB, identify the line number, apply the communications protocol (for example TTYUPC), and pass the character to the appropriate DMQ output buffer. The communication controller accepts data from the DMQ output buffer at regular intervals and transmits the characters to the device.

DMQ output buffers are maintained for each line. IRBs and ORBs are maintained for each user. These three types of buffers reside permanently in memory (that is, they are wired), even when a particular line is not in use.

The AMLBUF directive defines the size of the IRBs, ORBs, and DMQ buffers. For more information on the AMLBUF directive refer to the section on AMLBUF later in this chapter and to Chapter 10, CONFIGURATION DIRECTIVES.

The DMC tumble tables and DMQ input buffer sizes are set with the AMLIBL and ICS INPQSZ directives, respectively. The REMBUF directive defines the size of input and output ring buffers for PRIMENET remote login users. For more information on these directives see Chapter 10, CONFIGURATION DIRECTIVES.



Input and Output Buffers and Queues for Asynchronous Lines
Figure 11-3

If you do not specify buffer sizes in the system configuration file, PRIMOS uses the default values. The default IRB and ORB values are sufficient for most devices, but the default DMQ input size is too small when the line speed is greater than 1200 baud.

Block-mode terminals, asynchronous links from one computer to another, serial graphics devices, and certain printers require larger buffer sizes to operate efficiently. Refer to the installation guide for your device for the buffer sizes recommended by the manufacturer, or consult your Customer Service Representative.

> **Caution**
>
> Assigned lines use a rotating buffer pool system that cannot guarantee that the same buffers are used if a line is unassigned and later reassigned. It is not possible (except at cold start) to predetermine which ring buffers a particular assigned line receives. Therefore, when configuring an assigned line for a device that requires special ring buffer sizes, it is advisable to set all the buffers in the pool to the largest size that any individual device requires.
>
> If the System Administrator can be certain of the order in which lines are assigned (for example, when all lines are assigned in PRIMOS.COMI), then this procedure may not be necessary.

Default Buffer Number Assignments

The DMQ buffer number for login lines is permanently associated with the physical line number. The default buffer numbers for login lines is equal to the physical line number plus two. Although it is possible to change the default buffer assignment with the SET_ASYNC -LINE n -USER_NUMBER m option, this is not recommended.

Default Buffer Capacity

The NTUSR directive automatically reserves a complete set of IRB, ORB, and DMQ buffers for every terminal user on your system. IRBs and ORBs store two characters per halfword; DMQ buffers store one character per halfword. The NRUSR directive reserves a set of buffers for remote users. The NAMLC directive reserves a third set of buffers to create the rotating pool of IRBs and ORBs for assigned lines. The default sizes and ranges are as follows:

Buffer Capacity in Halfwords

	Default	Minimum	Maximum
	Decimal (Octal)	Decimal (Octal)	Decimal (Octal)
IRB	128 ('200)	1 ('1)	4095 ('7777)
ORB	192 ('300)	50 ('62)	4095 ('7777)
DMQ	32 ('40)	16 ('20)	1024 ('2000)

Determining IRB Capacity: OAS, FORMS, PRIMEWAY, DPTX, and PRIME/SNA products use block-mode terminals, which require large IRBs. The default IRB size allows 255 type-ahead characters on terminals using character-mode input. When a burst-type device sends more than 256 characters in a burst, the buffer overflows and characters are lost. The overflow characters are not echoed back to the screen. If the terminal is in block mode when the IRB overflows and the screen termination characters are discarded, the terminal may lock.

The recommended IRB sizes for these products are as follows:

Application	Input Buffer Size
OAS on a PT65	832 ('1500)
DPTX	1984 ('3700)
FORMS/PRIMEWAY	832 ('1500) or Maximum Screen Size (from Application FD) + 2 x (Number of fields on screen)
PRIME/SNA	1024 ('2000) (24 x 80 only) 2048 ('4000) (24 x 80 and 27 x 132)

Determining ORB Capacity: The default ORB size is generally sufficient for all character-mode terminals. If full screens of data are painted at 9600 baud or greater, the ORB can be made larger to increase throughput. Setting the ORB size too small does not cause data loss but may reduce throughput for high-speed devices. Perceptible bursts of output may occur on block-mode terminals using the following applications: OAS on the PT65, FORMS, PRIMEWAY, DPTX, and PRIME/SNA. Data loss can be prevented by increasing the ORB capacity. PRIME MEDUSA, EDMS™, and other products that send graphics output over serial (asynchronous) lines require increasing the size of the ORB beyond the maximum screen size to 2000 ('3700).

For improved throughput set the ORB capacity to equal or exceed the number of characters that can be transferred to the line in 1/2 second.

$$\text{ORB capacity} = \frac{\text{characters per second}}{4}$$

For example, a 9600-baud line sends 960 10-bit characters per second to a laser printer. This is equivalent to 480 halfwords per second, or 240 halfwords every 1/2 second. Therefore, the ORB size for this line would be 240 ('360).

Determining DMQ Buffer Capacity: The DMQ buffer capacity depends on the line speed, character length, and the interrupt rate. The DMQ buffer store 1 character per halfword. Legal values are '2000, '1000, '400, '200, '100, '40, and '20. Use the following formula to calculate the DMQ size and use the next highest value in the table below.

$$\text{DMQ Size} = \frac{\text{line speed divided by bits per character}}{\text{I/O interrupt rate per second}}$$

For example,

$$96 = \frac{\text{9600 divided by 10}}{10}$$

The result of 96 is rounded up to the next valid size of 128 ('200).

When DMQ buffer capacity is too small, the effective speed of the device decreases. When the DMQ capacity is too large, devices with small input buffers can overflow. Serial printers often either have very small internal buffers, or wait until their buffers are nearly full before sending an XOFF (or using DSS) to tell PRIMOS to stop sending output. When the default interrupt rate is used, the recommended DMQ buffer sizes are as follows:

Interrupt Rate	Line Speed (Baud rate)	Characters Per Second	Recommended DMQ Capacity
10	50	5	16 ('20)
10	75	7	16 ('20)
10	110	10	16 ('20)
10	134.5	13	16 ('20)
10	150	15	16 ('20)
10	200	20	16 ('20)
10	300	30	16 ('20)
10	600	60	16 ('20)
10	1200	120	16 ('20)
10	1800	180	32 ('40)
10	2400	240	32 ('40)
10	3600	360	64 ('100)
10	4800	480	64 ('100)
10	7200	720	128 ('200)
10	9600	960	128 ('200)
10	19200	1920	256 ('400)

The AMLBUF Directive

The AMLBUF directive defines the capacities of the Input Ring Buffers (IRB), Output Ring Buffers (ORB), and Direct Memory Queue (DMQ) output buffers for login and assignable lines. AMLBUF directives are stored in your PRIMOS configuration file and set buffer sizes at cold start.

AMLBUF directives are necessary only for lines connected to devices that do not run efficiently with the default buffer sizes. This directive has two distinct formats as shown below. Each login line requires a separate AMLBUF directive. Assignable lines may require two directives.

For login lines, only one AMLBUF directive is necessary:

```
AMLBUF  {line-number    IRB-size   ORB-size    DMQ-size}
```

Assignable lines may require two AMLBUF directives. The first AMLBUF directive sets the DMQ size only and the second sets the ring buffer sizes.

```
AMLBUF  {line-number       0          0        DMQ-size}

AMLBUF  {index          IRB-size    ORB-size     0      }
```

It is not possible (except at cold start) to predetermine which ring buffers a particular assigned line receives. Therefore, it is recommended that you set the ring buffers for all assignable lines to the largest size that any individual device requires with AMLBUF directives.

The arguments to the AMLBUF directive are as follows:

Argument	Description
line-number	The physical line number required for setting the DMQ buffer.
index	The value used by PRIMOS to reference ring buffers for assignable lines. Index ranges from (NTUSR + NRUSR -1) to (NTUSR + NRUSR -1 + NAMLC).
IRB-size	The number of characters stored in the IRB in halfwords (two characters per halfword). The minimum value is 1 and the maximum is 4095 ('7777). The default value is 128 ('200).

Argument — Description

ORB-size — The number of characters stored in the ORB in halfwords (two characters per halfword). The minimum value is 50 ('62) and the maximum is 4095 ('7777). The default value is 192 ('300).

DMQ-size — The number of characters stored in the DMQ output buffer (one character per halfword). The value must be a power of 2 between 16 ('20) and 1024 ('2000) inclusive. If you specify a value less than 16 ('20), the system sets the size to '20. The default value is '40 (32 decimal).

The total of DMQ buffers on the system cannot exceed 64,000 halfwords.

If any of the buffer size arguments to the AMLBUF directive are finally set to zero or are omitted, the buffer size becomes the default or, in certain cases, becomes the last specified value.

Comments can be added at the end of each AMLBUF directive to make your PRIMOS.COMI file easier to read and modify.

For a complete description of AMLBUF, See Chapter 10, CONFIGURATION DIRECTIVES.

The AMLBUF directives are included in your PRIMOS.COMI file. The sample AMLBUF directives shown in Figure 11-4 set the IRB, ORB, and DMQ buffer sizes for a system with 20 asynchronous lines.

In Figure 11-4 there are 9 login lines, 6 lines for PRIMENET remote login users, 3 assignable lines, and a supervisor terminal. Therefore, NTUSR = 10 ('12), NRUSR = 6 ('6), and NAMLC = 3 ('3).

Login line number 4 is unused. The DMQ buffer is set to the minimum, 16 ('20) halfwords to conserve memory.

Login line 5 is connected to a PRIME MEDUSA workstation model number PW 153 which does require an assigned line. The default IRB and DMQ sizes are used and the ORB size is increased to 1024 (2000) halfwords.

Login line number 6 is connected to a PT65 terminal running OAS. The IRB and ORB sizes are increased to 832 ('1500) halfwords and 320 ('500) halfwords, respectively. The DMQ size is increased to 200 ('128).

Login lines 7 and 8 are configured for 9600 baud and 1200 baud respectively. The DMQ buffer sizes for both lines is increased to 200 '(128).

```
/*    Lines 0 to 3 are using the default buffer sizes.
/*    Buffer assignments login lines 4, 5, 6, 7, and 8.
/*    line number  IRB-size ORB-size DMQ-size
AMLBUF     4         0        0        20     /* Unused login line
AMLBUF     5        200     2000       40     /* New Medusa PW 153
AMLBUF     6       1500      500      200     /* PT65 running  OAS
AMLBUF     7         0        0       200     /* 9600  baud   line
AMLBUF    10         0        0       200     /* 1200  baud   line
/*
/*    DMQ buffer assignments for assignable lines
/*    line number  IRB-size ORB-size DMQ-size
AMLBUF    21         0         0       20     /* LQ Printer
/*
/*    Ring buffer assignments for assignable lines
/*    Set all IRBs and ORBs to the maximum size
/*    required by any individual device. 1024 ('2000)
/*        index    IRB-size ORB-size DMQ-size
AMLBUF    17        2000     2000       0     /* MEDUSA  PW 95, or
AMLBUF    20        2000     2000       0     /* computer link, or
AMLBUF    21        2000     2000       0     /* serial printer.
```

Figure 11-4
Sample AMLBUF Directives

The remaining login lines are connected to devices that operate well with the default buffer sizes.

The three assignable lines are used for a bi-directional letter quality printer, a serial graphics line for PRIME MEDUSA PW 95 workstation, and an incoming only computer-to-computer link.

The bi-directional printer is connected to physical line number 17 ('21). To prevent its small internal buffer from overflowing, set the DMQ buffer to the minimum value, 16 ('20).

Although the printer can use the default ring buffer sizes, the serial graphics line requires a large ORB and the computer-to-computer link requires a large IRB.

Using the formula given above, the ring buffer index for assignable lines ranges from 15 ('17) to 17 ('21). Set both the IRBs and the ORBs to 1024 ('2000) for all possible values of index 15 ('17), 16 ('20), and 17 ('21).

12 Using EDIT_PROFILE

This chapter describes when and how to use EDIT_PROFILE.

EDIT_PROFILE provides a tool with which the System Administrator controls and tailors system security. Using EDIT_PROFILE, you add individual users to your system and create and modify profiles for each of them. If you use access groups or use more than one project on your system, you also create and maintain these groups and projects through EDIT_PROFILE.

Both users and projects have profiles.

- A user profile defines a user ID's login password, Initial Attach Point, default login project, command environment limits, and membership in systemwide and project-base access groups.
- A project profile may define an Initial Attach Point and membership in access groups for all the members of a particular project. It may also define a set of command environment limits that can be used by any member who does not have those limits set.

Profiles, groups, and other terms used within EDIT_PROFILE, are defined more fully in Chapter 3, PLANNING THE USER ENVIRONMENT. Chapter 3 also illustrates several possible approaches to planning your system.

Plan your system before you create any profiles using EDIT_PROFILE. The worksheets in Chapter 3 may be helpful for planning. You should also read Chapter 5, SECURITY, before you use EDIT_PROFILE.

OVERVIEW OF EDIT_PROFILE

System Administrators use EDIT_PROFILE to do two kinds of work:

- To create a new System Administration Directory (SAD). The SAD contains a data base that includes information about the users of your system and any groups and projects you create. When you install a Rev. 20.2 system for the first time, you must create a SAD before users can log in.

- To maintain system security, and to create, change, and delete profiles for individuals and for projects. For example, use EDIT_PROFILE to register a user ID and other user attributes when you add a new user to your system.

You can have a maximum of 4096 projects on your system. If you have two or more projects on your system, you can delegate some of these tasks to Project Administrators. After you have registered the administrator of a project with EDIT_PROFILE, that person can use EDIT_PROFILE to maintain the project.

EDIT_PROFILE Modes

EDIT_PROFILE operates in the following three modes, each of which has a different purpose:

- Initialization mode allows you to create the SAD. EDIT_PROFILE prompts you with a series of questions. After you have answered them, EDIT_PROFILE sets up the SAD for you.

- System Administrator mode allows you to create, maintain, and delete profiles for users and projects, after you have created the SAD. To do these operations, use the EDIT_PROFILE commands described in this chapter. These commands also provide some system-level security controls.

- Project Administrator mode allows Project Administrators to use some EDIT_PROFILE commands to maintain their projects. Because Project Administrator commands are only limited versions of the System Administrator commands, a System Administrator need not use Project Administrator mode at all.

This chapter explains how to use EDIT_PROFILE in each mode. If you have a SAD and are using EDIT_PROFILE only to maintain an existing set of user profiles, you do not have to read the explanation of Initialization mode.

Because Project Administrators do not use EDIT_PROFILE in Initialization mode or System Administrator mode, they should concentrate on the discussion of Project Administrator mode. Project Administrators can refer to the dictionary of EDIT_PROFILE commands in the PRIMOS Commands Reference Guide.

Converting to Rev. 20.2 EDIT_PROFILE

Conversion to Rev. 20.2 from a Rev. 19.2 or later version of PRIMOS occurs automatically the first time EDIT_PROFILE is invoked on a Rev. 20.2 system.

INITIALIZATION MODE

While you are in Initialization mode, EDIT_PROFILE prompts lead you through the series of steps necessary to create the SAD.

Before using EDIT_PROFILE, you must decide whether you will create a system default project (which is always named DEFAULT).

Using Project DEFAULT

If you are not using ACLs, your system must have project DEFAULT and cannot support any other projects. Project DEFAULT is created automatically in Initialization mode.

If you are using ACLs, your system must have at least one project and can support up to 4096 projects. One of these projects may be project DEFAULT.

Project DEFAULT is necessary in either of the following two situations:

- If you are not going to create separate projects on your system. As long as DEFAULT is the only project on your system, all users that you register are automatically added to DEFAULT.
- If you prefer not to be prompted for a default login project for each new user you add to the system.

If you decide not to use the system default project, you must create at least one project on your system. (You cannot add users to the data base unless it has at least one project.)

The only time you can create project DEFAULT is in Initialization mode. (See Step 4 in the section below, Initialization Procedure.) If necessary, you can delete project DEFAULT later, using System Administrator mode.

If you decide to use project DEFAULT, you must define the attributes for the project in Initialization mode. See the section below, Defining the System Default Project.

Entering Initialization Mode

To enter Initialization mode, issue the EDIT_PROFILE command. The form of the command you use depends on the directory in which you want to create the SAD.

The SAD that controls access to your system must be stored in the MFD of the command partition. (The command MFD is on logical disk 0.) This section describes how to create a SAD on the MFD of the command partition. For an explanation of creating a SAD other than in the MFD of the command partition, see the section below, Creating a SAD Outside the Command MFD.

Because no user can log in before the SAD on the command MFD is created, you must run EDIT_PROFILE in Initialization mode from the supervisor terminal.

To create a new SAD in the MFD of the command partition, issue the EDIT_PROFILE command without a pathname, as in the following format:

```
EDIT_PROFILE   [ { -MFD_PASSWD } password ]
               [ { -MPW        }          ]
```

If the MFD is password-protected, you must use the -MFD_PASSWD option and specify the owner password for the MFD. (XXXXXX is the Prime-supplied password.)

After you issue the EDIT_PROFILE command, EDIT_PROFILE enters Initialization mode, provided that no SAD exists in the directory where you want to create one. If a SAD does exist there, EDIT_PROFILE enters System Administrator mode.

If no SAD exists in the directory, the following text appears after you issue the EDIT_PROFILE command:

```
OK, EDIT_PROFILE
[Edit_Profile Rev 20.2 (c) Prime Computer 1985]
In initialization mode.
SAD does not exist.  Create it?
```

The "Create it?" prompt is the first of a series of EDIT_PROFILE questions that lead to the creation of the SAD.

Initialization Procedure

The following steps describe the prompts in EDIT_PROFILE initialization procedure that allow you to create a SAD:

1a. SAD does not exist. Create it?

Type YES to create a SAD. Type NO to terminate EDIT_PROFILE and return to PRIMOS.

1b. Do you want to convert the MFD to an ACL directory?

The prompt appears only in a password-protected MFD.

Type YES to convert the password-protected MFD to an ACL-protected MFD, thus allowing you to use ACLs, projects, and groups. It is recommended that you convert the MFD to an ACL directory.

Type NO if you do not want to use ACLs or projects other than the system default project.

2. Projected number of users:

Either enter the total number of users you expect to have using your system or press RETURN for the default value of 20. EDIT_PROFILE is most efficient with 21,000 or fewer users.

EDIT_PROFILE always creates space for more users than you specify, to allow for growth and maximum efficiency in searching the User Profile Data Base. If later you add more users than the SAD can accommodate, EDIT_PROFILE displays the following warning message:

```
Warning:  User Validation file is overloaded.
```

To rebuild the data base to accommodate more users, use the EDIT_PROFILE REBUILD command, explained later in this chapter in the section, The REBUILD Command.

3. System Administrator name:

This prompt appears only when you are creating a SAD from the supervisor terminal. Otherwise, the ID of the user who invoked the EDIT_PROFILE command is automatically registered as the System Administrator.

Entering the name SYSTEM enables a user at the supervisor terminal to run EDIT_PROFILE. Entering any other name prevents this, which means that the System Administrator has to use a user terminal and log in under the ID specified at this prompt.

If the supervisor terminal is accessible to users and security is a concern, it is recommended that you enter a name other than SYSTEM when EDIT_PROFILE prompts you for the name of the System Administrator. This enables you to run EDIT_PROFILE from a user terminal under an identifier known only to the System Administrator and prevents users who can access the supervisor terminal from corrupting the SAD.

4. Create project "DEFAULT"?

   This prompt appears only if your system is using ACLs. This is the only time you can create the system default project.

   Type YES to create the system default project, which is always named DEFAULT.

   Type NO if you are sure that you will never want a system default project. If you answer NO, you must create at least one project on your system. (To create a project other than DEFAULT, use the ADD_PROJECT command, described later in this chapter.)

5. Set system-wide attributes for "user-id":
       Password:
       Groups:
       Default login project:

   user-id is the System Administrator's ID, which you entered at Step 3. The "Password:" prompt is displayed whether the MFD is protected by ACLs or by a password. Although you can enter a null password, the System Administrator should always have a non-null password, for reasons of security.

   The "Groups:" prompt appears only if you are using ACLs. Enter the names of the systemwide access groups to which the System Administrator belongs. The names are automatically added to the system data base.

   The "Default login project:" prompt appears only if you did not create project DEFAULT. Enter the name of a project that you will create later or press RETURN to omit it.

At this point, EDIT_PROFILE displays the messages that it has created the following four files, which are part of the SAD:

- User Validation File
- Master Project File
- Master Group File
- System Default File

If you are using ACLs, EDIT_PROFILE also displays a message that a new group, .PROJECT_ADMINISTRATORS$, has been added to the system. All Project Administrators on your system belong to this access group.

If you did not create project DEFAULT, the EDIT_PROFILE right angle-bracket (>) prompt is displayed. Initialization is complete, but you must add at least one project to your system before anyone can log in. If you created project DEFAULT, you are prompted for a definition of it, as explained in the next section.

Defining Project DEFAULT

If you created project DEFAULT, you are prompted for its project limits and then for its project profile. The following procedure begins with Step 6 because it immediately follows Step 5 in the preceding section.

6. Set limits for project "DEFAULT":

```
    Groups:
    Maximum number of command levels:
    Maximum number of live program invocations per command
    level:
    Maximum number of private, dynamic segments:
    Maximum number of private, static segments:
```

   At this prompt, you set the maximum limits for the project. When you define the project profile at Step 9 below, the profile's groups must be among the groups defined here and the four command environment values must be equal to or less than the values that you set here.

   The "Groups:" prompt appears only if you are using ACLs. At this prompt, enter the names of all the access groups that can be used later with project DEFAULT. The groups are automatically added to the project's data base.

   The next four prompts ask for the maximum limits for the four command environment values for project DEFAULT. An invalid entry produces an error message and displays the prompt again. See the chart below for valid values.

Attribute	Minimum	Maximum	Recommended
Command levels	1	100	10
Live invocations per level	1	100	5
Dynamic segments	16	504	40
Static segments	8	496	40

   The sum of the private dynamic and static segments may not exceed 512. To submit batch jobs, a user must have at least two command levels. (A user's batch jobs will fail if the user is set up with only one command level.)

When you define command environment values for individual users, no value may exceed these maximum limits, although they may be higher than the project profile values you enter at Step 10 below.

The System Administrator can change project limits later, using the CHANGE_PROJECT command in System Administrator mode, but a Project Administrator cannot change project limits.

7. Set attributes for user "user-id" in project "DEFAULT":
   Groups:
   Initial attach point:

   user-id is the ID of the System Administrator. This prompt defines the project-based attributes for the System Administrator when using the DEFAULT project.

   At the "Groups:" prompt, enter the IDs of any project-based groups to which the System Administrator will belong. These groups must be among those that you included in Step 6.

   At the "Initial attach point:" prompt, either enter the absolute pathname (including partition name) of the Initial Attach Point for the System Administrator in project DEFAULT, or press RETURN to omit it.

8. Create/change user attributes?

   Type NO if you do not want to specifically set the four values for command environment limits. The user uses the project defaults (defined in Step 6) for these limits when logging in to this project.

   Type YES if you want to set the user's command environment limits. The project limits (defined in Step 6) are displayed. You can set the values to be equal to or less than the values you entered at Step 6. If you enter only carriage returns at all four prompts, the command environment limits are not set. If, however, you enter at least one value at any prompt, a carriage return at another prompt results in the limit being set to the value displayed in the corresponding project limit prompt.

9. Set profile attributes for project "DEFAULT":
   Groups:
   Initial attach point:

   At Steps 9 and 10, you define the project profile. (The project profile can also be considered the default values for the project. Therefore, if you later add a user to this project and do not specifically set the user's project attributes, the user assumes the attributes of the project profile.)

   The "Groups:" prompt appears only on ACL systems. At the prompt, enter the project-based access groups to which members of DEFAULT will belong. These groups must be among those that you included in the project limits in Step 6.

   At the "Initial attach point:" prompt, enter the absolute pathname (including partition name) that will be the Initial Attach Point for users logging in as members of the project.

10. Attribute limits for the project:

    After this prompt, EDIT_PROFILE displays the command environment limits that you entered at Step 6 above. At each of the prompts for the four command environment attributes, enter a number that is equal to or smaller than the maximum limits defined in Step 6. If you enter only carriage returns at all four prompts, the command environment attributes are not set. If, however, you enter at least one value at any prompt, a carriage return at another prompt results in the attribute being set to the value displayed in the corresponding project limit prompt.

11. Check entry?

    If you type NO, initialization is complete and the EDIT_PROFILE prompt (>) is displayed.

    If you type YES, the values for the project limits and profile are displayed and EDIT_PROFILE asks you if you want to change them with the prompt "Change entry?".

Leaving Initialization Mode

After you have answered all the prompts in the initialization dialogue, the EDIT_PROFILE right angle-bracket (>) is displayed. This prompt tells you that the SAD has been created and that you are now in System Administrator mode.

At the > prompt, enter any EDIT_PROFILE command. When you are ready to quit, type QUIT (or Q) in response to the prompt.

Creating a SAD Outside the Command MFD

You can create a SAD outside the command MFD in any ACL-protected directory that does not already contain one. (A SAD that is not in the command MFD is often called a test SAD.) Creating a SAD outside the MFD of the command partition is useful for the following two reasons:

- For networked systems, you can create a SAD for a remote system to which your system is linked by PRIMENET. You can either do this directly on that system, or create the SAD on your local system and then copy it to the remote system.
- For testing purposes, you can create a new SAD without disrupting other users of your system. You can delegate the task to someone else and check that it has been done properly before using it as the control SAD in the MFD. You can also use a test SAD for practice using EDIT_PROFILE.

To create a SAD other than in the command partition MFD, use the following command format from any user terminal:

```
EDIT_PROFILE pathname
```

pathname is the pathname of the parent directory of the new SAD. The parent directory must be an ACL directory.

For example, to create a SAD on the partition SEA, in the subdirectory CHANNEL of the top-level ACL directory ENGLISH, give the command as follows:

```
EDIT_PROFILE <SEA>ENGLISH>CHANNEL
```

To create a SAD in the directory to which you are currently attached, issue the command in the following format:

```
EDIT_PROFILE *
```

The current directory must be an ACL directory.

Note

You cannot use Project Administrator mode on SADs that are outside the command MFD. Also, many of the EDIT_PROFILE system commands (described in the section below, SYSTEM COMMANDS) cannot be used in those SADs. If used, the commands display an error message similar to the following:

```
Change_sa command may not be used on test SADs.
```

Examples of Using Initialization Mode

Only on an ACL system can you specify more than one project and create access groups. Your dialog with EDIT_PROFILE therefore depends on whether you use ACLs and on where you create the SAD. The examples in the next three sections illustrate these differences.

Example of Initializing an ACL System: The following example shows how to create a SAD for a system using ACLs, projects, and groups. The System Administrator is working in a password-protected MFD and converts it to an ACL directory. The Administrator forgets to use the -MFD_PASSWORD option, receives an error message, and then uses the option correctly.

```
OK, EDIT_PROFILE
[Edit_Profile Rev 20.2 (c) Prime Computer 1985]
In initialization mode.
SAD does not exist.  Create it? YES
Do you want to convert the MFD to an ACL directory? YES
Insufficient access rights.  Converting MFD. (edit_profile)
ER! EDIT_PROFILE -MPW XXXXXX
[Edit_Profile Rev 20.2 (c) Prime Computer 1985]
In initialization mode.
SAD does not exist.  Create it? YES
Do you want to convert the MFD to an ACL directory? YES
*** From PRIMOS: Priority ACL set on partition "STAFF"
    by user "SYSTEM" (#1) at 04 Aug 86 10:36:52 Monday.
*** Creating User Validation File.  Projected number of users: 200
System administrator name: JOHN

Create project "DEFAULT"? YES

Set system-wide attributes for user "JOHN":
    Password: WASH
    Groups: .OPERATORS
*** New group added to system: ".OPERATORS".

User Validation File created 04 Aug 86 10:37:44
    268 entries in prime area; file is 13 records long.

Master Project File created 04 Aug 86 10:37:44

Master Group File created 04 Aug 86 10:37:44

System Default File created 04 Aug 86 10:37:44
*** New group added to system: ".PROJECT_ADMINISTRATORS$".

Set limits for project "DEFAULT":
    Groups: .DEALERS .KINGS
*** New group added to system: ".DEALERS".
*** New group added to system: ".KINGS".
```

```
    Maximum number of command levels: 10
    Maximum number of live program invocations per command level: 10
    Maximum number of private, dynamic segments: 50
    Maximum number of private, static segments: 50

Set attributes for user "JOHN" in project "DEFAULT":
    Groups:
    Initial attach point: <STAFF1>ADMIN
Create/change user attributes? NO

Set profile attributes for project "DEFAULT":
    Groups: .DEALERS
*** New group added to project: ".DEALERS".
    Initial attach point: <STAFF1>DEALERS

Attribute limits for the project:
    Maximum number of command levels: 10
    Maximum number of live program invocations per command level: 10
    Maximum number of private, dynamic segments: 50
    Maximum number of private, static segments: 50
-------------------------------------------------------
    Number of command levels: 5
    Number of live program invocations per command level: 5
    Number of private, dynamic segments: 40
    Number of private, static segments: 40
Project "DEFAULT" created.
    268 entries in prime area; file is 13 records long.
Check entry? YES

*****************************************************************************
Project: DEFAULT                          Administrator: JOHN
    Version 2 validation file.
    One entry in use out of 268.

Master project limits:
    Groups: .DEALERS .KINGS
Attribute limits for the project:
    Maximum number of command levels: 10
    Maximum number of live program invocations per command level: 10
    Maximum number of private, dynamic segments: 50
    Maximum number of private, static segments: 50
-------------------------------------------------------

Project profile:
    Groups: .DEALERS
    Initial attach point: <STAFF>DEALERS
    Number of command levels: 5
    Number of live program invocations per command level: 5
    Number of private, dynamic segments: 40
    Number of private, static segments: 40
************************************************************************
Change entry? NO
> QUIT
OK,
```

Example of Initializing a Non-ACL System

On a system that does not use ACLs, project DEFAULT is created automatically in Initialization mode. The System Administrator has to administer project DEFAULT. No other project can be created. Because you cannot use access groups without ACLs, no group-related questions are asked during initialization.

EDIT_PROFILE works correctly only when the SAD has a null owner password. The User Profile Data Base is much less secure on systems that do not use ACLs.

In the following example, project DEFAULT is created automatically because System Administrator JANE chooses not to use ACLs. JANE expects fifty users on the system, and she enters that number as the projected number of users. JANE then defines her personal characteristics in project DEFAULT, and the attributes of the project itself.

```
OK, EDIT_PROFILE -MPW XXXXXX
[Edit_Profile Rev 20.2 (c) Prime Computer 1985]
In initialization mode.
SAD does not exist.  Create it? YES
Do you want to convert the MFD to an ACL directory? NO
Warning: security and project support cannot be provided without ACLs.
*** From PRIMOS: Priority ACL set on partition "TEXT"
    by user "SYSTEM" (#1) at 21 Aug 86 11:08:56 Thursday.
*** Creating User Validation File.  Projected number of users: 50
System administrator name: JANE


Set system-wide attributes for user "JANE":
    Password: CALAMITY

User Validation File created 21 Aug 86 11:09:16
    92 entries in prime area; file is 5 records long.

Master Project File created 21 Aug 86 11:09:16

System Default File created 04 Aug 86 10:37:44

*** Creating project "DEFAULT".

Set limits for project "DEFAULT":
    Maximum number of command levels: 10
    Maximum number of live program invocations per command level: 10
    Maximum number of private, dynamic segments: 50
    Maximum number of private, static segments: 50

Set attributes for user "JANE" in project "DEFAULT":
    Initial attach point: <STARS1>JANE
Create/change user attributes? NO

Set profile attributes for project "DEFAULT":
    Initial attach point:

Attribute limits for the project:
    Maximum number of command levels: 10
    Maximum number of live program invocations per command level: 10
    Maximum number of private, dynamic segments: 50
    Maximum number of private, static segments: 50
---------------------------------------------------------
    Number of command levels: 10
    Number of live program invocations per command level: 10
    Number of private, dynamic segments: 40
    Number of private, static segments: 40
Project "DEFAULT" created.
    92 entries in prime area; file is 5 records long.
Check entry? YES

*************************************************************************
```

```
Project: DEFAULT                         Administrator: JANE
    Version 2 validation file.
    One entry in use out of 92.

Master project limits:
Attribute limits for the project:
    Maximum number of command levels: 10
    Maximum number of live program invocations per command level: 10
    Maximum number of private, dynamic segments: 50
    Maximum number of private, static segments: 50
------------------------------------------------------

Project profile:
    Initial attach point: <none>
    Number of command levels: 10
    Number of live program invocations per command level: 10
    Number of private, dynamic segments: 40
    Number of private, static segments: 40
**********************************************************************
Change entry? NO
> QUIT
OK,
```

Example of Initializing a SAD Outside the Command MFD: In the following example, user DAVE creates a SAD in his current directory. Because the SAD is not being created in the command MFD, EDIT_PROFILE enters the ID DAVE as the name of the System Administrator. DAVE chooses not to create project DEFAULT, which means that he must later create at least one project on the system (using the ADD_PROJECT command in System Administrator mode). Because DAVE did not create project DEFAULT, he is prompted for a default login project, but does not specify one.

```
OK, EDIT_PROFILE *
[Edit_Profile Rev 20.2 (c) Prime Computer 1985]
In initialization mode.
SAD does not exist.  Create it? YES
*** Creating User Validation File.  Projected number of users: 100
System administrator = "DAVE".

Create project "DEFAULT"? NO

Set system-wide attributes for user "DAVE":
    Password: VORPAL
    Groups: .ADMINISTRATORS
*** New group added to system: ".ADMINISTRATORS".
    Default login project:

User Validation File created 05 Aug 86 10:03:56
    124 entries in prime area; file is 6 records long.

Master Project File created 05 Aug 86 10:03:56

Master Group File created 05 Aug 86 10:03:56

System Default File created 05 Aug 86 10:03:56
>
```

SYSTEM ADMINISTRATOR MODE

After you have initialized the User Profile Data Base, you use EDIT_PROFILE in System Administrator mode. In this mode, you can use EDIT_PROFILE commands to add, change, and delete attributes of users and projects.

Table 12-1 lists the commands of EDIT_PROFILE that the System Administrator uses. You can use all these commands in System Administrator mode. The table also shows which commands a Project Administrator can use in Project Administrator mode.

As shown in the table, the commands can be divided into the following three categories:

- System commands provide control of the system as a whole. Using these commands, the System Administrator can enforce system requirements for the handling of passwords, list system information about users, groups, and projects, and perform other system-related tasks.

- Project commands provide control of all the projects on the system, including project DEFAULT, which is usually managed by the System Administrator. The System Administrator is the only person who can add or delete projects and change project limits, but Project Administrators can use the other commands to manage their own projects.

- User-control commands provide control of the attributes of individual users. The System Administrator is the only person who can verify users, or add or delete them from the system. Project Administrators can add or delete individual users from their own projects, or change a user's project-based attributes.

Each time you add a project to the system, you must specify a Project Administrator to manage the project. The Project Administrator can then use EDIT_PROFILE in Project Administrator mode, described later in this chapter. The System Administrator can administer all projects.

The following sections explain how to use each of the EDIT_PROFILE commands. To display online help screens, use the EDIT_PROFILE HELP command.

Table 12-1
EDIT_PROFILE Commands

Command	Used by	Function
System Command		
CHANGE_SYSTEM_ADMINISTRATOR	SA	Changes ID of SA
CHANGE_SYSTEM_DEFAULTS	SA	Changes the system defaults for the command environment attributes
FORCE_PASSWORD	SA	Prohibits the use of passwords on the login line
HELP	SA/PA	Displays syntax of all EDIT_PROFILE commands
LIST_SYSTEM	SA	Displays system and other attributes
MINIMUM_PASSWORD_LENGTH	SA	Sets minimum length for user passwords
NO_NULL_PASSWORD	SA	Prohibits use of null passwords
QUIT	SA/PA	Ends an EDIT_PROFILE session
REBUILD	SA/PA	Rebuilds the validation files
SET_DEFAULT_PROTECTION	SA	Restores protection to the SAD
SYSTEM_DEFAULTS	SA	Overrides the project-based and user-based command environment limits with system-default values

Table 12-1 (continued)
EDIT_PROFILE Commands

Command	Used by	Function
Project Command		
ADD_PROJECT	SA	Creates a new project
ATTACH_PROJECT	SA/PA	Specifies the current project
CHANGE_PROJECT	SA/PA	Changes the attributes of a project profile
DELETE_PROJECT	SA	Removes a project from the system
DETACH_PROJECT	SA/PA	Detaches current project
LIST_PROJECT	SA/PA	Lists the attributes of a project profile
User-control Command		
ADD_USER	SA/PA	Adds a user to the system or to a project
CHANGE_USER	SA/PA	Changes a user's system or project attributes
DELETE_USER	SA/PA	Removes a user from the system or from a project
LIST_USER	SA/PA	Lists a user's system or project attributes
VERIFY_USER	SA	Checks for existence of a user ID on networked systems

SYSTEM COMMANDS

The eleven commands described in this section allow the System Administrator to perform system-related tasks.

The CHANGE_SYSTEM_ADMINISTRATOR Command

Use the CHANGE_SYSTEM_ADMINISTRATOR command to change the user ID of the System Administrator. Such a change is necessary if another person will be administering the system or if you want to change your own user ID. After the change is made, only the new System Administrator can run EDIT_PROFILE in System Administrator mode. Before you use this command to change the System Administrator, make sure that user who will be the new System Administrator can log in on the system.

After you have entered the user ID of the System Administrator in Initialization mode, you cannot change the ID of the System Administrator until you have re-booted the system. The reason is that PRIMOS reads the ID of the System Administrator only when the system is booted, and does not allow the System Administrator be changed unless it recognizes the previous Administrator making the change.

The format of the CHANGE_SYSTEM_ADMINISTRATOR command is as follows:

```
{ CHANGE_SYSTEM_ADMINISTRATOR }  [user-id] [-ALL]
{ CSA                         }
```

user-id identifies the new System Administrator. If you do not specify user-id, you are prompted for it.

The -ALL option makes user-id the Project Administrator of any projects administered by the previous System Administrator. -ALL is assumed if your only project is DEFAULT.

After you issue the CHANGE_SYSTEM_ADMINISTRATOR command, you are prompted for a confirmation that you want to change the System Administrator. Typing YES (or Y) changes the System Administrator.

When the System Administrator is changed, EDIT_PROFILE changes all the ACLs protecting the SAD and its subdirectories to reflect the user ID of the new System Administrator. The changes are made in such a way that if you had previously altered these ACLs, the changes are lost. (Never alter these ACLs in any case.)

EDIT_PROFILE automatically terminates after the System Administrator has been changed.

The CHANGE_SYSTEM_DEFAULTS Command

Use the CHANGE_SYSTEM_DEFAULTS command to change the system defaults for command environment attributes. These attributes are the number of command levels, number of live invocations of programs per command level, number of private dynamic segments, and number of private static segments. After you change the defaults, the new defaults take effect the next time the system is cold started.

The format of the CHANGE_SYSTEM_DEFAULTS command is as follows:

```
{ CHANGE_SYSTEM_DEFAULTS }  option-1 [...option-4]
{ CSD                    }
```

You must supply at least one option to the command. If an option is not specified, the previous value of the attribute remains unchanged.

The options to the CHANGE_SYSTEM_DEFAULTS command are as follows:

```
{ -DYNAMIC_SEGMENTS }  n
{ -DS               }
```

Sets the system default for private dynamic segments to n. n must range from 16-504, inclusive. The Prime-supplied value for n is 32. The sum of private dynamic and static segments cannot exceed 512. EPFs use dynamic segments.

```
{ -LEVELS }  n
{ -LEV    }
```

Sets the system default for the number of command levels to n. n must range from 1-100, inclusive. The Prime-supplied value for n is 10. Users must have a minimum of two command levels to run batch jobs. PRIMOS uses command levels to allow users to suspend program invocations.

```
{ -PROGRAMS }  n
{ -PROG     }
```

Sets the system default for the number of live invocations of programs that can reside in a command level to n. n must range from 1-100, inclusive. The Prime-supplied value for n is 10.

```
{ -STATIC_SEGMENTS }  n
{ -SS              }
```

Sets the system default for private static segments to n. n must range from 8-496, inclusive. The Prime-supplied value for n is 32. The sum of private static and dynamic segments cannot exceed 512. Programs loaded with SEG and LOAD use static segments. Certain programs (such as DBG) require at least 32 segments.

The FORCE_PASSWORD Command

Use the FORCE_PASSWORD command to prevent PRIMOS from accepting passwords entered on the same line as the LOGIN command. Users must wait for the "Password?" prompt before typing a login password, which is not echoed on the terminal screen. If the password is supplied on the login line, the user is not allowed to log in and the following error message is displayed:

```
    Passwords may not be specified in the LOGIN command.
```

The format of the FORCE_PASSWORD command is as follows:

```
    { FORCE_PASSWORD } [ { -ON  } ]
    { FPW            } [ { -OFF } ]
```

The -ON option forces password prompts. -ON is the default. The -OFF option allows passwords on the login line.

See also the MINIMUM_PASSWORD_LENGTH and NO_NULL_PASSWORD commands later in this chapter.

The HELP Command

Use the HELP command to display information for one or all EDIT_PROFILE commands. The information includes the format, arguments, options, and option arguments.

The format of the HELP command is as follows:

```
    HELP [command-name]
```

command-name is an EDIT_PROFILE command. If you specify command-name, EDIT_PROFILE displays the command's format, argument (if any), and options (if any).

If you do not specify command-name, EDIT_PROFILE lists all commands, with their arguments and options. The output pauses after 22 lines of text and displays a "--More--" prompt. Type N, NO, Q, or QUIT to stop the output; press RETURN or type any character to display the rest of the output. The following example illustrates the output from the HELP command.

```
OK, EDIT_PROFILE
[Edit_Profile Rev 20.2 (c) Prime Computer 1985]
In system administrator mode.
> HELP
The following table lists the commands which the
profile editor accepts, along with a list of their
respective arguments and option names.  Capital
letters in the names show the abbreviations, e.g. "AU"
is the abbreviation for "Add_User."  For more detailed
information about each command, type "HELP <command_name>."

Command name        Argument    Options
------------        --------    --------------------------
Add_Project         project     -PA, -CReate_pa, -SIZE
                                -No_Query, -LIKE
Add_User            user        -LIKE, -PROJect, -PROFile, -No_Query
                                -SYStem, -DeFauLT
                                -PassWord, -Verify_NS
ATtach_Project      project     none
Change_Project      project     -PROFile, -SIZE, -LIST
                                -PA, -LIMits
Change_System_Administrator
                    SA name     -ALL
Change_System_Defaults
                    none        -Dynamic_Segments, -Static_Segments
                                -LEVels -PROGrams
--More--<CR>
Change_User         user        -PROJect -LIST
                                -SYStem -PassWord
Delete_Project      project     none
Delete_User         user        -PROJect
DeTach_Project      project     none
Force_PassWord      none        -ON, -OFF
HELP                command     none
List_Project        project     -PROFile, -USER, -ALL
                                -OUTput, -TTY, -APPend
List_System         none        -USers, -GRoups, -PROJects, -ALL
                                -OUTput, -TTY, -APPend
                                -DETail
List_User           user        -PROJect, -ALL
Minimum_PassWord_Length
                    length      none
No_Null_PassWord    none        -ON, -OFF
REbuild             none        -PROJect, -SIZE
Set_Default_PRotection
                    none        -CoNVert
System_Defaults     none        -ON, -OFF
Verify_User         user        -ALL

>
```

The LIST_SYSTEM Command

Use the LIST_SYSTEM command to display system, group, project, and user attributes, depending on the options you specify. The display may include the following system attributes. (Text within parenthesis explains why the attribute is present.)

- SAD not ACL-protected.

- System-wide groups enabled. (only on ACL systems, where they are always enabled)

- Project-based groups enabled. (only on ACL systems, where they are always enabled)
- Non-DEFAULT projects exist. (only on ACL systems)
- Passwords always requested at login. (FORCE_PASSWORD was used)
- Minimum password length is n. (MINIMUM_PASSWORD_LENGTH was used with a length of n characters)
- Null passwords not allowed. (NO_NULL_PASSWORD was used)

The format of the LIST_SYSTEM command is as follows:

$\left\{\begin{array}{l}\text{LIST\_SYSTEM}\\ \text{LS}\end{array}\right\}$ [options]

The options for the LIST_SYSTEM command are as follows:

Option	Meaning
-ALL	Lists all the information provided by the combination of the -USERS, -GROUPS, and -PROJECTS options.
-APPEND	Adds the output of the command to the end of the file specified with the -OUTPUT option. Use -APPEND only in conjunction with the -OUTPUT option. If you do not use -APPEND with -OUTPUT, the contents of the output file are overwritten.
-DETAIL	Lists additional information depending on the other options you select. With -DETAIL specified, -USERS includes the list of projects to which each user belongs, -GROUPS lists the membership of users and projects in each group, and -PROJECTS lists which users and groups belong to each project.
-GROUPS	Lists all groups on the system.
-OUTPUT pathname	Writes the output of the command into the file named pathname. If you specify a simple filename, the file is opened in the SAD. Use the -APPEND option also to prevent pathname from being overwritten. -OUTPUT is useful with the -ALL option, which may produce voluminous output.

-PROJECTS — Lists all projects on the system, with their attributes.

-TTY — Displays the output of the command at your terminal, which is the default. Use -TTY to send the output both to your terminal and to a file specified with the -OUTPUT option.

-USERS — Lists systemwide attributes of all system users.

If you use the LIST_SYSTEM command without any options, the output displays the ID of the System Administrator and a summary of system attributes, as shown in the following example:

```
OK, EDIT_PROFILE
[Edit_Profile Rev 20.2 (c) Prime Computer 1985]
In system administrator mode.
> LIST_SYSTEM

*****************************************************************************
<SYS01>SAD                      Administrator: SEGOVIA
    Version 2 validation file.
    One entry in use out of 1772.
    System-wide groups enabled.
    Project-based groups enabled.
    Null passwords not allowed.
    Passwords always requested at login.
    System default attributes are disabled at login.
    System default attributes:
    1. Maximum number of command levels is 10.
    2. Maximum number of program invocations is 10.
    3. Maximum number of private dynamic segments is 32.
    4. Maximum number of private static segments is 32.
*****************************************************************************
>
```

The MINIMUM_PASSWORD_LENGTH Command

Use the MINIMUM_PASSWORD_LENGTH command to set the minimum length for user passwords. The format of the MINIMUM_PASSWORD_LENGTH command is as follows:

```
{ MINIMUM_PASSWORD_LENGTH }  length
{ MPWL                    }
```

length is a decimal integer ranging from 0-16, inclusive. The default length is zero, which is equivalent to the command NO_NULL_PASSWORD -OFF. A length of one is equivalent to the command NO_NULL_PASSWORD -ON.

The MINIMUM_PASSWORD_LENGTH command overrides a previous NO_NULL_PASSWORD command. Similarly, issuing a NO_NULL_PASSWORD command overrides the current minimum password length.

After you have set a minimum password length, you cannot specify login passwords shorter than the number specified by length. Users cannot use the PRIMOS CHANGE_PASSWORD command with a new password shorter than length. Existing passwords are not affected.

When you set a minimum password length greater than zero, the user IDs of all users who have null login passwords are displayed, as shown in the following example. (The display does not list users whose passwords are at least one character in length but shorter than the newly defined minimum.)

```
> MINIMUM_PASSWORD_LENGTH 3
Warning:  the following users currently have null passwords:
    COLLEEN
    STEPHEN
    CAROLINE
>
```

If a minimum password length is in effect, the LIST_SYSTEM command displays the minimum length when the command is invoked.

The NO_NULL_PASSWORD Command

Use the NO_NULL_PASSWORD command either to prohibit or allow null passwords on your system. (A null password is a password with a length of zero.) Prohibiting null passwords improves system security.

The format of the NO_NULL_PASSWORD command is as follows:

```
{ NO_NULL_PASSWORD }  [ { -ON  } ]
{ NNPW             }  [ { -OFF } ]
```

The -ON option (which is the default) prohibits null passwords. After you issue the command to prohibit null passwords, users who have null passwords are listed so that you can assign passwords to them. The new passwords must be at least as long as specified by the MINIMUM_PASSWORD_LENGTH command.

The -OFF option allows the use of null passwords. PRIMOS allows null passwords unless you explicitly forbid it by using the -ON option.

After you have prohibited null passwords, no user can specify a null password with the CHANGE_PASSWORD command, nor can the System Administrator assign a null password to any user.

See also the FORCE_PASSWORD and MINIMUM_PASSWORD_LENGTH commands earlier in this chapter.

The QUIT Command

Use the QUIT command to terminate your EDIT_PROFILE session. Q is the abbreviation for QUIT.

The REBUILD Command

Use the REBUILD command to rebuild the User Profile Data Base, at either system level or at individual project level. You may want to rebuild the data base for the following reasons:

- If you have added many users to the system or to a particular project, EDIT_PROFILE issues a warning message indicating that a file is overloaded, which means you should rebuild it.
- If you expect to add many users, you may want to rebuild in anticipation of the increase.
- If you want the User Profile Data Base to be cleaned up, REBUILD accomplishes this by removing obsolete user entries.
- If you need to conserve disk space, REBUILD cleans up redundant material and allows you to specify the size of files.

Caution

Never use REBUILD while users can log in to your system. Use the operator command MAXUSR 0 before using the REBUILD command. See the *Operator's Guide to System Commands* for a description of MAXUSR.

The format of the REBUILD command is as follows:

```
    REBUILD [-PROJECT [project-id]] [-SIZE entry-count]
```

The -PROJECT Option: Use the -PROJECT option to rebuild files related to an individual project. If you do not specify *project-id*, EDIT_PROFILE assumes your current project (see the ATTACH_PROJECT command). If you have no current project, you are prompted for a project ID.

If you do not use the -PROJECT option, EDIT_PROFILE rebuilds the User Profile Data Base for the whole system. When you use REBUILD on a system with only one project, EDIT_PROFILE automatically rebuilds the project-related files every time you rebuild the system-related files.

The -SIZE Option: The -SIZE option specifies how many users you need space for, either in the system or the project-related data base. EDIT_PROFILE always allows space for at least 20 users, both for the system and for each project, and can accommodate up to 21,000 user profiles per system, and up to 20,000 per project.

If you do not use the -SIZE option, EDIT_PROFILE selects the new size of the user or project validation file. EDIT_PROFILE expands or decreases the size based on the number of entries currently in use in the prime area and the number of entries in use in the overflow area. (EDIT_PROFILE sets the size of validation files to a prime number chosen to make searching the data base as efficient as possible.)

Example of Using the REBUILD Command: In the following example, a System Administrator rebuilds the entire User Profile Data Base. Because the REBUILD command is issued without any options, EDIT_PROFILE selects the new size of the user validation file. The Administrator deletes the old files because no problems are encountered during the rebuilding.

```
OK, EDIT_PROFILE
[Edit_Profile Rev 20.2 (c) Prime Computer 1985]
In system administrator mode.
> REBUILD
*** UVF backed up into file "UVF.OLD" 07 Mar 86 11:02:28.
*** MPF backed up into file "MPF.OLD" 07 Mar 86 11:02:28.
*** MGF backed up into file "MGF.OLD" 07 Mar 86 11:02:28.

The following project id's have been removed from the MPF:
    PROD

*** MPP for project "DEFAULT" backed up into
    "DEFAULT>MPP.OLD" 07 Mar 86 11:02:32
*** MPP for project "DEALERS" backed up into
    "DEALERS>MPP.OLD" 07 Mar 86 11:02:36
*** MPP for project "PARTS" backed up into
    "PARTS>MPP.OLD" 07 Mar 86 11:02:36
*** MPP for project "FRITZ" backed up into
    "FRITZ>MPP.OLD" 07 Mar 86 11:02:36

*** Rebuild complete 07 Mar 86 11:02:36! ***
    Version 2 validation file.
    4 entries in use out of 20.
Delete old files? YES
>
```

The SET_DEFAULT_PROTECTION Command

Use the SET_DEFAULT_PROTECTION command to restore the default ACL protection in the SAD. (If possible, make sure that the default ACL is never changed.) SET_DEFAULT_PROTECTION also restores the default read/write lock settings in the SAD, both for password-protected and ACL-protected systems.

The format of the SET_DEFAULT_PROTECTION command is as follows:

```
{ SET_DEFAULT_PROTECTION }  [-CONVERT]
{ SDPR                   }
```

The -CONVERT option converts a password SAD to an ACL SAD.

The SYSTEM_DEFAULTS Command

Use the SYSTEM_DEFAULTS command to specify whether the system default command environment attributes or the project-based and user-based attributes are assigned to users when they log in.

The four command environment attributes are the number of command levels, the number of program invocations per command level, the number of private dynamic segments, and the number of static segments.

The format of the SYSTEM_DEFAULTS command is as follows:

```
{ SYSTEM_DEFAULTS }  [ { -ON  } ]
{ SD              }  [ { -OFF } ]
```

If -ON is specified, all users are logged in with the system default number of levels and segments. If -OFF is specified, project-based and user-based levels and segments are enabled. -ON is the default.

The SYSTEM_DEFAULTS -ON command is particularly useful when you are converting from a PRIMOS revision prior to Rev. 19.4. Prior to Rev. 19.4, command environment attributes did not exist, and therefore, no users or projects have their own values already set up. After you assign default values to your projects and users, you can turn off the system defaults.

PROJECT COMMANDS

The six commands described in this section are used to administer projects.

The ADD_PROJECT Command

Use the ADD_PROJECT command to create a new project on your system. Project Administrators cannot use this command.

When you use ADD_PROJECT, EDIT_PROFILE creates a new project directory in the SAD and defines the project according to the specified options.

Each time you add a project, you register the user ID of the person you want to be Project Administrator. If you specify a Project Administrator who is not yet a registered user of your system, EDIT_PROFILE asks you if you want to create a profile for the new user. If you type NO, the project is not added and the command terminates.

When you register a user as a Project Administrator, EDIT_PROFILE makes that user a member of the systemwide group .PROJECT_ADMINISTRATORS$. Because no user can belong to more than 16 system groups, you are queried if you register a Project Administrator who already belongs to 16 system groups. You must either delete one of the groups or not make the user a Project Administrator.

The format of the ADD_PROJECT command is as follows:

```
{ ADD_PROJECT } [project-id [options]]
{ AP          }
```

If you specify any options, you must also specify project-id, which is the name of the project to be created.

The options for the ADD_PROJECT command are as follows:

Option	Meaning
-CREATE_PA -CR	Specifies that you want to define the attributes of the Project Administrator as a member of the new project. (Project Administrators do not have to belong to the project that they administer.)
-LIKE reference	Specifies that the new project is to have the same attributes as reference, which is the ID of an existing project.

-NO_QUERY
-NQ

Stops EDIT_PROFILE from asking you whether you want to check or change the newly created project definition. Using -NO_QUERY is the same as typing NO at the check or change prompts.

-PA [user-id]

Specifies the user ID of the Project Administrator of the new project. If you do not use -PA or do not specify user-id, you are prompted for user-id.

-PROFILE
-PROF

Specifies that you will define the profile of the new project while creating it. If you do not use this option, the profile is set up with null entries.

-SIZE entry-count

Specifies how many users will belong to the project. If you do not use this option, EDIT_PROFILE assumes the default entry-count of 20 project members. For projects, as for the whole system, EDIT_PROFILE notifies you if you add more users than the data base can efficiently handle. This warning enables you to rebuild the data base, specifying a new size if you wish.

If the only project on your system is DEFAULT when you start an EDIT_PROFILE session, then DEFAULT is defined as your current project. However, as soon as you create another project, DEFAULT ceases to be your current project. You will not have a current project until you use the ATTACH_PROJECT command to specify a current project. (For information on current projects, see the ATTACH_PROJECT command below.)

Example of Using the ADD_PROJECT Command: In the following example, a System Administrator uses the ADD_PROJECT command to create a new project called DEALERS. Because the Project Administrator (whose ID is DLR_MAN) is not yet a registered user of the system, the System Administrator must register DLR_MAN before the project can be added.

The System Administrator then defines the project limits. (Project limits are the groups that can be used in the project, and the maximum command environment attributes that project members can have.) The System Administrator creates an entry for DLR_MAN as a member of the project, and finally creates and checks the project profile.

```
OK, EDIT_PROFILE
[Edit_Profile Rev 20.2 (c) Prime Computer 1985]
In system administrator mode.
> ADD_PROJECT
Enter project_id: DEALERS
Project administrator name: DLR_MAN
User DLR_MAN isn't registered, do you want to register DLR_MAN? YES

Set system-wide attributes for user "DLR_MAN":
    Password: DOLLAR
    Groups: .MANAGERS
*** New group added to system: ".MANAGERS".
    Default login project: DEALERS
*** New project added to system: "DEALERS".

User "DLR_MAN" added to system.
Check entry? NO
*** New group added to system: ".PROJECT_ADMINISTRATORS$".

Set limits for project "DEALERS":
    Groups: .CARS .PARTS
*** New group added to system: ".CARS".
*** New group added to system: ".PARTS".
    Maximum number of command levels: 10
    Maximum number of live program invocations per command level: 10
    Maximum number of private, dynamic segments: 32
    Maximum number of private, static segments: 32
Create administrator's entry? YES

Set attributes for user "DLR_MAN" in project "DEALERS":
    Groups: .CARS
*** New group added to project: ".CARS".
    Initial attach point: <MARKET>MANAGER
Create/change user attributes? YES

Attribute limits for the project:
    Maximum number of command levels: 10
    Maximum number of live program invocations per command level: 10
    Maximum number of private, dynamic segments: 32
    Maximum number of private, static segments: 32
------------------------------------------------------------
    Number of command levels: 10
    Number of live program invocations per command level: 10
    Number of private, dynamic segments: 32
    Number of private, static segments: 32
Create project profile? YES

Set profile attributes for project "DEALERS":
    Groups: .CARS .PARTS
*** New group added to project: ".PARTS".
    Initial attach point: <MARKET>DEALER

Attribute limits for the project:
    Maximum number of command levels: 10
    Maximum number of live program invocations per command level: 10
    Maximum number of private, dynamic segments: 32
    Maximum number of private, static segments: 32
------------------------------------------------------------
    Number of command levels: 5
    Number of live program invocations per command level: 5
    Number of private, dynamic segments: 16
    Number of private, static segments: 16
Project "DEALERS" created.
    20 entries in prime area; file is 1 record long.
Check entry? YES
```

```
**************************************************************************
Project: DEALERS                           Administrator: DLR_MAN
    Version 2 validation file.
    One entry in use out of 20.

Master project limits:
    Groups: .CARS .PARTS
Attribute limits for the project:
    Maximum number of command levels: 10
    Maximum number of live program invocations per command level: 10
    Maximum number of private, dynamic segments: 32
    Maximum number of private, static segments: 32
    ------------------------------------------------------

Project profile:
    Groups: .CARS .PARTS
    Initial attach point: <MARKET>DEALER
    Number of command levels: 5
    Number of live program invocations per command level: 5
    Number of private, dynamic segments: 16
    Number of private, static segments: 16
**************************************************************************
Change entry? NO
> QUIT
OK,
```

The ATTACH_PROJECT Command

Use the ATTACH_PROJECT command to specify a particular project as your current project. A current project serves as a default. If you use an EDIT_PROFILE command that allows you to specify a project ID and you do not specify the ID, the command is performed on your current project.

A project becomes your current project in one of three ways:

- If DEFAULT is the only project on a system, it is the current project.
- If you give the EDIT_PROFILE command using the -PROJECT option to specify a project ID, that project becomes the current project.
- If you use the ATTACH_PROJECT command, the project you specify becomes the current project.

The format of the ATTACH_PROJECT command is as follows:

```
{ ATTACH_PROJECT }  [project-id]
{ ATP            }
```

project-id is the project that will be your current project. If you do not specify project-id, you are prompted for it.

See also the DETACH_PROJECT command later in this chapter.

The CHANGE_PROJECT Command

Use the CHANGE_PROJECT command to change the attributes or the size of a project. The format of the CHANGE_PROJECT command is as follows:

```
{ CHANGE_PROJECT } [project-id [options]]
{ CP             }
```

project-id identifies the project to be changed. You must specify project-id to use an option.

If you enter a blank line in response to any of the CHANGE_PROJECT prompts, no change is made to the attribute specified in the prompt.

The options to CHANGE_PROJECT are as follows:

Option	Meaning
-LIMITS	Specifies that you want to change the master project limits. (Limits refer both to access groups and command environment attributes for the project.)
-LIST	Displays the project attributes after other changes you specify in the command line have been made.
-PA [user-id]	Specifies that you are changing the Project Administrator of the project to user-id. If you omit user-id, you are prompted for it. (See the ADD_PROJECT command earlier in this chapter for details on registering a new Project Administrator.)
-PROFILE	Specifies that you want to change the profile of the project. Two examples of using -PROFILE are associating a new ACL group with the project and changing the limits for command environment attributes.
-SIZE [entry-count]	Specifies that you want to change the amount of space reserved in the User Profile Data Base for information related to the project. entry-count specifies the number of project members for whom you wish space allocated. If you omit entry-count, you are prompted for it.

Using the -SIZE Option: The -SIZE option, which conserves disk space, is the only way to control the entry count with the CHANGE_PROJECT command. If, however, you are not changing other project attributes, the REBUILD command is recommended when changing the entry-count.

Specifying Access Groups: When changing project attributes, you are prompted to enter the project's access groups. If you want only to add or delete a group from the list, you need not reenter the entire list. To add a group, reply in the following format to the "Groups:" prompt:

```
-ADD groupname-1 [...groupname-n]
```

To delete a group from the list, reply in the following format to the "Groups:" prompt:

```
-DELETE groupname-1 [...groupname-n]
```

Example of Using the CHANGE_PROJECT Command: In the following example illustrating the use of the CHANGE_PROJECT command, the command environment limits of project DEALERS are changed and the access group .LABOR is added.

```
OK, EDIT_PROFILE
[Edit_Profile Rev 20.2 (c) Prime Computer 1985]
In system administrator mode.
> CHANGE_PROJECT
Enter project_id: DEALERS
Change administrator? NO
Change project profile? NO
Change project limits? YES

Master project limits:
    Groups: .CARS .PARTS
Attribute limits for the project:
    Maximum number of command levels: 10
    Maximum number of live program invocations per command level: 10
    Maximum number of private, dynamic segments: 32
    Maximum number of private, static segments: 32
------------------------------------------------------

Set limits for project "DEALERS":
    Groups: -ADD .LABOR -DELETE .PARTS
*** New group added to system: ".LABOR".
    Maximum number of command levels: 5
    Maximum number of live program invocations per command level: 5
    Maximum number of private, dynamic segments: 32
    Maximum number of private, static segments: 16
Project "DEALERS" updated 06 Aug 86 08:31:16.
>
```

The DELETE_PROJECT Command

Use the DELETE_PROJECT command to remove a project from your system. The DELETE_PROJECT command cannot be used on a non-ACL system nor can it be used by Project Administrators.

If any project members are using the project when you issue the command, a prompt allows you to change your mind. If you delete a project that is a user's default login project, that user cannot log in unless the user is a valid member of another project, and specifies that project when logging in.

The format of the DELETE_PROJECT command is as follows:

```
  { DELETE_PROJECT }  [project-id]
  { DP             }
```

project-id is the name of the project to be deleted. If you have a current project and omit project-id, the current project is deleted. If you have no current project and omit project-id, you are prompted for the project ID. (Current project is described under the ATTACH_PROJECT command earlier in this chapter.)

Example of Using the DELETE_PROJECT Command: In the following example, the System Administrator deletes the project DUMMY.

```
OK, EDIT_PROFILE
[Edit_Profile Rev 20.2 (c) Prime Computer 1985]
In system administrator mode.
> DELETE_PROJECT
Project to delete: DUMMY
Project "DUMMY" currently contains 5 entries.
Do you want to delete it? YES
*** Project "DUMMY" deleted 07 Aug 86 11:12:44.
    (3 default projects reset.)
>
```

The DETACH_PROJECT Command

Use the DETACH_PROJECT command to clear the setting of a current project set by a previous ATTACH_PROJECT or other EDIT_PROFILE command. The format of the DETACH_PROJECT command is as follows:

```
  { DETACH_PROJECT }  [project-id]
  { DTP            }
```

You need not specify project-id, which is your current project.

After using the DETACH_PROJECT command, you have no current project. If you subsequently want to issue an EDIT_PROFILE command that affects a project, you must either use the ATTACH_PROJECT command first or specify the project ID in the command line.

See also the ATTACH_PROJECT command earlier in this chapter.

The LIST_PROJECT Command

Use the LIST_PROJECT command to list the attributes of a project. Attributes listed always include the project limits, and may include user and other attributes, depending on the options you select.

The format of the LIST_PROJECT command is as follows:

```
{ LIST_PROJECT }  [project-id [options]]
{ LP           }
```

project-id is the project to be listed. You must specify project-id to use an option.

The options to the LIST_PROJECT command are as follows:

Option	Meaning
-ALL	Lists the profiles of all project members.
-APPEND	Adds the output of the command to the end of the file specified with the -OUTPUT option. Use -APPEND only in conjunction with the -OUTPUT option. If you do not use -APPEND with -OUTPUT, the contents of the output file are overwritten.
-OUTPUT pathname	Writes the output of the command into the file named pathname. If you specify a simple filename rather than a full pathname, the file is opened in the SAD. Use the -APPEND option also to prevent the contents of the specified output file from being overwritten. The -OUTPUT option is particularly useful with the -ALL option, which may produce voluminous output.
-PROFILE	Lists the project profile, which shows project-based groups, command environment attributes, and the Initial Attach Point.

-TTY — Displays the output of the command at your terminal, which is the default. Use -TTY to send the output both to your terminal and to a file specified with the -OUTPUT option.

-USER user-id — Lists the profile of the specified project member. To list only user attributes without project attributes, use the LIST_USER command, described below.

Example of Using the LIST_PROJECT Command: The following example shows the listing of project DEFAULT, where the Administrator has chosen to list the project profile as well as the master project limits.

```
OK, EDIT_PROFILE
[Edit_Profile Rev 20.2 (c) Prime Computer 1985]
In system administrator mode.
> LIST_PROJECT DEFAULT -PROFILE

******************************************************************************
Project: DEFAULT                          Administrator: CHRIS
    Version 2 validation file.
    One entry in use out of 1772.

Master project limits:
    Groups: .ADMIN .SALES .PARTS .PERSONNEL
Attribute limits for the project:
    Maximum number of command levels: 10
    Maximum number of live program invocations per command level: 10
    Maximum number of private, dynamic segments: 32
    Maximum number of private, static segments: 32
-------------------------------------------------------

Project profile:
    Groups: .ADMIN
    Initial attach point: <SYS1>DEF
    Number of command levels: 10
    Number of live program invocations per command level: 10
    Number of private, dynamic segments: 32
    Number of private, static segments: 32
******************************************************************************
>
```

USER-CONTROL COMMANDS

The five commands described in this section are used to administer the attributes of individual users.

The ADD_USER Command

Use the ADD_USER command to add a user to the system, a project, or both, and to create the user's profile. The format of the ADD_USER command is as follows:

```
{ ADD_USER }  [user-id [options]]
{ AU       }
```

user-id is the user to be added. You must specify user-id to use an option.

The options to the ADD_USER command are as follows:

Option — **Meaning**

```
{ -DEFAULT } [project-id]
{ -DFLT    }
```

Specifies the project to which the user is being added, and makes that project the user's default login project. -DEFAULT implies the -SYSTEM option. You cannot specify both -PROJECT and -DEFAULT.

If you do not specify -DEFAULT when adding a user to the system, you are prompted for the user's default login project, unless the only project on your system is project DEFAULT. In this case, project DEFAULT is the user's default login project.

If you have a current project and you omit project-id, EDIT_PROFILE assumes your current project. If you do not have a current project and you omit project-id, you are prompted for a project ID.

-LIKE user-id2

Specifies that the new user is to have the same attributes as an existing user named user-id2. If you also specify a project with the -DEFAULT or -PROJECT options, user-id2 must belong to that project.

-NO_QUERY
-NQ

Stops EDIT_PROFILE from asking you whether you want to check or change the newly created user profile. Using -NO_QUERY is the same as typing NO at the check or change prompts.

-PASSWORD [password]

Specifies a login password for the new user whom you are adding to the system. This option implies the -SYSTEM option. You are prompted for a password if you do not use the -PASSWORD option or if you use it but omit the password argument. (If you allow null passwords on your system, you may specify a null password by entering only a carriage return at the prompt.)

-PROFILE

Specifies that you want to create the user's profile explicitly (by responding to EDIT_PROFILE prompts). If you do not use this option, the profile is set up from the default attributes in the project profile. You must use -PROFILE in conjunction with the -PROJECT option to set a user's command environment attributes.

-PROJECT [project-id]

Specifies the project to which you are adding the user. (Although a user can belong to several projects, you can add a user to only one project at a time.) You must use -PROJECT when adding a user for whom you want to set individual command environment attributes. This option does not affect the user's default login project. You cannot specify both -PROJECT and -DEFAULT.

If you omit project-id, EDIT_PROFILE assumes your current project. If you omit project-id and you do not have a current project, you are prompted for a project ID.

-SYSTEM

Specifies that you are adding the user to the system. -SYSTEM is the default in System Administrator mode. -SYSTEM implies both -PASSWORD and -DEFAULT.

{ -VERIFY_NS -VNS }	Searches the SADs of the systems that are attached to your system by PRIMENET and that recognize user IDs defined on your system, to determine whether the new user ID already exists on another system. If the user ID does exist on another system, a warning message is displayed, listing the PRIMENET nodenames of the systems where identical user IDs were found. The -VERIFY_NS option helps prevent duplication of user IDs across the network.

Because the -SYSTEM option is the default in System Administrator mode, the new user ID is added only to the system, except in the following situations:

- If you specify the -DEFAULT or -PROJECT options, you explicitly add the user to a project.
- If the only project on your system is project DEFAULT, all users are automatically added to DEFAULT when you add them to the system.
- If you specify no options and you have a current project, the user is added to the current project.

To add a user to a project rather than to the system, use the -PROJECT option, and do not use -PASSWORD, -SYSTEM, or -DEFAULT.

Note

To enter any limits for an individual user's command environment attributes, you must specify both the -PROJECT and the -PROFILE options.

Specifying Access Groups: When you add a new user to the system or to a project, you are prompted to enter the groups to which the user will belong. If, after you enter the names of those groups, you check the entry and decide to add or delete a group from the user's list, you need not reenter the entire list.

To add a group to the user's list, reply in the following format to the "Groups:" prompt:

```
-ADD groupname-1 [...groupname-n]
```

To delete an access group from the user's list, reply in the following format to the "Groups:" prompt:

```
-DELETE groupname-1 [...groupname-n]
```

Example of Using the ADD_USER Command: In this example, user ALFRED is added to a system. The System Administrator uses the -VERIFY_NS option because the system is on a network. Although the ID ALFRED is found on two other systems, the Administrator creates the user profile.

The System Administrator makes ALFRED a member of the group .KINGS. After checking the entry, the Administrator decides that ALFRED should also belong to the group .EARLS, and therefore types YES at the "Change Entry?" prompt. After the Administrator adds the group .EARLS to ALFRED's groups, ALFRED belongs to .KINGS and .EARLS.

```
OK, EDIT_PROFILE
[Edit_Profile Rev 20.2 (c) Prime Computer 1985]
In system administrator mode.
> ADD_USER ALFRED -VERIFY_NS
Warning: user "ALFRED" found on system(s):
    ENGL
    UK.1

Set system-wide attributes for user "ALFRED":
    Password: CAKES
    Groups: .KINGS
*** New group added to system: ".KINGS".
    Default login project: SAXON

User "ALFRED" added to system.
Check entry? YES

***********************************************************************
System-wide attributes for user "ALFRED":
    Groups: .KINGS
    Default login project: SAXON
***********************************************************************
Change entry? YES

System-wide attributes for user "ALFRED":
    Groups: .KINGS
    Default login project: SAXON

Set system-wide attributes for user "ALFRED":
    Groups: -ADD .EARLS
*** New group added to system: ".EARLS".
    Default login project:
User "ALFRED" updated 10 Aug 86 15:52:08.
>
```

The CHANGE_USER Command

Use the CHANGE_USER command to change the attributes of an existing user. You can alter systemwide attributes, project-based attributes, or both.

The format of the CHANGE_USER command is as follows:

```
{ CHANGE_USER } [user-id [options]]
{ CU          }
```

user-id is the ID of the user whose attributes are to be changed. You must specify user-id to use any option.

The options to the CHANGE_USER command are as follows:

Option	Meaning
-LIST	Lists the user's attributes after the changes have been made.
-PASSWORD [password]	Specifies a new login password for the user. If you omit password, you are prompted for it.
-PROJECT [project-id]	Specifies that you are changing the user's project-based attributes in the project identified by project-id. If you omit the project ID, EDIT_PROFILE assumes your current project. If you omit the project ID and have no current project, you are prompted for a project ID. You must specify -PROJECT to change a user's command environment attributes.
-SYSTEM	Specifies that you are changing the user's systemwide groups, default login project, or command environment attributes.

If you enter a blank line in response to any of the CHANGE_USER command's prompts, the previous value remains in effect.

Specifying Access Groups: When you change a user's attributes, you are prompted to enter the groups to which the user will belong. If you want only to add or delete a group from the user's list, you need not reenter the entire list.

To add a group to the user's list, reply in the following format to the "Groups:" prompt:

```
-ADD groupname-1 [...groupname-n]
```

To delete a group from the user's list, reply in the following format to the "Groups:" prompt:

```
-DELETE groupname-1 [...groupname-n]
```

Example of Using the CHANGE_USER Command: The following example shows how the System Administrator uses the CHANGE_USER command to change the attributes of user CHRIS in project PARTS.

CHRIS's groups and Initial Attach Point do not change because the Administrator enters a carriage return in reply to those prompts. Because the Administrator wants to change the command environment limits of CHRIS, she replies YES to the "Create/change user attributes?" prompt.

The command environment limits for the project are displayed, to remind the Administrator that the user cannot be assigned limits that exceed those of the project. The Administrator then enters the new command environment limits for user CHRIS.

```
OK, EDIT_PROFILE
[Edit_Profile Rev 20.2 (c) Prime Computer 1985]
In system administrator mode.
> CHANGE_USER CHRIS -PROJECT PARTS

Attributes for user "CHRIS" in project "PARTS":
    Groups: .PARTS
    Initial attach point: <SYS1>PARTS
    Number of command levels: 5
    Number of live program invocations per command level: 5
    Number of private, dynamic segments: 32
    Number of private, static segments: 16

Set attributes for user "CHRIS" in project "PARTS":
    Groups:
    Initial attach point:
Create/change user attributes? YES

Attribute limits for the project:
    Maximum number of command levels: 5
    Maximum number of live program invocations per command level: 5
    Maximum number of private, dynamic segments: 32
    Maximum number of private, static segments: 16
------------------------------------------------------
    Number of command levels: 2
    Number of live program invocations per command level: 2
    Number of private, dynamic segments: 16
    Number of private, static segments: 8
User "CHRIS" updated 06 May 86 10:41:52.
>
```

The DELETE_USER Command

Use the DELETE_USER command to remove a user from your system or from a project. When you delete a user from the system, the user is also removed from all projects to which the user belongs.

The format of the DELETE_USER command is as follows:

```
{ DELETE_USER } [user-id [-PROJECT [project-id]]]
{ DU          }
```

user-id is the user whom you are deleting. If you do not specify user-id, you are prompted for it.

The -PROJECT Option: If you do not specify the -PROJECT option, the user is removed from the system and from all projects. If you specify the -PROJECT option, the user is removed only from the specified project.

If you specify -PROJECT but omit project-id, EDIT_PROFILE assumes your current project. If you do not have a current project and omit the project ID from the -PROJECT option, you are prompted for a project ID. (For an explanation of current projects, see the ATTACH_PROJECT command earlier in this chapter.)

Examples of Using the DELETE_USER Command: The following three examples illustrate the use of the DELETE_USER command at both the system and the project level.

In the first example, the System Administrator removes user JOEY from the Administrator's current project (which is project DEFAULT), and therefore does not have to specify the project ID.

```
> DELETE_USER JOEY -PROJECT
*** User "JOEY" deleted from project "DEFAULT" 25 Aug 86 11:05:48.
>
```

In the second example, the System Administrator must explicitly specify the project ID when removing user TOM_TURKEY from project THANKS because the Administrator's current project is DEFAULT.

```
> DELETE_USER TOM_TURKEY -PROJECT THANKS
*** User "TOM_TURKEY" deleted from project "THANKS" 25 Aug 86 11:05:56.
>
```

In the third example, the System Administrator removes user JIMMY from the system.

```
> DELETE_USER JIMMY
*** User "JIMMY" deleted from system 25 Aug 86 11:07:00.
*** User "JIMMY" deleted from project "EDUCATION" 25 Aug 86 11:07:00.
*** User "JIMMY" deleted from project "BAD_BOYS" 25 Aug 86 11:07:04.
    (Project "BAD_BOYS" is now empty.)
*** User "JIMMY" deleted from project "DEFAULT" 25 Aug 86 11:07:08.
>
```

The LIST_USER Command

Use the LIST_USER command to list a user's attributes. Depending on the command format, the attributes listed are one of the following:

- Systemwide only (if DEFAULT is not the only project)
- Systemwide and as a member of one project
- Systemwide and as a member of all the user's project

The format of the LIST_USER command is as follows:

```
{ LIST_USER }   [user-id   [option]]
{ LU        }
```

user-id is the user whose attributes are to be listed. If you do not specify user-id, you are prompted for it. You cannot use either option unless you specify user-id on the command line.

The two options for the LIST_USER command are listed below. You cannot specify both options at the same time. If you do not specify an option and DEFAULT is not the only project, the command lists only the systemwide attributes of the user.

Option	Meaning
-ALL	Lists the user's attributes systemwide and in all the user's projects.
-PROJECT [project-id]	Lists the user's attributes systemwide and as a member of project project-id. EDIT_PROFILE assumes -PROJECT if DEFAULT is the only project on your system. If you omit project-id, EDIT_PROFILE assumes your current project. If you omit project-id and you do not have a current project, you are prompted for a project ID.

Example of Using the LIST_USER Command: In the following example, the LIST_USER command lists the attributes of user CAROL in all her projects.

```
OK, EDIT_PROFILE
[Edit_Profile Rev 20.2 (c) Prime Computer 1985]
In system administrator mode.
> LIST_USER CAROL -ALL

***********************************************************************
System-wide attributes for user "CAROL":
    Groups: .CARS .PARTS
    Default login project: PARTS

Attributes for user "CAROL" in project "PARTS":
    Groups: .PARTS
    Initial attach point: <SYS1>PARTS
    Number of command levels: 5
    Number of live program invocations per command level: 5
    Number of private, dynamic segments: 32
    Number of private, static segments: 16

Attributes for user "CAROL" in project "SALES":
    Groups: .EAST
    Initial attach point: <SALES>EAST
    Number of command levels: 5
    Number of live program invocations per command level: 5
    Number of private, dynamic segments: 50
    Number of private, static segments: 50
***********************************************************************
>
```

The VERIFY_USER Command

Use the VERIFY_USER command to find out if user IDs on your system also exist on remote systems. If the command finds identical IDs elsewhere, it displays a list of the PRIMENET nodenames of the systems that have the duplicate IDs. The other systems must be connected to your system with PRIMENET and they must recognize the IDs defined on your system. You cannot use VERIFY_USER if your system is not on a network.

The VERIFY_USER command, like the -VERIFY_NS option of the ADD_USER command, helps prevent duplication of user IDs across the network.

The format of the VERIFY_USER command is as follows:

```
{ VERIFY_USER }  { user-id }
{ VU          }  { -ALL    }
```

If you specify user-id, the SADs of the other systems are searched only for that ID and, if found, a list of the systems that have the duplicate IDs is displayed.

If you specify the -ALL option, the SADs of the other systems are searched for all the IDs on your system. If duplicate IDs are found, a list of the systems that have the duplicate IDs is displayed.

PROJECT ADMINISTRATOR MODE

A system that does not use ACLs can support only the system default project, named DEFAULT, and the System Administrator is also the Administrator of DEFAULT. On a non-ACL system, therefore, the System Administrator does not use Project Administrator mode in EDIT_PROFILE.

On an ACL system, the System Administrator may define more than one project and delegate some of the work of maintaining projects. When creating a project (other than project DEFAULT), the System Administrator has to specify the user ID of someone as Project Administrator for that project. A Project Administrator can then use a limited set of EDIT_PROFILE commands in Project Administrator mode.

A Project Administrator can change the attributes only of members of the particular project (or projects) that he or she administers. The following discussion is addressed to Project Administrators.

Note

> Project Administrator mode cannot be used on test SADs (that is, on SADs created outside the command MFD).

Entering Project Administrator Mode

To use EDIT_PROFILE in Project Administrator mode, you must specify the -PROJECT option with the project ID of your project. The Project Administrator therefore issues the command in the following format:

```
EDIT_PROFILE [partition-name] -PROJECT project-id
```

Supply partition-name only when your project is not on your local system. If your project is on a system other than the one to which you logged in, specify the name of the partition (MFD) that contains the SAD in which your project is kept.

For example, suppose you are Project Administrator for project HARKNESS on the partition HAMPER, which is on system SYS.H.

If you are logged in on system SYS.H, issue the EDIT_PROFILE command as follows:

```
EDIT_PROFILE -PROJECT HARKNESS
```

If, however, you are logged in to a system other than SYS.H, issue the EDIT_PROFILE command as follows:

```
EDIT_PROFILE <HAMPER> -PROJECT HARKNESS
```

In either format, the following message is displayed when you enter EDIT_PROFILE:

```
[Edit_Profile Rev 20.2 (c) Prime Computer 1985]
In project administrator mode.
>
```

PROJECT ADMINISTRATOR COMMANDS

Project Administrators can use the following EDIT_PROFILE commands:

Command	Meaning
ADD_USER	Adds a new member to the project
CHANGE_PROJECT	Changes the profile of the project
CHANGE_USER	Changes the attributes of an existing individual project member
DELETE_USER	Removes a user from the list of project members
HELP	Lists main argument, options, and option arguments for one or all of the EDIT_PROFILE commands available in Project Administrator mode
LIST_PROJECT	Lists the attributes of the project and of one or more project members
LIST_USER	Lists the attributes of an individual project member
QUIT	Ends an EDIT_PROFILE session
REBUILD	Rebuilds project lists and project files

If you manage more than one project, you may also use the ATTACH_PROJECT and DETACH_PROJECT commands, which are described earlier in this chapter, in the section PROJECT COMMANDS.

The ADD_USER Command in Project Administrator Mode

Use the ADD_USER command to add a user to your project and to define the user profiles. The format of the command is as follows:

```
{ ADD_USER } [user-id [options]]
{ AU       }
```

user-id is the user who is to be added to your project. If you do not supply user-id, you are prompted for it. You must specify user-id to use an option.

If you specify user-id without the -PROFILE option, the user is added to your project with a user profile containing the default attributes described in the project profile. To establish a different profile, use the -PROFILE option.

The ADD_USER options that Project Administrators may use are as follows:

Option	Meaning
-LIKE user-id2	Specifies that the new user is to have the same attributes as an existing user named user-id2.

{ -NO_QUERY }
{ -NQ }
	Stops EDIT_PROFILE from asking you whether you want to check or change the newly created user profile. Using -NO_QUERY is the same as typing NO at the check or change prompts.

-PROFILE	Specifies that you want to create the user's profile explicitly (by responding to EDIT_PROFILE prompts). If you do not use this option, the user profile is set up from the default attributes in the project profile. You must use -PROFILE in conjunction with the -PROJECT option to set a user's command environment attributes.
-PROJECT [project-id]	Specifies the project to which you are adding the user. If you omit project-id, EDIT_PROFILE assumes your current project. Use -PROJECT if you administer several projects or if you are adding a user for whom you want to set individual command environment attributes.

The CHANGE_PROJECT Command in Project Administrator Mode

Use the CHANGE_PROJECT command to change the profile of your project. The format of the command is as follows:

```
{ CHANGE_PROJECT }  [project-id [options]]
{ CP             }
```

The three Project Administrator options are -LIST, -PROFILE (abbreviated -PROF), and -SIZE. Use the -PROFILE option to change the project profile. For details on these options, see the CHANGE_PROJECT command described in the section PROJECT COMMANDS, earlier in this chapter.

The CHANGE_USER Command in Project Administrator Mode

Use the CHANGE_USER command to change the user profile of a member of your project. Note that your System Administrator may restrict the attributes that you can change. For example, a Project Administrator may assign access groups for project members only from the list of groups assigned to that project by the System Administrator.

The format of the CHANGE_USER command is as follows:

```
{ CHANGE_USER }  [user-id  [-PROJECT [project-id]] [-LIST]]
{ CU          }
```

user-id is the project member whose attributes are to be changed. If you do not supply user-id, you are prompted for it. You must specify user-id to use an option.

The -PROJECT option is useful only if you administer several projects or if you want to change a user's command environment limits within the maximum boundaries as set by the System Administrator.

The -LIST option displays the user's attributes after the changes.

The DELETE_USER Command in Project Administrator Mode

Use the DELETE_USER command to delete a user from your project. The command format is as follows:

```
{ DELETE_USER }  [user-id [-PROJECT [project-id]]]
{ DU          }
```

user-id is the user who is to be deleted from your project. If you do not supply user-id, you are prompted for it.

The -PROJECT option is useful only if you administer several projects.

The LIST_PROJECT Command in Project Administrator Mode

Use the LIST_PROJECT command to list the attributes of the specified project, as well as those of either one or all users in the project. The list always includes the project limits imposed by your System Administrator.

The command format is as follows:

```
{ LIST_PROJECT } [project-id  [options]]
{ LP           }
```

Specify project-id to use an option or to list the attributes of another project (other than the current project) that you administer.

For details on the options, see the LIST_PROJECT command described earlier in this chapter, in the section entitled PROJECT COMMANDS.

The LIST_USER Command in Project Administrator Mode

Use the LIST_USER command to display the attributes of an individual member of your project. The format of the command is as follows:

```
{ LIST_USER }  user-id  [ { -ALL                    } ]
{ LU        }           [ { -PROJECT   [project-id] } ]
```

user-id is the member of your project whose attributes are to be listed. If you do not specify user-id, you are prompted for it.

The -ALL option displays the user's attributes in each project to which the user belongs, provided that you administer that project.

The -PROJECT option displays the user's attributes in the project named project-id, which by default is your current project. Use this option if you administer several projects.

You cannot use the -ALL and -PROJECT options at the same time.

The REBUILD Command in Project Administrator Mode

Use the REBUILD command to rebuild your project to hold more members. Project members cannot log in to your project while EDIT_PROFILE rebuilds it.

The format of the command is as follows:

```
    REBUILD [-PROJECT project-id] [-SIZE entry-count]
```

Use the -PROJECT option if you administer several projects.

Use the -SIZE option to specify the total number of members you want in the project. The total should include the number of new members you expect to add to the project. If you do not use -SIZE, EDIT_PROFILE determines the new project size, based on the current total of project members.

CARE OF THE SAD

On systems using ACLs, EDIT_PROFILE automatically generates the ACL protecting the SAD, if the SAD is not ACL-protected already. The System Administrator is given ALL rights and all other users (identified as $REST) are given only List (L) and Use (U) rights.

It is extremely important that anyone acting as System Administrator observe the following rules:

- Do not alter the ACLs protecting the SAD or its contents. Any change in the ACLs may allow breaches in the security of your system, or may cause EDIT_PROFILE to work incorrectly.
- Do not alter the read/write locks protecting the SAD.
- Do not try to copy individual parts of the SAD. When copying the SAD, you must copy its entire contents, using the -COPY_ALL option of the COPY command.
- Keep a backup copy of your SAD in case it gets damaged. A copy of the SAD on a backup disk or tape would serve the purpose.

If the ACLs on the SAD or its contents, or the read/write locks on its contents are altered, restore them to their original settings by using the SET_DEFAULT_PROTECTION command in System Administrator mode.

PART III

Maintaining the System

13
Equipment and Environment

After the system is up and running, the System Administrator is responsible for overseeing the day-to-day operation of the system. A major part of this operation is the maintenance of the environment and the hardware of the system, which is the subject of this chapter.

Other maintenance tasks described elsewhere in this book include controlling interactions between users and the system (Chapters 5, 9, 12, and 15), setting schedules for backups (Chapter 14), maintaining system software (Chapter 16), and monitoring system usage (Chapter 17).

If you need assistance with problems in these areas, call your Customer Support Center.

ENVIRONMENTAL AND HARDWARE MAINTENANCE

The hardware of the system is any physical part, such as the CPU, disk or tape drives, printers, terminals, and other peripherals.

The environment of the system is the physical space in which, and the conditions under which, the hardware is set up and functioning. The environment includes the machine room temperature and humidity controls, air filtration equipment, and electric power.

The System Administrator's responsibility for environmental and hardware maintenance includes the following tasks:

- Defining user and operator procedures for handling hardware and environmental processes.
- Ensuring that disks and tapes are handled and stored properly
- Establishing a set of machine room rules
- Defining rules for emergencies

These tasks are described in the following sections.

USER AND OPERATOR PROCEDURES

You can decide whether users are allowed in the machine room. If you allow users into the machine room, make sure that they are trained to use the machines correctly. However, it is generally a good idea to restrict access to the machine room to those people who are essential to the operation of the system.

Note

> Remember that giving users access to the supervisor terminal gives them access to the entire system. Almost all privileged commands can be given from that terminal, no matter who is using it.

Other procedures you may want to set up are the following:

- Rules about who may use the machine room and such peripherals as printers and plotters.
- Training sessions for anyone who will use the machines and the machine room.
- Procedures by which users can request operations on machines to which they do not have physical access.
- Procedures by which users can inform you or the operator of problems with the hardware and software. (Users should always inform you or the operator of problems with terminals or other equipment. They should never attempt to do repairs themselves.)

Procedures vary from installation to installation. The System Administrator determines the procedures for an installation. For assistance, call your Customer Support Center.

HANDLING DISKS AND TAPES

Handle disks and tapes with care. Disks, in particular, are fragile. Improper handling of disks may damage the disk, which may cause a head crash when that disk is used.

Handle disks with care whether they are inside or outside their protective covers. Disks should not be carried in large piles because the disks may be damaged if dropped. Dropping a disk may distort the platters or crack the magnetic surfaces. A damaged disk pack may also damage the disk drive. If a disk has been dropped, it should not be used until it has been inspected by a technician.

Disk drives should not be banged or kicked when a disk is mounted because damage to the disk and disk drive, as well as loss of data, may occur.

Tapes are not as fragile as disks. However, careless handling can stretch, crease, scratch, or soil the tape. Even a fingerprint on the tape may cause a problem. If a tape is damaged, data may be lost. If the loss occurs at the beginning of the tape, the entire tape may be unusable.

STORING DISKS AND TAPES

The storage requirements for disks and tapes are similar, although the temperature and humidity ranges for tapes are a little larger than those for disks. Call your Customer Support Center if you are unsure whether your disk and tape storage meets requirements.

Keep a log book in the storage area. The log book should contain information about every disk and tape in the archive. Label all disks and tapes with their contents and date of creation, and enter this information in the log book. When a tape or disk is taken from storage, make an entry in the log book showing the name and date of creation of the tape or disk, the date of withdrawal, and the name of the person who takes it. This practice ensures that the whereabouts of all storage media is always known.

Storage media that contain confidential information should be kept in a special secure area. This area may be a locked strong box, cupboard, or room, depending on the number of disks and tapes to be stored. Establish special rules for access to this area.

MACHINE ROOM RULES

Your machine room contains various devices that are designed to keep the computer system at the right temperature and humidity, and to exclude most environmental contaminants. These devices include heating systems, air conditioners, air filters, sealed windows, and anti-static mats. Environmental problems result if operators or users circumvent or alter these devices. You must therefore establish a set of rules that govern the machine room.

General Rules for the Machine Room

The set of rules that you establish for the machine room should include the following four rules:

Rule 1. Prohibit smoking, food, and beverages in the machine room. There should be no exceptions to this rule.

Rule 2. Keep the machine room free of dust and other contaminants.

Rule 3. Maintain the machine room environment within the temperature and humidity limits specified by your Customer Support Center.

Rule 4. Keep the machine room closed to unauthorized personnel.

These rules are discussed in the remaining sections of this chapter. You will probably want to set other rules that are essential to your installation, but these four rules are essential to the smooth operation of the machines.

Smoking, Food, and Beverages

Smoking, food, and beverages are contaminants to a computer system, particularly to disk and tape storage media and their attendant drives. A smoke particle or a fingerprint is larger than the space between a disk's surface and the moving read/write head above it.

A head crash, therefore, can be caused by careless handling of a disk or by the intake of smoke through the drive. Head crashes usually occur when the head hits a particle of smoke or dust on the spinning platter, causing serious damage to the head and disk.

All personnel should wash their hands before handling magnetic media. A doughnut eaten at coffee break can leave a residue on the fingers that may cause major problems if that person handles a tape. If the surface of the tape becomes sticky, the contaminant may be transferred to the read/write heads during normal operation. Reel-to-reel tapes, whose recording surfaces are handled by an operator during a load, are especially susceptible to this type of contamination.

Dust and Dirt

Dust can cause a major malfunction of the system. A speck of dust on one of the disks can cause a head crash and the loss of many days' work. While you cannot completely eradicate the possibility of a head crash, you can make it much less likely to occur, and make sure that, if it does happen, you can recover from it. Recovery from head crashes and other data loss situations is discussed in Chapter 14, BACKUPS.

Paper dust from a printer can be a major source of airborne dust. Your printers should be vacuumed regularly by your servicing agency or by your own personnel. As an alternative, you can put the printers in another room.

If the machine room has filters on the air intakes to trap airborne dust, do not leave the doors and windows open, because the air filtering system will not work properly. If the machine room does not have filtered air, you can reduce the amount of airborne dust by keeping the windows sealed and the doors closed as much as possible.

Cleaning

All machine rooms should be cleaned regularly with vacuum cleaners. Do not use brooms or dry mops, because they throw dust into the air and increase contamination. Air filters on machines (such as disk drives) and the heads on tape drives should be cleaned regularly.

Consult your Customer Support Center to discuss which cleaning operations should be carried out by your staff, which should be left to your Customer Support Center, and how often the cleaning should be performed. After such a consultation, draw up in-house maintenance schedules (including the methods and rules) for jobs to be done by your own personnel and maintenance schedules for jobs to be handled by the Customer Support Center or other outside personnel.

Environmental Controls

Environmental controls are important because your machines give optimum performance only within the range of operating environments specified by your system installer. Moreover, failure to conform to the environmental specifications may invalidate your sales or support contracts. Prime computer systems are designed to operate at temperatures between 68 and 78 degrees Fahrenheit (20 and 26 degrees Celsius) and at humidities between 40% and 60%.

If the machine room regularly exceeds the maximum temperature or humidity requirements, do not try to solve the problem by opening doors or windows, because this lets in dust. Resolve this problem by consulting with your manager and a representative from your Customer Support Center.

If the system has a severe overheating problem, shut down the system until the problem can be resolved. Opening the windows and doors may keep the system running, but it may also cause problems (such as head crashes) that are troublesome and expensive to resolve.

Make sure the space around the machines is kept clear. Store cables and boxes of supplies (such as printer paper) away from all hardware. Obstructions may impede the airflow around the machine, which can cause overheating even if you have reliable air conditioning. The obstructions may also cause accidents and interfere with exit routes.

Unauthorized Personnel

You are responsible for deciding who is allowed into the machine room. Your operators must have access to it, and some users may also need access.

However, unauthorized personnel in the machine room can cause problems such as misusing the supervisor terminal and mishandling the equipment. Users trying to load tapes on tape drives or paper on printers can damage that equipment if they do not know the correct method.

Keeping the machine room doors closed helps prevent access by unauthorized personnel. If you cannot or do not want to lock the machine room, you should ensure that every person who is allowed in is adequately trained to use the machines it contains.

Installation-specific Rules

Because you know the special requirements of your installation, you must decide exactly what rules, other than those listed above, are necessary.

Most installation-specific rules deal with how authorized personnel use the machine room and its equipment, including who performs specific functions and how tapes and disks are moved and stored.

EMERGENCIES IN THE MACHINE ROOM

There are many kinds and degrees of emergencies. Major emergencies range from the system suddenly going down (perhaps from an electrical problem), to the illness of a key operator, to a disaster such as a fire that destroys the entire machine room. This section deals with the more commonplace kinds of emergencies.

Accidental Data Loss

Perhaps the most common emergency is the accidental erasure of data from a storage medium. These erasures are generally caused by system crashes, disk crashes, loss of electrical power, voltage spikes, and human errors. If you have repeated system crashes, perform tape dumps for use by the System Analyst.

If you have a set of recent backup tapes or disks, an accidental loss of data need not be disastrous. At worst, your users lose the data entered since the last backup.

> **Caution**
>
> Disks involved in head crashes must never be mounted again on any drive. If such a damaged disk is used, the read/write heads of the drive will be ruined by the loose magnetic oxide from the damaged disk surface. Similarly, drives involved in the head crashes must be serviced before they can used, because drives that are not serviced will ruin disks.

System Halts

System crashes have a variety of causes. To help you discover these causes, you should maintain an up-to-date system log book and perform tape dumps if you have frequent system crashes.

If the system is crashing often, try to determine if the crashes share any common conditions. For example, have the crashes always occurred during a thunder storm? Is the site near a potential source of electromagnetic radiation, such as an arc-welding shop or a physics laboratory? You can often detect such coincidental occurrences by checking the system log book and by checking COMOUTPUT files generated during system monitoring sessions about the time the trouble started.

If the system has previously been stable, check for changes in the area near your installation. For example, a new company that has moved next door may have machinery that produces electromagnetic radiation.

When you have surveyed the possibilities, you may have the answer to your problems. If so, call your Customer Service Representative to discuss methods of alleviating the problem. If not, call your Customer Support Center for help.

Warm Starts Versus Cold Starts: When your system halts (either because of a system error or because you halted it), you must decide whether to perform a warm start or a cold start. Use the following general guidelines in making this decision.

- Cold starts incur more risks than warm starts. Cold starts have a far greater potential than warm starts of causing a loss of data and a broken file system (such as mismatched pointers and damaged directories).

- Warm starts maintain the integrity of the file system and almost never cause the loss of data.

- Warm starts increase system availability because they cause only a slight interruption of the system and do not require the use of FIX_DISK to maintain file system integrity.

- If a halted or hung system cannot be warm started successfully and must be cold started to get it running, it is recommended that you run FIX_DISK on all partitions to ensure the integrity of the file system. Using this procedure is time-consuming, but failure to use FIX_DISK may result in a loss of files in the future because of a broken file system. You must decide between the tradeoff of availability (getting the system running quickly by not running FIX_DISK) and reliability (using FIX_DISK).

- Use of the MEMHLT NO configuration is recommended, but only if your system is serviced regularly and if it is not running ROAM-based data management products. If you use MEMHLT NO and your system still halts because memory parity errors, you should have your system serviced in the near future. Otherwise an undetectable or falsely corrected error may occur because the system is running with faulty memory.

- Warm starts are recommended for a system that is not running application programs with their own built-in recovery procedures. Examples of such application programs are the ROAM-based DISCOVER, PRISAM, and DBMS products.

- Systems running ROAM-based products should use the MEMHLT YES configuration directive and should always be cold started after any halt. A warm start may cause loss of data. In addition, FIX_DISK should be run on the partitions.

For detailed information on system halts, see your CPU handbook.

Accidents

Accidents that occur in the machine room can often be prevented. Check the machine room periodically for possible danger points. If you cannot remove a hazard immediately, post a warning about it.

A cable snaking across the floor is a good example of a potential danger for at least three reasons: an employee tripping over it may break a bone; it is a potential source of electrocution; and, if stepped on, it can cause a system crash, resulting in loss of data.

Electric Shock

While extremely rare, electric shock is the greatest hazard in the computer room. The high voltage electric currents that computer systems use are dangerous if handled incorrectly.

In many cases, the effects of an electric shock can be mitigated by quickly applying cardiopulmonary resuscitation (CPR) techniques to the victim. If possible, have at least one person trained in CPR in or near the machine room at all times. (This person may also be useful if an employee has a heart attack.)

Alert all personnel of the danger of electric shock. Under no circumstances should they touch any internal components of the system.

14
Backups

This chapter provides some guidelines to help you plan your strategy for backups. The commands and procedures used to perform backups are explained in the Operator's Guide to System Backups.

REASONS FOR BACKUPS

A backup operation is a procedure for making a tape or disk copy of the current contents of the system's online storage devices (that is, disks). These copies are available if data is lost, or if a user needs a file in the form it was in at the backup date.

Major losses of data may be caused by the following:

- Hardware problems, such as disk crashes
- Environmental problems in the machine room, such as overheating
- Operator errors, such as running MAKE on a disk that is in use
- Natural catastrophes, such as fires and electrical storms

Minor losses may be caused by the following:

- Power failure before or during a write operation
- Accidental truncation or deletion of a file by a user or an operator (this is the most common cause of data loss)

If your system suffers a major loss of online data, your only hope of recovery is to have a recent copy of the lost data. Such a copy should have been created by your most recent backup operation. Using this copy, you can restore your entire data base as it was on the date that the backup copy was made.

If the loss is minor, you need to restore only a small part of the backup copy. On a system with good backup procedures, either major or minor restoration can be performed easily whenever necessary.

GUIDELINES FOR BACKUPS

Each site has different needs for backups. The following are some questions you should consider when deciding on your backup procedure:

- What data should be backed up?
- How much does your system data change from day to day?
- Are all backups going to be full backups, or are some of them going to be incremental backups?
- How often and at what hours should backups be performed? (Keep in mind that you must restrict access to the disk while performing the backup and that the task requires some operator time.)
- How quickly can you restore the system to its pre-crash state?
- What media should be used (disk-to-disk backups, disk-to-tape backups, or a mixture of the two)?
- Where and for how long should the backup copies be stored?
- Do you have any users with special backup needs?

This chapter provides some guidelines for you to use in answering these questions.

TYPES OF BACKUPS

A backup copy is made to a disk or to a tape and is either a full backup or an incremental backup. Each type of backup has its advantages and disadvantages. The following paragraphs discuss disk-to-disk and disk-to-tape backups and full and incremental backups.

Note

Whether you are backing up to tape or disk, perform backups under PRIMOS, not PRIMOS II. PRIMOS II cannot write on Rev. 20 disks and cannot save access control or quota information.

Disk-to-disk Backups

The COPY_DISK command copies the contents of one disk to another disk. Disk-to-disk copies are fast. Typically, a fully used 300 megabyte disk pack can be copied in about one hour. Smaller packs take less time.

More information can be held on a single disk than on a single tape. A full 300 megabyte disk requires nine reels of tape when recorded at a density of 1600 bpi.

A disk backup can be used directly without requiring a restore operation. The disk is immediately and rapidly accessible in the normal way, using the directory tree structures. (Both a current disk and one of its backups can run simultaneously, if you change the name of one of the two disks when you add it. To change the name of a disk, use the ADDISK command with the -RENAME option.)

The advantages of disks listed above must be weighed against their disadvantages. Disks are expensive and require special handling and storage because they have lower tolerances for mechanical and environmental changes. Disks are also more difficult than tapes to transport from site to site.

Disk-to-tape Backups

A major reason for choosing tapes for backups is that they are much less expensive than disks, even though several tapes are required to hold the same amount of data as a single disk. Tapes are also easier to store and transport.

Disk-to-tape or tape-to-disk copies, however, are slower than disk-to-disk copies. The fastest way to copy a 300 megabyte disk to tape is by using the PHYSAV utility. Such a transfer takes about 45 minutes at 6250 bpi. However, the restoration of the backup to disk also takes about 45 minutes, which means a total of 1 1/2 hours for the backup and restoration procedures.

The three utilities for making disk-to-tape backups are BACKUP, MAGSAV, and PHYSAV. These utilities are discussed in the following paragraphs.

The BACKUP Utility: The BACKUP utility copies the data file by file. BACKUP keeps a catalog (that is, an online list) of the files that were backed up during each backup session. When you are performing a partial restoration, the catalog enables you to locate quickly a specific file and the tape on which it is stored.

BACKUP sets the Date/Time Backedup attribute on objects, but does not set the Rev. 20 Date/Time Accessed attribute.

If you use BACKUP, create a system ACL group named .BACKUP$ and include in it any user who is authorized to make backups. Users who do not belong to .BACKUP$ receive an "Insufficient Access Rights" error message when they attempt to use BACKUP.

The BACKUP_RESTORE utility restores data saved on BACKUP tapes.

The MAGSAV Utility: The MAGSAV utility copies data file by file. The MAGRST utility restores data from a MAGSAV tape.

Rev. 20 or greater versions of MAGRST and MAGSAV work with Rev. 19 versions as follows:

- Rev. 20 or greater MAGRST reads Rev. 19 MAGSAV tapes.

- Rev. 19 MAGRST reads Rev. 20 or greater MAGSAV tapes if the tapes were created with Rev. 20 or greater MAGSAV's -REV19 option.

- Rev. 20 or greater MAGSAV writes the Rev. 20 or greater new system boot on MAGSAV tapes only if the -REV19 option is not used.

The PHYSAV Utility: Unlike BACKUP and MAGSAV, the PHYSAV utility makes an exact copy of the disk contents as the contents appear on the disk. A PHYSAV tape cannot be used to restore a single file, because the file is spread over the tape as it was on the disk, and tapes cannot be used for random access to data. PHYSAV copies have to be restored to a disk before any access operation is possible.

A PHYSAV operation that transfers a full disk takes less time and uses fewer tapes than an equivalent BACKUP or MAGSAV transfer. However, because BACKUP and MAGSAV can restore data file by file, you can restore a single file quickly and efficiently using the BACKUP_RESTORE or MAGRST utility.

Full and Incremental Backups

A backup operation is either a full backup or an incremental backup.

A full backup copies the entire contents of the specified partition, MFD, UFD, or files, regardless of when they were created or altered.

An incremental backup copies only those files that have changed since the last backup copy was made. Incremental backups can be made using either BACKUP or MAGSAV (for disk-to-tape backups) or the COPY command (for disk-to-disk backups). Incremental backups are faster to make, because fewer records are copied. However, it is sometimes slower to restore a complete data base from them because each increment must be reloaded separately.

Incremental backups may supplement full backups. For example, incremental backups can be used when activity is low on the system as a whole (thus not requiring frequent full backups) but is high on a few directories or files. In this case, the backup schedule would consist of full backups done on the basis of the overall system activity, while incremental backups would keep the high-activity files up-to-date.

BACKUP GENERATIONS

It is usually a good idea to have a three-level backup in operation. A three-level backup consists of three generations of backups, with each generation kept in a different location. When a new backup is made, the generations are rotated, so that the oldest is deleted.

The latest (most recent) backup disk or tape should not be kept in exactly the same place as the originating data, but it should be easily and quickly accessible. This is the copy that restores data to the system in the form that requires the least updating.

The intermediate copy should be kept in a secure (and preferably fireproof) location, different from the location of the latest copy but possibly in the same building. This copy should be quickly accessible if both the current disk and the latest backup are destroyed, but the system and the disk and tape drives are still operational.

The oldest copy should be kept off site, preferably in a different building. The off-site copy is the copy that is least likely to be needed.

DATA ARCHIVES

Backups can also be used to create data archives. Data archives contain copies of inactive files that may be required at a future time. After an inactive file is archived, the file may be removed from the disk, thus freeing disk space for active use.

You may want to have your normal backups serve for archiving as well as for security against data loss. In this case, you are essentially archiving your entire data base and plan to keep copies for a relatively long time.

Alternatively, you can keep archived material separate from backed-up material. Under this scheme, archived copies are considered long-term storage (perhaps for an indefinite period of time), while backup copies are short-term storage, with the oldest disk or tape being reused as soon as two or three newer copies are made.

SCHEDULING BACKUPS

How often you perform backups depends on several factors, including how often the data in your system changes, how important it is that the data is up-to-date, and other details specific to the installation.

The first factor to consider is how much your system changes from week to week, from day to day, or even from hour to hour. All backups take time and use system resources. You have to decide what combination of data security and time/system use is best for your installation and resources.

If your system is highly changeable, you may need to back it up frequently, probably at least once a day. Remember that in the event of a disk crash, all the data entered since the last backup will have to be reentered to restore the system to its pre-crash state. The closer to the time of the crash that the backup copy was made, the less data will have to be reentered.

With a highly changeable system, you may find the incremental backup plan useful. Incremental backups can reduce the number of required full backups, thus also reducing the amount of system and operator time spent in processing backups.

If your system changes slowly, you may prefer to perform a full backup only once a week. If your backups are this widely spaced, it is a good idea to perform an incremental backup at least once between full backups.

The second factor to consider is the degree of protection you want for active data. Few backups may be needed on a system that gets data from off site (such as from cards, tapes, or a half-duplex network), processes it, and sends out the results. On such a system, data is rarely resident and the programs that handle it change little.

On the other hand, a system with many transactions but little processing (for example, a blood bank) needs frequent backups because it requires much data entry and many changes to data files.

The third factor to consider is system resources, which may include the physical plant (disk and tape drives) and the personnel. Four of the system resource considerations that influence the timing of backups are the following:

- The type of media used affects the time that a backup takes. Disk-to-disk backups are faster than disk-to-tape backups.
- Backups require that any disk used be accessible only to the person performing the backup during the time that the backup is being performed.
- The amount, type, and timing of the use that your system gets must be considered. If your system is very busy throughout the normal workday, you should schedule backups before or after normal working hours. If some of the disks are busy at certain times but idle at others, you should take this into account when scheduling backups.
- The amount of time that your operators have to perform backups. Operator time can be conserved (and the probability of error reduced) by running backups from a CPL program or a COMINPUT file.

Example of a Backup Strategy

The following example shows how a typical development system might be handled. The online storage devices consist of two 300-megabyte storage modules and one 80-megabyte storage module. Most of the system activity is concentrated on the 300-megabyte drives.

- All backups are full backups because all the data on the system must be absolutely current and quickly restorable.
- The first 300-megabyte disk is backed up to another disk on Monday, Wednesday, and Friday mornings before normal working hours.
- The second 300-megabyte disk is backed up to disk on Tuesday and Thursday mornings.
- The 80-megabyte drive is not backed up during the week because it is not as active.
- All three disks are backed up to tape every weekend. These tapes are all kept for two months.
- The first set of tapes created each month is kept for two years.

The degree of data protection given by this regimen is probably well in excess of that required by most installations. Because each installation has its own special requirements, the System Administrator is responsible for deciding which backup strategy to use for that particular installation.

15 Looking After Users

Users call on the System Administrator for help in many areas. Operators and Project Administrators may assist you with some user problems, but some responsibilities remain yours. Among the System Administrator's duties that concern users are the following:

- Adding new users to the system
- Helping users with some common problems
- Handling the problem of full disks

This chapter discusses these duties.

ADDING USERS TO THE SYSTEM

Before a user can log in to and use the system, you must supply the user with a set of system attributes and one or more sets of project attributes.

The system attributes must include a user ID and a password (possibly null). The system attributes may also include a default project and membership in a maximum of 16 systemwide ACL groups.

The minimal project attributes are the user ID (which is placed in the project data base) and an Initial Attach Point (also called the origin directory). The Initial Attach Point may be specified for the user, or it may be the project's default Initial Attach Point. In addition, project attributes may include membership in a maximum of 16 project-based ACL groups and command environment limits.

Use the EDIT_PROFILE utility, as explained in Chapter 12, USING EDIT_PROFILE, to define a user's system attributes and project attributes.

Origin Directories

It is not enough to define the user's Initial Attach Point with EDIT_PROFILE. The System Administrator or the Project Administrator must also ensure that the origin directory exists and that the user has appropriate access to the directory.

Users do not have to log in to top-level directories. A user's Initial Attach Point may be anywhere in the directory structure. Often, therefore, a new user's Project Administrator can create the user's origin directory. If, however, the user needs a top-level directory as an Initial Attach Point, the System Administrator may be the only person with sufficient rights to the MFD to create the directory.

HELPING USERS WITH PROBLEMS

Among the more common problems that users have are unsuccessful logins, inability to access directories, insufficient disk space, and errors with EPFs and segments.

Unsuccessful Logins

The action that you take when a user cannot log in depends on what message the user receives. Normally, the messages concern user IDs and/or passwords, origin directories, project IDs, or the Login Server.

Incorrect User Id or Password: Use the following procedure if the user receives the message "Invalid user id or password; please try again."

- If you have more than one computer at your site, find out which computer the user thinks he or she should be logging in to. Then use EDIT_PROFILE to ensure that the user ID given by the user is actually registered in that system's SAD.

- If the user ID is correct, determine whether the user's terminal is connected to the proper system. If not, the user must either use a different terminal or log in remotely.

- If the user ID is correct and the user is trying to log in to the right system, the password is probably incorrect. You cannot check the password because passwords are stored in an unreadable form. Assign the user a new password with EDIT_PROFILE. After the user logs in with the new password, he or she can retain the new password or change it with the CHANGE_PASSWORD command.

Unavailable Initial Attach Point: After a user issues the LOGIN command, the following error message indicates that the user could not be attached to the system:

```
Unable to attach to your initial UFD:
Not found. (nlogin)
Please contact System Administrator.
```

The problem may have one of four causes:

- The user's Initial Attach Point was not entered correctly into the user's project data base in the SAD.

- The origin directory itself does not exist because it has not been created, it has been deleted, or its name has been changed.

- The directory exists, but is on a remote partition.

- The user may not have appropriate access rights to the Initial Attach Point.

Check the user's project data base in the SAD with EDIT_PROFILE to find out the user's Initial Attach Point and then check the relevant partition to make sure the directory exists. Also check the ACL rights to the directory to ensure that the user has at least Use (U) rights.

If the partition is on a remote system, either create a directory on a local partition or assign the user a new Initial Attach Point. Alternatively, you can change the user's line to connect to the remote system and add the user ID to that system.

Incorrect Project ID: If the message is "Invalid project id", either the user misspelled the project name or the user is not a member of any project. (This can happen if a user is removed from one project before being added to another.) Check the user's project affiliation with EDIT_PROFILE's LIST_USER command. (Use the format "LIST_USER user-id -ALL".)

A user who had not been specifying a project ID at login may find that the system is now demanding a project ID. The cause is that the user's default login project has been deleted. Either assign the user a new default login project, or have the user specify a project ID at login.

Logins Blocked: If the Login Server stops, it sends the following message to all logged-out terminals:

```
Logins are blocked -- Login Server is logged out. (lsr)
```

If an internally detected error causes the Login Server to stop, the supervisor terminal displays an error message. Users who try to log in while the Login Server is down receive no messages at their terminals. At the supervisor terminal, type the START_LSR command to start the Login Server.

Login Server Logs Out Abnormally: If the Login Server logs out abnormally after it is started, a search rules problem may be indicated. Check that all entries in the SEARCH_RULES*>ENTRY$.SR file are on the command device, that all pathnames are correct, and that the ENTRY$.SR file contains no typographical errors. From the supervisor terminal, fix the search rules and start the Login Server with the START_LSR command.

Access Problems

Users may come to you because their programs are failing due to access problems. Access problems are signalled by messages such as those explained in the section below, Access Error Messages.

How you handle this situation may depend on how much time you have to fix it. Situations in which time is at a premium (for example, when an end-of-the-month accounting package cannot run) require different handling than less critical situations. Suggested strategies for both cases are discussed in the following two sections.

Time-critical Situations: If the message is "Insufficient access rights" and time is of the essence, set a priority ACL on the partition on which the directory exists, thus allowing the program to run. To set the priority ACL, use the SET_PRIORITY_ACCESS command described in Chapter 9, SETTING SYSTEM ACCESS.

Either set the priority ACL to allow the original user to run the program, or run the program yourself. After the program has finished, remove the priority ACL with the REMOVE_PRIORITY_ACCESS command.

Ordinary Situations: Before attempting remedial action, follow the steps below:

1. Find out exactly where the access problems are occurring and what protection is causing them. See the next section for error messages caused by access problems.

2. Check whether the user really should have the right to run these particular programs, or to access the data being denied.

3. When you have collected the facts, you can decide how to remedy the situation, so that the programs in question work correctly for those users who need them.

Access Error Messages: The four error messages below are caused by access problems. Following each error message is a possible course of action to alleviate the problem indicated by the error message.

- Bad password. dir-name (df_unit_)

The user attempted to attach to a passworded directory, named dir-name, with an incorrect or missing password. To solve this problem, you have three choices:

1. Give the user the password.

2. Remove the password, let the user complete the task, and then replace the password.

3. Remove the password and ACL the directory, giving the user the appropriate rights to the directory to accomplish the task.

- Insufficient access rights. obj-name (cmd-name)

The user attempted to access an ACL-protected object, named obj-name, to which the user has no rights. (Some commands do not print the object's name.) cmd-name is the PRIMOS command (such as ATTACH) or module (such as OPENR or std$cp) that returned the error. Give the user appropriate access rights to accomplish the needed work.

- No information. obj-name (cmd-name)

The user attempted to list information for an ACL-protected object, named obj-name, to which the user has no rights. cmd-name is the command (for example, LIST_ACCESS or LIST_QUOTA). Give the user appropriate access rights to accomplish the needed work.

- Top-level directory not found or inaccessible. dir-name (cmd-name)

The user attempted to access a directory, named dir-name, that was not available. cmd-name is the command (for example, ATTACH or LD). Some of the causes and solutions to this error message are as follows:

- The user does not have the appropriate ACL rights to the directory. Give the user appropriate access rights to accomplish the needed work.

- The partition on which the directory exists has not been added. Use the ADDISK command to add the disk to the system.

- The directory is on a remote partition, but the user cannot access it because of problems with the network. If your network has not been started, use the START_NET command to start it. If the remote system has shut down its network, call the System Administrator of that system to determine the problem. For information on networks, see the PRIMENET Guide or the Network Planning and Administration Guide.

- The user made a typographical error in typing the name of the directory.

- The directory does not exist.

Problems With Full Disks

When a disk is full, the user receives an error message when trying to write to the disk, as in the following example:

```
OK, COPY <DEPT4>JED>LOG.BOOK
The disk is full.  (cp$$fl)
ER,
```

The user is also pushed down a command level to allow him or her to delete files from the disk. The user can continue by issuing the START command. If, for some reason, the user cannot delete files, follow one of the solutions listed in the section below, PROBLEMS WITH CROWDED DISKS.

Quota Problems

On a system where directory quotas are in use, users with quota problems may require your intervention.

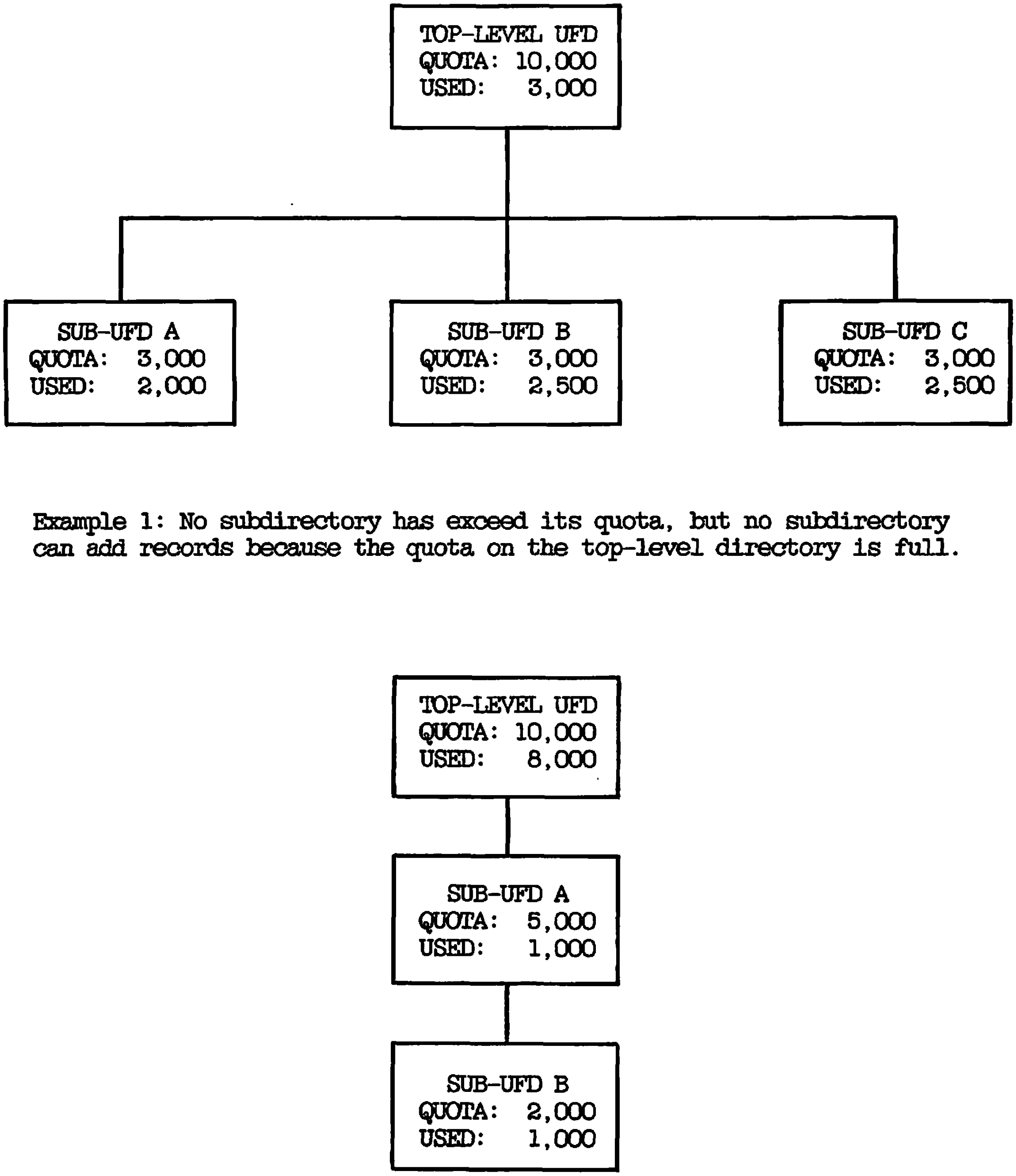
The most common case is the following: A user cannot write to a directory and receives the message "Maximum quota exceeded." The user issues the LIST_QUOTA command and discovers that there are several unused records left. The user comes to you to determine why the two messages conflict.

The probable cause is that the quota has been exceeded, not in the user's own directory, but in a higher parent directory. (Figure 15-1 shows two examples illustrating how this situation can occur.) A user with List rights to parent directories can trace the quota problem. If the user does not have the necessary access rights, you must do the checking.

If you find out that a particular directory is causing the problem, use one of the following solutions:

- Grant more space to that directory.
- Request that users of the parent directory, or of subordinate directories within that tree, clean out those directories. You may have to archive some files to make the cleanup possible.
- Adjust quotas on all or most top-level directories. See the section below, PROBLEMS WITH CROWDED DISKS.

If you determine that there is sufficient space in the directory where the user is having the problem and in all higher level directories in the tree, the user's program may be creating a temporary file that fills up the directory. In this case, the program must be modified so that it deletes the temporary file when the quota is exceeded.



Example 1: No subdirectory has exceed its quota, but no subdirectory can add records because the quota on the top-level directory is full.

Example 2: The blockage for SUB-UFD B occurs even higher in the tree. Users of SUB-UFD B must search up two levels to find it.

Two Examples of Unexpected Quota Errors
Figure 15-1

EPF Level Problems

Users with EPF level problems should release any unneeded levels (using the RELEASE_LEVEL command) before they contact you for assistance. The following paragraphs assume the user has done so.

You should assign each user at least 10 command levels and 5 invocations per command level. PRIMOS gives a user a new command level each time the user uses CONTROL-P or the BREAK key. (The RDY command can indicate the command level of a user.)

A new command level is also created after a runtime error in a program. This allows a programmer to suspend a program in order to issue a command that may affect the state of a program and then restart the program with the REENTER or START command.

Command levels are useful for debugging programs that incur runtime errors. The programmer can read the runtime stack with the DUMP_STACK command.

Users may have problems if the limits on the number of command levels and number of invocations per level are too low. If users complain that they frequently reach mini-command level, they may need a greater number of command levels to accomplish their work. Increase the number depending on the following:

- If system defaults are enabled, increase the system default numbers.

- If system defaults are disabled (in which case project and user defaults are enabled), increase the project defaults or the user's individual limits.

If you increase the user's number of command levels, or the number of live invocations of EPFs at a command level, the user must log in again for the new limits to take effect. If you increase the system defaults, you must cold start the system for the new numbers to take effect.

Users should consult the Programmer's Guide to BIND and EPFs for more information on solving EPF problems.

Static Segment Problems

You should assign each user at least 40 private static segments. If certain commands or utilities do not function, the user may not have enough static segments and the following message is displayed:

```
Error: condition "ILLEGAL_SEGNO$" raised at 4nnn/nnnn
(Referencing segno (ring)/offset)
```

If system defaults are enabled, use EDIT_PROFILE to increase the number of default static segments. If system defaults are disabled (project and user-based limits are enabled), increase the number of static segments for that user. If several users in the same project are getting the same message, the project limits or defaults may be too small. You, or the Project Administrator, can increase the number of static segments.

If you increase the user's number of static segments, the user must log in again for the new limit to take effect. If you increase the system default number of static segments, you must cold start the system for the new defaults to take effect.

Dynamic Segment Problems

It is recommended that you allocate at least 40 private dynamic segments to each user. If a user reports any of the following error messages, it is likely that the user does not enough dynamic segments allocated. These messages are further explained in the Programmer's Guide to BIND and EPFs.

```
Not enough segments. COMMAND_NAME (std$cp).

No space available from process class storage heap.

STORAGE raised in PROGRAM_NAME at nnnn
   (insufficient space for ALLOCATE)

ERROR raised in PROGRAM_NAME at nnnn
   (no on-unit for STORAGE)
```

The last two messages are likely to appear together, and may mean that the user is running a program that has not defined an on-unit.

If system defaults are enabled, increase the system default number of dynamic segments. If system defaults are disabled (project and user defaults are enabled), increase the project defaults or the user's individual limit for dynamic segments.

If you increase the user's number of dynamic segments, the user must log in again for the new limit to take effect. If you increase the system default number of dynamic segments, you must cold start the system to enable the new defaults.

The LIST_LIMITS command lists the number of private dynamic and static segments allocated to a user.

Note

To run some programs, such as SEG and DBG, a user needs more segments than the minimum allowable number of 16 dynamic segments and 8 static segments.

PROBLEMS WITH CROWDED DISKS

Your system may experience chronic problems with crowded disks. Depending on which directories or how many directories are crowded, and depending on how badly your users need space, consider one or more of the following suggestions:

- Get more disks at your installation.
- Instruct users to increase space by deleting outdated or obsolete files.
- Build a tape archive and put in it some of the outdated or obsolete files. You can instruct your users how to archive their files.
- Move user groups or directories from one partition into another, less crowded partition. This move requires changing the users' Initial Attach Points with EDIT_PROFILE.
- Compress UFD space by using the FIX_DISK utility. The FIX_DISK utility is fully explained in the Operator's Guide to File System Maintenance.
- Adjust the quotas on directories. (See the next section, Adjusting Quotas.)

Careful monitoring of the system allows you to warn users when disks begin to get full, so that users can delete old material before the disks are full. (See Chapter 17, SYSTEM MONITORING.) Users can also monitor the remaining space on their own partitions with the AVAIL command. However, if disk usage keeps increasing and you cannot use any of the last four suggestions listed above, you may have no choice but to add more disks to the system.

Adjusting Quotas

Following are some strategies for adjusting the quotas when your disk space becomes crowded:

- If you have employed an undercommitted quota strategy (as discussed in Chapter 4, DISKS AND TAPE DRIVES), increase the quota limit for the directories that most need extra space.

- Reset the quotas on the top-level directories across the board. You are thereby taking extra space away from a directory that may have ample space and giving it to a directory that is about to run out of space.

- Set the quota down to a limit below the level of records the user has already consumed. For example, if a user directory has a quota of 20,000 records and has already used up about 19,500 records, set the quota below 19,500 -- perhaps to 15,000. The purpose of this very strong measure is to force users to delete unneeded data and to become more efficient in their use of space. Users would repeatedly get the warning message "Maximum quota exceeded" until they deleted or moved enough data out of their directory to go below the new lower limit. (Use this strategy as a last resort.)

16
Adding and Modifying System Software

This chapter contains the following information on adding and modifying system software:

- Adding your commands to the command directory (CMDNC0)
- Using customer-defined file suffixes
- Changing defaults for compilers
- Adding files to the HELP data base

ADDING COMMANDS TO CMDNC0

You can add your commands to the command UFD CMDNC0. The commands must be either runfiles (that is, compiled and loaded programs) or CPL programs, but they cannot be segment directories.

Use the following procedure to add a command to CMDNC0:

1. Use ED or EMACS to create the source file.
2. If the source program is not a CPL program, create a runfile by compiling the source file with the appropriate compiler and loading the binary file with either BIND or LOAD.
3. 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 the runfile 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.

After you add your command to CMDNC0, you should create a HELP file on that command 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
```

Note

> CPL, EPF (.RUN), and V-mode .SAVE (loaded with SEG) programs cannot be run under PRIMOS II. To run R-mode programs suffixed with .SAVE under PRIMOS II, you must enter the suffix when issuing the command. When running under PRIMOS II, do not execute any programs that are intended to write on Rev. 20 disks, because PRIMOS II cannot write on these disks.

Adding CPL Programs

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

```
    COPY NEW_PROG.CPL CMDNC0>NEW_PROG.CPL
```

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. Wait a while and try again.

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

Adding EPF Programs

With EPFs created by BIND, you need not be concerned with whether a previously existing version of the 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.)

Note

> Binary files compiled by a Rev. 20.2 compiler cannot be loaded with a pre-20.2 Rev. of BIND or SEG.

Copy the EPF runfile into the UFD 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.RPn (where program is the name of the old version and n is a number from 0-9).

The user of the old version of the EPF is never aware of the change, and continues to execute the old version. Any new user who invokes the program gets the new version. You can delete the old version when it is no longer in use.

Following is an example of replacing an in-use EPF:

```
OK, COPY NEW.RUN CMDNC0>==
EPF file "CMDNC0>NEW.RUN" already exists, do you wish to replace it? YES
New version of EPF file CMDNC0>NEW.RUN now in place.
Old version of active EPF file now named CMDNC0>NEW.RP0.
OK,
```

If there is already a file named NEW.RP0, the old version is named 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 CMDNC0, COPY queries you, as shown in the following example:

```
OK, COPY NEW.RUN CMDNC0>==
EPF file "CMDNC0>NEW.RUN" already exists, do you wish to replace it? YES
ok to delete EPF file CMDNC0>NEW.RP0?  YES
New version of EPF file CMDNC0>NEW.RUN now in place.
Old version of active EPF file now named CMDNC0>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 CMDNC0>==
EPF file "CMDNC0>NEW.RUN" already exists, do you wish to replace it? YES
EPF replace files are all in use.
Unable to complete file copy. (copy)
ER!
```

If you or your users frequently modify versions of runfiles in CMDNC0, you should delete unused versions from time to time to save space. Use the DELETE command as follows:

```
DELETE CMDNC0>@@.RP(0 1 2 3 4 5 6 7 8 9) -NO_VERIFY -REPORT
```

Adding R-mode Programs

R-mode programs can be written only for the FTN compiler and the PMA assembler. To install an R-mode program into CMDNC0, 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 CMDNC0 as follows:

```
OK, COPY FARLEY.SAVE CMDNC0>FARLEY.SAVE
```

A "File in use." or "File open on delete." error message indicates that the current copy of the program in CMDNC0 is being used. Wait a while 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

The PRIMOS command processor interprets the following wildcard options, treewalking options, and special characters on the command line. For details on these features, see the PRIMOS Commands Reference Guide.

Wildcard Option	Abbreviation
-ACCESS_CATEGORY	-ACAT
-ACCESS_AFTER date	-ACA
-ACCESS_BEFORE date	-ACB
-AFTER date	-AF
-BACKEDUP_AFTER date	-BKA
-BACKEDUP_BEFORE date	-BKB
-BEFORE date	-BF
-CREATED_AFTER date	-CRB
-CREATED_BEFORE date	-CRA
-DIRECTORY	-DIR
-FILE	
-MODIFIED_AFTER date	-MDA
-MODIFIED_BEFORE date	-MDB
-NO_VERIFY	-NVFY
-SEGMENT_DIRECTORY	-SEGDIR
-VERIFY	-VFY

Treewalking Option	Abbreviation
-BOTTOM_UP	-BOTUP
-WALK_FROM level	-WLKFM
-WALK_TO level	-WLKTO

Special Character	Meaning
~	Syntax suppressor
;	Command separator
% %	Global variables
[ ]	Command functions
( )	Iteration
@ @@ + ^	Treewalking
@ @@ + ^	Wildcarding
= == ^= ^== +	Name generation

You can create commands that prevent PRIMOS from processing one or more command line features. Because the invocation of such commands appears 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. 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 Prime User's Guide for details on filenames. Following are examples of filenames with user file suffixes:

```
SALES.UDATA
UPDATE.UTRANS
PROBLEMS.UXB
```

CHANGING COMPILER DEFAULTS

Compilers (also called translators) process source programs into object code, which can be loaded into an EPF or a runtime memory image by one of Prime's linkers or loaders. The compilers also perform other related operations such as error message printing and concordance generation. These operations are governed by command line options.

The procedure for changing compiler option defaults varies according to the type of compiler that is involved.

In Prime's newer compilers, defaults are changed within a data file that is accessed by the execution of a driver program. The commands for invoking the newer compilers are CBL (for COBOL 74), F77 (for FORTRAN 77), PASCAL (for Pascal), PL1G (for PL/I G), and VRPG (for RPG II V-mode).

In Prime's older compilers (FTN and PMA), defaults are changed by supplying new octal values for the A register and B register. These registers are used in .SAVE memory image files that are accessed by commands such as PM, SAVE, and RESTORE. COBOL 66 (the COBOL command) and RPG II (the RPG command) use only the A register for default options. The octal values set the default bits on (1) or off (0) in the registers.

Note

When invoking FTN or PMA, the user can supply on the command line either the desired options or the new octal values for the A and B registers. Supplying options is the recommended method.

Changing Defaults With Driver Programs

The F77 (FORTRAN 77), PASCAL, PL1G (PL/I G), and VRPG (RPG II V-mode) compilers have their option defaults set with driver programs. These driver programs are supplied on the Master Disk in the UFDs F77>TOOLS, PASCAL>TOOLS, PL1G>TOOLS, and VRPG>TOOLS. The programs are copied to the top-level directory TOOLS when the compiler is installed.

The default information is stored in data files supplied in UFD SYSOVL. These data files are not text files, and must be modified only by the driver programs. The System Administrator can move the driver programs to other directories, but the data files must remain in SYSOVL.

Set the protection on the driver programs and data files as follows:

- Set ACLs for the directory in which the driver programs are stored so that the driver programs are set to $REST:NONE. This setting prevents unauthorized execution to change defaults.

- Set protection on the data files to allow you Read and Write access, and to allow users only Read access.

Changing the Defaults: To change the defaults of a newer compiler, use the following sequence of commands:

```
    ATTACH directory
    RESUME driver-program new-options
```

directory is the UFD in which the driver programs are resident. (The directory is supplied as TOOLS.)

driver-program is the name of the compiler driver program: F77DF for FORTRAN 77, PASCALDF for Pascal, PL1GDF for PL/I G, and RPGDF for RPG II V-mode.

new-options are the new default options for the compiler.

Defaults for the Newer Compilers

The following sections list the Prime-supplied compiler option defaults that are set by driver programs for FORTRAN 77, Pascal, PL/I G, and RPG II V-Mode. For more information on these compilers, see the *FORTRAN 77 Reference Guide*, the *Pascal Reference Guide*, the *PL/I Subset G Reference Guide*, the *RPG II V-Mode Compiler Reference Guide*, and their updates.

FORTRAN 77 (F77) Defaults: The Prime-supplied F77 compiler defaults are as follows:

```
    -ALLOW_PRECONNECTION
    -BINARY
    -DYNM
    -INTL
    -LOGL
    -NO_BIG
    -NO_DCLVAR
    -NO_DEBUG
    -NO-DO1
    -NO_ERRLIST
    -NO_EXPLIST
    -NO_FRN
    -NO_FTN_ENTRY
    -NO_LISTING
    -NO_MAP
    -NO_OFFSET
    -NO_OVERFLOW
    -NO_PBECB
    -NO_PRODUCTION
    -NO_RANGE
    -NO_STANDARD
    -NO_STATISTICS
    -NO_STORE_OWNER_FIELD
    -NO_XREF
    -OPTIMIZE 2
    -SILENT
    -TIME
    -UPCASE
    -64V
```

Pascal (PASCAL) Defaults: The Prime-supplied PASCAL compiler defaults are as follows:

```
-ALLOW_PRECONNECTION
-BINARY
-CONFORMANT_ARRAYS
-ERRTTY
-FRN
-NO_BIG
-NO_DEBUG
-NO_ERRLIST
-NO_EXPLIST
-NO_EXTERNAL
-NO_LISTING
-NO_OFFSET
-NO_OVERFLOW
-NO_OVERFLOW
-NO_PRODUCTION
-NO_RANGE
-NO_STATISTICS
-NO_STATISTICS
-NO_XREF
-OPTIMIZE 2
-SILENT 1
-STORE_OWNER_FIELD
-TIME
-UPCASE
-64V
```

PL/I G (PL1G) Defaults: The Prime-supplied PL1G compiler defaults are as follows:

```
-ALLOW_PRECONNECTION
-BINARY
-COPY
-ERRTTY
-NO_BIG
-NO_DEBUG
-NO_ERRLIST
-NO_EXPLIST
-NO_FRN
-NO_LISTING
-NO_NESTING
-NO_OFFSET
-NO_OVERFLOW
-NO_PRODUCTION
-NO_RANGE
-NO_STATISTICS
-NO_XREF
-OPTIMIZE 2
-SILENT 1
-STORE_OWNER_FIELD
-TIME
-UPCASE
-64V
```

RPG II V-mode (VRPG) Defaults: The Prime-supplied RPG II V-Mode compiler defaults are as follows:

```
-ALLOW_PRECONNECTION
-BINARY
-ERRTTY
-NO_BANNER
-NO_DEBUG
-NO_ERRLIST
-NO_EXPLIST
-NO_LISTING
-NO_OFFSET
-NO_PRODUCTION
-NO_RANGE
-NO_SEQCHK
-NO_STATISTICS
-NO_XREF
-OPTIMIZE 2
-SILENT 1
-STATUS
-STORE_OWNER_FIELD
-TIME
-UPCASE
-64V
```

Changing Defaults With Register Values

In FTN (FORTRAN IV) and PMA, the defaults are changed by setting new octal values for the A and B registers. COBOL (COBOL 66) and RPG (RPG II R-mode) use only the A register.

To change the defaults, use the following sequence of commands:

```
ATTACH CMDNC0
RESTORE compiler.SAVE
SAVE compiler.SAVE [3/A-register] [4/B-register]
```

compiler is the compiler utility: FTN, COBOL, RPG, or PMA. *A-register* and *B-register* are the new octal values of the A and B registers. If either value is omitted, the current value is unchanged.

FORTRAN IV (FTN) Defaults: The Prime-supplied FORTRAN IV compiler defaults are as follows:

A register: '1707

- -BINARY YES
- -ERRTTY
- -INPUT
- -LISTING NO
- -NOTRACE
- -32R

B register: 0

- -FP
- -INTS
- -NOBIG
- -NODEBUG
- -NOXREF
- -SAVE
- -STDOPT

See the *FORTRAN Reference Guide* for values of the A and B registers.

COBOL 66 (COBOL) Defaults: The Prime-supplied COBOL compiler defaults are as follows:

A register: '2777

- -BINARY YES
- -INPUT
- -LISTING YES
- -NOEXPLIST
- -64V

See the *COBOL Reference Guide* for values of the A register. The B register is not used.

RPG II (RPG) Defaults: The Prime-supplied RPG II compiler defaults are as follows:

A register: '3777

```
-BINARY YES
-ERRTTY
-INPUT
-LISTING YES
-NOBANNER
-NOEXPLIST
-NOOBDATA
-NOSEQCHK
-STATUS
-XREF
```

See the RPG II Programmer's Guide for values of the A register. The B register is not used.

Note

> The RPG II compiler interprets an A-register setting of 0 as '3777 (the default setting).

Assembler (PMA) Defaults: The Prime-supplied Assembler defaults are as follows:

A register: '0777

```
-BINARY YES
-ERRLIST
-INPUT
-LISTING YES
-NOEXPLIST
-XREFL
```

See the Assembly Language Programmer's Guide for values of the A register. The B register is not used.

LINKERS AND LOADERS

For complete details on Prime's linkers and loaders, see the Programmer's Guide to BIND and EPFs and the SEG and LOAD Reference Guide.

BIND

Default library: PFTNLB

SEG (V-Mode)

Stack size: '6000 half words
Default library: SPLLIB, PFTNLB and IFTNLB (FORTRAN libraries)

LOAD (R-Mode)

Memory location: '122770 to '144000
Default library: FTNLIB (FORTRAN library)
Mode: D32R
Sector Zero Base Area:
  Base start at location '200
  Base range '600 half words
COMMON: Top at location '077777

ADDING HELP FILES

System Administrators can add HELP files (on any subject) to the HELP data base supplied by Prime. After these site-created HELP files are installed, the PRIMOS HELP command can display them.

The HELP Data Base

The HELP data base 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 HELP* directory. The HELP* directory also contains the following two text files:

- HELP_INDEX.HELP
- HELP_SEARCH_LIST

The HELP_INDEX.HELP file is a list of all the HELP files in the HELP* directory. If you add your own HELP file to the directory, edit HELP_INDEX.HELP to include the name of the new file. The HELP_INDEX.HELP file is displayed at one of two times: when a user enters the HELP command without an argument; or when the HELP command cannot find an appropriate HELP file and the user answers YES to the command's prompt, "Can't find x; do you want a list?".

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.

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.)

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.
2. File it to HELP*>command.HELP (where command is the name of the new command).
3. Edit the HELP_INDEX.HELP file to include the new command.

Protecting the HELP Data Base

When your system is first installed, HELP* is accessible to anyone. You should limit Write 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 data base and LUR access to $REST.

17
System Monitoring

The System Administrator must always be aware of whether the system is running normally or malfunctioning. This chapter discusses the three methods to use to keep track of system events:

- The system logbook
- Event loggers
- System-monitoring commands

The system logbook should contain information about external events that may cause problems, such as power failures. The event log files and the use of system-monitoring commands disclose such conditions as the status of the system hardware and the network.

With a series of logs and reports from regular system monitoring samples, you may foresee system problems and take measures to forestall them. If a problem does develop, you can backtrack through the logs and COMOUTPUT files of monitoring sessions to look for use or event patterns that may disclose a cause. The logs and monitor output files are particularly useful for finding causes of intermittent, unpredictable problems.

THE SYSTEM LOGBOOK

Every system should have a handwritten logbook in which operators record information about system status and operation. Prime does not define the format of the logbook or what type of information goes into it. Rather, it is up to you, as the System Administrator, to determine the makeup of the logbook. (Below are some suggestions to help you with this decision.) You must also ensure that all operators know what to enter into the logbook, and how to enter the information.

The Purpose of the System Logbook

The primary purpose of a system logbook is to allow backtracking if a problem occurs. Many apparently sudden problems give unrecognized warnings before they occur. If these warnings are entered into the logbook, they may provide clues to your system support personnel as to the nature of the problem. The problem can then be tracked down and solved faster than if nothing were written down.

Format of the Logbook

To help you determine the format of the system logbook, some suggested standards and procedures that have been used successfully by operators of Prime systems are listed below.

- Logbooks are numbered and dated with the dates of the first and final entries.
- Logbooks are bound, not loose-leaf. Loose-leaf pages are easily detached and lost, particularly if they are used often.
- Logbooks should stay flat when open, thus making it easier to write in them.
- The page size should be large enough to allow printouts and listings to be pasted in. The exact page size, however, is not important.
- Each entry is labeled with its date and time. Labeling provides an historical record, which helps you to reconstruct a system crash or other unexpected event, and allows you to correlate the entry with external events, such as power failures.
- Each entry is signed or initialed by the person making the entry. You or your Customer Service Representative then know whom to ask for further information about a specific event.
- All entries are made in indelible ink, not in pencil or erasable ink. An incorrect entry should be neatly crossed out and initialed by the person deleting it.

Contents of the Logbook

The exact contents of your system logbook are up to you, because you are the only person who knows the exact needs of your system. However, the following lists recommend some types of information and events that should be recorded in a system logbook.

Hardware Information:

- The physical system configuration, including the model number and serial number of every piece of equipment. You may want to list each type of equipment with others of the same type (that is, list all disk drives in a group, all terminals in a group, and so on).
- Changes to the original configuration, including any addition, deletion, alteration, or substitution of any piece of equipment.
- Any change in the operating status of any component, such as component failure and unexpected occurrences (even if not fatal).

Environmental Information:

- All abnormal temperature or humidity conditions. If possible, include the date, time, and duration of the conditions.
- Other unusual conditions, such as smoke, dust, or chemical spillage. If possible, note the date, time, and duration of the conditions.
- Any unauthorized access to the computer room, with the date and time that the unauthorized access was discovered and the name of the person who discovered it.
- Any equipment loss or damage, with the date, time, and cause, if known.
- Any unauthorized use of the computer, including attempts at remote login.
- All other unusual or unexpected events or results.
- All actions taken to correct an environmental problem.

Software Information:

- A listing of the system startup file (PRIMOS.COMI or C_PRMO). If you have several alternate configurations, listings of all the alternate startup command files.

- A listing of the system configuration file (usually CONFIG).
- A listing of the system default entrypoint search file (SYSTEM>ENTRY$.SR).
- A list of the segment numbers of all memory segments allocated as shared. Note that these numbers are octal representations.
- A list of the contents of the command directory CMDNC0, and the library directories, LIB and LIBRARIES*.
- A listing of the memory loadmaps RING0.MAP and RING3.MAP for the PRIMOS version used by the system.
- A listing of the network configuration as produced by CONFIG_NET.
- A listing of the environment files for the printers.
- A listing of the configurations of the batch queues.
- All additions, deletions, alterations, or replacements to any of the above.

Operations Information:

- Every system startup. Special conditions (such as the omission of the BATCH or FTS system startup) should also be noted.
- Use of FIX_DISK, with the name and physical device number of the partition being processed and the result of the operation.
- All disk formattings, with the name of the partitions created, and which disk drive was used.
- Information about backups performed, including the names of the partitions copied, the date of the copy, the type of copy (for example, incremental, total, COPY_DISK, MAGSAV), the type of media used (disk or tape), the media statistics (such as tape speed and density), and the number of recoverable and nonrecoverable errors (if any).
- The names of files or directories restored to the system, with the date, time, and reason for the restoration.
- The name of any file or directory that is archived (removed from the active disks to storage for possible later use), with information about the type of media to which it is archived, the date and time of the archiving operation, and the place in which the archive is kept.
- The date, time, and place of storage of any event logger printout. (The event loggers are described below.)

- The addition, deletion, alteration, or replacement of any commands in CMDNCO or libraries in LIB or LIBRARIES*, with the date, time, and reason for the action.

- All changes to the default entrypoint search list, SYSTEM>ENTRY$.SR.

- All system shutdowns, including information about their extent (partial or complete), date and time, and cause (such as environmental factors, plant shutdown, configuration change, or system update).

- All top-level directories that are added to or deleted from the system.

- All users who were added or deleted from the system.

- All passwords that were changed or revealed to users.

- All telephone requests for passwords or for telephone numbers.

Information on Halts:

- The status of the system when it halted. The status is usually provided by the halt message, which includes the segment number at which the system halted (this gives a reason for the halt), and the contents of the status words (DSWSTAT, DSWRMA, DSWPB, and, for some systems, DSWPARITY).

- The contents of the X, A, and B registers, if the system halted on an uncorrected parity error.

- Whether a crash dump to tape was performed (a tape dump is recommended if more than one halt has occurred recently).

- Whether a warm start or a cold start was performed after the halt. For more information on whether to perform a warm start or a cold start, see Chapter 13, EQUIPMENT AND ENVIRONMENT, and your CPU handbook.

- After the restart, the behavior of the machine should be noted at various times. For instance, did the system function correctly immediately after the restart? Did it continue to function correctly after a half hour?

Procedures for handling halts (including information on tape dumps) are described in detail in the appropriate CPU handbook for your machine. The information listed above is the minimum that should be recorded in the system logbook during or after a halt.

EVENT LOGGERS

Prime supplies automatic event loggers for the system and the network. An event logger is a software utility that automatically records information about significant system or network events. Events that are logged include cold starts, machine checks, disk errors, and network link problems. The output from these loggers can be useful in tracking problems, especially those problems that develop or worsen over time.

PRIMOS includes two event loggers: one that controls system event logging and one that controls network event logging.

Event logging, and the contents of the events logs, are discussed in detail in the Operator's Guide to System Monitoring (for system event logging) and in the PRIMENET Guide (for network event logging). The following discussion provides a brief overview and some suggestions on the use of the event logs.

System Event Logging

System event logging may be enabled or disabled in two ways:

- The LOGREC configuration directive enables or disables logging when the system is cold started. To enable event logging, specify a 0 to the directive in the system configuration file (that is, LOGREC 0). To disable event logging, specify a negative value for the directive (for example, LOGREC '177777). You must disable event logging if you want to prevent logging messages onto a write-protected disk.

- After the system is running, the EVENT_LOG command controls event logging. EVENT_LOG or EVENT_LOG -ON enables logging, while EVENT_LOG -OFF disables it.

System event logs are kept in the directory LOGREC*. Because the files are in binary form, they cannot be edited with ED or EMACS, displayed with SLIST, or printed with SPOOL. (To display or print system event logs, see the later section, Printing Log Files.)

System event logs are named LOG.mm/dd/yy, where mm/dd/yy stands for the date (month, day, year) on which event logging was last enabled. Thus, if you start your system on Monday, 08/11/86, with event logging enabled, the log filename bears Monday's date (LOG.08/11/86). If you shut down the system and then restart it on Wednesday, 08/13/86, a new file is begun, with Wednesday's date as part of its filename (LOG.08/13/86). However, if you perform a second cold start on Wednesday (or if you turn event logging off and on again with the EVENT_LOG command), the LOG.08/13/86 file is reopened, and new entries are appended to it. You can have a maximum of one log file per day.

Note

The system event log file is opened or closed only by the EVENT_LOG command. An attempt to close the file with the CLOSE command produces the error message "Insufficient access rights."

Network Event Logging

The procedures for network event logging parallel those for system event logging. Network event logging is controlled in two ways:

- By the NETREC configuration directive. NETREC 0 enables logging and NETREC '177777 disables it.
- By the EVENT_LOG command. EVENT_LOG -NET -ON enables logging and EVENT_LOG -NET -OFF disables it.

Network event log files are kept as binary files in the directory PRIMENET*. The files are named NET_LOG.mm/dd/yy, where mm/dd/yy represents the date of either the last cold start or the last use of the command EVENT_LOG -NET -ON. As with system event logs, you cannot edit network event log files.

Access Rights to Log Directories

System event logging is performed by User 1 (SYSTEM). Network event logging is performed by NETMAN. SYSTEM needs at least ALURW rights to the LOGREC* directory, and NETMAN needs ALL rights to the PRIMENET* directory. System Administrators and operators should have at least DALURW rights to LOGREC* and PRIMENET* so they can write to the logging files (with the PRINT_SYSLOG or PRINT_NETLOG commands). $REST can be given LUR rights or, for a more restricted system, NONE rights. SLAVE$ must have at least LUR rights.

Error Handling

All errors that arise during event logging are reported every five minutes to the supervisor terminal. The errors fall into three categories:

- Quota exceeded
- Disk full
- Disk shutdown

Quota Exceeded: To prevent log files from consuming too much disk space, set a quota (using the SET_QUOTA command) on LOGREC* and/or PRIMENET*. When the quota on either directory is exceeded, one of the following messages is printed on the supervisor terminal every five minutes until more space is created for the files.

- Exceeding quota on LOGREC*. System event logging not taking place. (LOGEV2)
- Exceeding quota on PRIMENET*. Network event logging not taking place. (NETEV2)

Alternatively, you can halt the messages by using the EVENT_LOG command to disable event logging. For more information on how to create more disk space, see the section below, Controlling the Size of Event Log Files.

Disk Full: Event logging no longer takes place if the partition on which logging is taking place becomes full (either because the log files are too large or because other directories are using up all the space). In this case, one (or both) of the following error messages is printed every five minutes at the supervisor terminal, until either event logging is disabled or space is made available on the disk:

- Disk full. System event logging not taking place. (LOGEV2)
- Disk full. Network event logging not taking place. (NETEV2)

Disk Shutdown: Shutting down logical disk 0 (the command device) while event logging is enabled closes the event log files. If the disk is then added back to the system, event logging must be reenabled with the EVENT_LOG command. If you do not reenable event logging, one of the following sets of error messages is printed, once only, at the supervisor terminal:

- Disk has been shut down. System event logging not taking place. (LOGEV2)
  Disk has been shut down... Please reenable system event logging. (LOGEV2)
- Disk has been shut down. Network event logging not taking place. (NETEV2)
  Disk has been shut down. Please reenable network event logging. (NETEV2)

Both types of event logging are then disabled until you issue the EVENT_LOG commands to reenable them.

Other Events: All other errors cause one of the following messages to be printed at the supervisor terminal every five minutes until you correct the error or disable event logging:

- System event logging not taking place.
- Network event logging not taking place.

Controlling the Size of Event Log Files

If allowed to grow and multiply indefinitely, event log files consume large amounts of disk space. You can do the following two things to control the amount of space consumed:

- Put a quota on the directories LOGREC* and/or PRIMENET*.
- Print out and delete old log files at regular intervals.

Using Quotas on Log Directories

Use the SET_QUOTA command to set a quota on an event-logging directory, as you would on any other directory. For information on SET_QUOTA, see the Prime User's Guide or the PRIMOS Commands Reference Guide.

Event logging stops when the directory's quota is filled. To warn you of this fact, an "Exceeding quota" error message (as shown in the preceding pages) appears every five minutes at the supervisor terminal, until you do one of the following:

- Increase the directory's quota with the SET_QUOTA command
- Print old log files with the PRINT_SYSLOG and PRINT_NETLOG commands (as explained below) and then delete them with the DELETE command
- Delete empty log files with the DELETE command

If the system is running normally, it is not disastrous to exceed the quota and lose some events from the log files. However, more events are logged when the system is experiencing problems. Thus, the times when you most need log files are also the times most likely to cause quota problems. In general, you should keep plenty of space in your directory by printing and deleting log files at regular intervals.

Printing Log Files

To print log files, use the PRINT_SYSLOG and PRINT_NETLOG commands with the -SPOOL option. The PRINT_SYSLOG command prints the system event log. The PRINT_NETLOG command prints the network event log. Each command converts a log file from its binary format to a readable format.

Below are brief descriptions of these commands. The PRINT_SYSLOG command is explained in full in the Operator's Guide to System Monitoring. PRINT_NETLOG is detailed in the PRIMENET Guide.

Printing the Latest Log File: To print the most recent log file, use the -SPOOL option for the command, as follows:

```
PRINT_SYSLOG -SPOOL

PRINT_NETLOG -SPOOL
```

This format writes the output to a file whose default name is LOGLST (for the system event log file) or NETLST (for the network event logging file). The LOGLST or NETLST output file is then spooled. If the output file already exists, you are first asked if it should be deleted.

If you use both the -SPOOL and -DELETE options, the LOGLST or NETLST output file is deleted after it has been submitted to the spool queue:

```
PRINT_SYSLOG -SPOOL -DELETE
```

The -DELETE option deletes only the LOGLST or NETLST output file. To delete an event log file from the directory, you must use the DELETE command.

An existing LOGLST or NETLST can be spooled with the SPOOL command.

Printing Old Log Files: To print earlier log files, use the -INPUT and -SPOOL options (with or without the -DELETE option). If you are not attached to LOGREC* or PRIMENET*, use a pathname for the input file.

For example, the following command spools the system event log file from August 11, 1986, and then deletes the LOGLST output file:

```
PRINT_SYSLOG -INPUT LOGREC*>LOG.08/11/86 -SPOOL -DELETE
```

In general, you should print log files at least once a week to keep your hard copy records up-to-date. After you print old log files, you can delete them to keep the directories clean.

Storing Log Files

General rules for storing hard-copy log files are as follows:

- For quick reference, keep the most recent files with the system logbook.
- Store older files in a safe archive (such as a file cabinet). The amount of time files are kept varies from installation to installation. You decide what makes the most sense for your system.

When deciding how often to print and delete log files and how long to keep the printed output, consider the following issues:

- How much disk space the on-line files can occupy
- How much storage space you have for the printed files
- How your system has been operating over the last few weeks or months

These issues vary widely from site to site, and only a person with first-hand knowledge of the installation can make such decisions. If problems arise, consult your Customer Support Center.

SYSTEM-MONITORING COMMANDS

Use the following PRIMOS commands to monitor your system:

- STATUS monitors higher level system events, such as information about users, the status of devices and the network, the current version of PRIMOS, and the amount of physical memory.
- USAGE monitors the status and performance of the CPU and other system internals. The STATUS and USAGE commands are complementary because both monitor system usage.
- LIST_QUOTA displays the number of records used in a directory tree.
- AVAIL reports the usage and availability of disk space.
- MONITOR_NET monitors the events on the network.
- FIND_RING_BREAK locates breaks in the ring.

These commands are discussed briefly in the sections below.

You can also use the LOOK, SIZE, and LD commands for monitoring. The LOOK command is discussed in Appendix D of this guide, OBSOLETE AND RARELY USED COMMANDS AND DIRECTIVES. The SIZE command displays the size (in 2048-byte records) of files and the number of entries in directories. The -SIZE and -SORT_SIZE options of the LD command display the size of the contents of a directory.

All these commands send their output to the screen, not as hard copy (unless your supervisor terminal is a hard-copy terminal). To obtain a hard copy of the output, open a COMOUTPUT file before you begin the monitoring sequences. After you close the file (using the command COMOUTPUT -END), print it with the SPOOL command.

The STATUS Command

The STATUS command monitors higher level system events. When invoked without an argument at the supervisor terminal, STATUS displays the following information:

- The version of PRIMOS your system is running
- The size of main memory
- Your username (SYSTEM) and the network node name of your system.
- Your open files
- All currently assigned magnetic tape drives, their physical and logical device numbers, and the user IDs and numbers of the assignees
- All currently started partitions, including their names, logical device numbers, physical device numbers (for local partitions only), and node names
- Semaphore values
- All configured network nodes and their status (UP or DOWN)
- The physical device numbers of the command device (COMDEV), the primary paging device (PAGDEV), and, if present, the alternate paging device (ALTDEV)
- All logged-in users, including their user IDs, user numbers, terminal line numbers, in-use partitions, and assigned devices

The STATUS COMM format of the command displays information on communications controllers.

The STATUS command thus allows you to find out such things as the following:

- Whether users are still on the system (necessary when you are about to shut down the system)
- Whether anyone is using a partition that is about to be backed up or reformatted
- Which are the currently started partitions
- Which tape drives are in use and by whom
- Which user is using which terminal
- What remote users, phantoms, and slave processes are using the system
- How the communication controllers are configured

Before beginning any system operation that may affect users, operators should use the STATUS command to determine the state of the system. The operator can warn users so that they can take the action necessary to ensure that their work is not harmed. Such system operations include shutting down the system for preventive maintenance, formatting a partition with MAKE, and performing a backup.

If you are not monitoring system status from the supervisor terminal, keep in mind that the operation of the STATUS command is slightly different when invoked from a user terminal. At the supervisor terminal, the STATUS default is ALL (that is, typing STATUS is the same as typing STATUS ALL). At a user terminal, typing STATUS without an argument omits information about other users and about assigned tape drives. Furthermore, some information (such as the amount of main memory and the physical device numbers for COMDEV, PAGDEV, and ALTDEV) is displayed only at the supervisor terminal.

For further details on STATUS, see the Operator's Guide to System Monitoring.

The USAGE Command

The USAGE command is a system metering tool that monitors events internal to the system at the hardware level. Such events include the total CPU time used since the system was started, the number of input/output operations occurring per second through the sampling time, CPU and I/O usage statistics for each user, and information on disk I/O operations.

Any user at any terminal can invoke USAGE. A sequence of one or more USAGE samples can be generated automatically or manually.

USAGE is an especially useful tool for the System Administrator because it determines the degree to which individual users and processes are using system resources and thus affecting system performance. The operation, options, and output of USAGE are documented in the *Operator's Guide to System Monitoring*.

The LIST_QUOTA Command

The LIST_QUOTA command lists the maximum quota on a directory, the total number of records used by the entire directory tree, and the number of records used by the particular directory. LIST_QUOTA is useful for metering disk usage.

To use LIST_QUOTA, you must have List access to the target and parent directories, and Use access to any higher level directories. However, this restriction can be overridden through the use of a priority ACL. Priority ACLs are discussed in Chapter 9, SETTING SYSTEM ACCESS.

The format of the LIST_QUOTA command is as follows:

```
    LIST_QUOTA  [pathname]  [-BRIEF]
```

pathname is the directory for which you want quota information. If you do not specify a name, information on the current directory is listed. LQ is the abbreviation for the command.

The following example shows quota information for subdirectory STATS, which is contained in a higher level directory called TEST.

```
    OK, LIST_QUOTA TEST>STATS

    Maximum records allowed on "TEST>STATS" = 500.
    Total records used = 425.
    Records used in this directory = 28.
    OK,
```

The output shows that the maximum number of records that can be used by the subdirectory STATS and all of its subdirectories is 500. Of this quota, 425 records have already been used, leaving the STATS and its subdirectories 75 records before the directory tree runs out of space. STATS has used 28 records for files out of the total 425 records used. The other 397 records are used by the subdirectories of STATS.

If no quota has been set on the directory, a message to that effect appears in place of the maximum number of records. The total number of records used by the directory and by the entire subtree is displayed.

The -BRIEF option (abbreviated -BR) displays the quota data in tabular form. This option does not display a message if the directory is a non-quota directory, but instead gives the maximum number of records as zero.

The LIST_QUOTA command is also explained in the PRIMOS Commands Reference Guide and in the Prime User's Guide.

The AVAIL Command

The AVAIL command monitors the utilization of disk space. For any specified partition, the AVAIL command displays the following information:

- The size of the partition
- The number of records still available for use
- The percentage of records used

The format for the command is as follows:

```
AVAIL [disk-id] [-NORM]
```

disk-id is one of the following: the name of a partition (including a remote partition); -LDEV n, where n is the logical device number of a partition; or * (see below for this format). If you do not specify disk-id, information is displayed for the partition to which you are attached.

The default output gives the number in terms of physical records. A physical record contains 2048 bytes. The term physical record comes from the fact that this is the size of each slot for a user-data record on the disk. In fact, each record on the disk requires some identification data as well, so the total size of each disk record is actually 2080 bytes.

The -NORM option displays the information in normalized records. Normalized records contain 880 bytes.

Access Rights: If you want any form of the AVAIL command to be accessible to users, you must grant them Read rights to the DSKRAT file on each disk, and List and Use rights to the MFD. If your disks are password-protected, one of the passwords on the MFD must be XXXXXX. If you do not want users to use AVAIL, the simplest method is to deny them rights to the AVAIL command itself, in CMDNC0.

The AVAIL * Format: The AVAIL * format is particularly useful because it displays data, in tabular form, for all partitions on the system.

The AVAIL * command works by reading information from a file. To make AVAIL * work, therefore, the System Administrator (or the operator) must take the following steps:

1. Use ED or EMACS to create a file named DISCS within the directory SYSTEM.

2. Give users List and Read access to the DISCS file.

3. Place information on each of the system's partitions within the DISCS file. If the system is networked, you may also include information on remote partitions. The DISCS file must contain one or more columns of text. The first column contains the names of all partitions to be listed, one per line. The other columns may contain any other information on each of the partitions. Such information may include (in any order) the logical device number, the physical device number (for local partitions), the name of the system to which a remote partition is physically connected, or the fact that a partition is write-protected.

4. Update the file as needed, to keep it current with your system's actual usage of disks.

When a DISCS file exists in the directory SYSTEM, issuing the AVAIL * command displays the file's contents. For each partition that is actually running, it also gives information on space usage. For other partitions, a message appears indicating that the partition is not running.

Following is an example of a DISCS file, and of the output from an AVAIL * command that uses the file:

```
OK, SLIST SYSTEM>DISCS
CLOUDS   0    460
FOREST   1  12060
OCEAN    2  52061
HILLS    3  22062
PLAINS   4  61463
OK, AVAIL *

Volume      Total       Free    %       Comments
  ID        recs        recs   Full
-------------------------------------------------
CLOUDS      14814        376   97.5      0     460
FOREST      59256        909   98.5      1   12060
OCEAN       66663      31017   53.5      2   52061
HILLS       59256      32765   44.7      3   22062
PLAINS      51849      30316   41.5      4   61463
```

Exceeding Disk Space: If the AVAIL command shows that your system is frequently running out of disk space, you may need more or larger disks. If you have several partitions, and only one or two of them are regularly more than 95 percent full, you should consider increasing the size of these partitions. However, if you do this, make sure that you are not making any other partitions too small. You must also take care not to reformat all or part of a disk that is in use, because the data on it will be lost. (For further information on disks, see Chapter 4, DISKS AND TAPE DRIVES.)

The MONITOR_NET and FIND_RING_BREAK Commands

If you have PRIMENET on your system, use the MONITOR_NET and FIND_RING_BREAK commands to monitor and maintain your network.

MONITOR_NET displays information about RINGNET™ , synchronous lines, and virtual circuits for your system. The information includes performance, traffic, and status data. You can select any one of three monitors (RING, SYNCHRONOUS LINE, or VIRTUAL CIRCUIT) or the Main Menu. The ability to run MONITOR_NET as a phantom process is especially useful.

FIND_RING_BREAK locates hard breaks in RINGNET (that is, breaks that cause complete interruption of the signals on the ring). Although FIND_RING_BREAK cannot detect a malfunctioning RINGNET Repeater, it can isolate the break to between two active nodes. You should run FIND_RING_BREAK in the following situations:

- The RING monitor of MONITOR_NET indicates a break.
- The "RING MAY BE DOWN" error message appears on the supervisor terminal.
- The STATUS NETWORK command indicates "down" nodes.

For detailed information on the operation and options of MONITOR_NET and FIND_RING_BREAK, see the PRIMENET Guide.

APPENDIXES

A
External Login and Logout Programs

Since Rev. 19, you can write separate external programs to monitor and control logins and logouts. (Prior to Rev. 19, one program had to serve both needs.)

At login time, the PRIMOS operating system looks for a program in CMDNC0 named LOGIN and runs it if it exists. At logout time, PRIMOS looks first for a program named LOGOUT in CMDNC0. If the LOGOUT program does not exist, PRIMOS then looks for LOGIN. (Suffixes are not allowed on either name.)

These programs cannot be EPFs; they must be static-mode programs created by SEG or LOAD.

GUIDELINES FOR LOGIN AND LOGOUT PROGRAMS

The external login program may regulate the use of the operating system in addition to the LOGIN facility provided by PRIMOS. The program may access confidential system information, such as valid user IDs, project IDs, and per-user accounting information. Therefore, take precautions when writing an external login program to prevent inadvertent or malicious misuse of the program.

The following are factors you should consider for external login and logout programs:

- External login and logout programs must reside in the directory CMDNC0.

- The external login program must be named LOGIN. The external logout program must be named LOGOUT. No suffix is allowed.

- Both programs must be static-mode, created with SEG or LOAD.

- Access to the programs should be strictly controlled.

- All files that are opened by both programs must be closed before the program completes execution.

- CONTROL-P (used to break out of programs) is inhibited when the external login program begins execution. Breaks are disabled because if a user breaks in the middle of the external login program, files are left open and the user is logged in without having gone through all validation checks. The external login program must reenable breaks when it completes execution.

- The subroutine PRJID$ allows external login programs to record project IDs given at login time. This subroutine is not available in R mode. The calling sequence is as follows:

```
DCL PRJID$ ENTRY (CHAR (32) VAR);

CALL PRJID$ (PROJECT_ID_NAME)
```

- If user input from the terminal is required during the external LOGIN process, the external login program must set a timeout condition to avoid an indefinite wait for a user response. Use the TTY$IN subroutine in conjunction with the T1IN or C1IN subroutines.

- If a password or other validation code is required for a login, give the user a finite number of chances to enter the correct password. If the correct password is not entered in this number of trials, the external login program should log out the user.

- The external login program may forcibly log out a user if the user does not pass the validation process.

- Design the external login program to meet the needs of your individual site. Because these needs may vary over time, design the program to be easily understood and modified.

SAMPLE EXTERNAL LOGIN AND LOGOUT PROGRAMS

Below are sample external login and logout programs in Pascal. The external login program is as follows:

```
{ External login program                                          }
{ Initial coding:  Bob Eastwood 2/8/86                            }
{ This program intercepts all incoming logins and asks all users  }
{ coming in through the network for an extra password.            }

PROGRAM MAIN;

CONST
   turn_echo_on = 8192;        { Terminal in full duplex, XOFF enabled }
   turn_echo_off = -8192;      { Terminal in half duplex, XOFF enabled }
   convert_to_upcase = 1;      { Convert password to upper case        }
   correct_passwd = 'XXXXXX';  { PASSWD must be 6 chars                }
   from_remote_user = 3;       { Value for any user coming thru net    }
   len = 6;                    { Length of passwd in chars             }

TYPE
   char_array = ARRAY [1..8] OF CHAR;
   user_array = STRING [56];

VAR
   code : integer;             { return code from logo$$       }
   this_user_is : integer;     { user login status stored here }
   password : char_array;      { user entered password here    }
   user_duplex : integer;      { user's current characteristics }
   dummy : integer;
   userid_array : user_array;

FUNCTION duplx$  (A:integer); INTEGER; EXTERN;
PROCEDURE utype$ (VAR this_user_is:integer); EXTERN;
PROCEDURE case$a (A:integer; VAR passwd:char_array; C:integer); EXTERN;
PROCEDURE logo$$ (A,B,C,D:integer; E:longinteger; VAR code:integer);
  EXTERN;
PROCEDURE timdat (VAR A:user_array; B:integer); EXTERN;

BEGIN
   utype$ (this_user_is);       { How did user log in? }
   timdat (userid_array, 28);   { What is the user ID? }
IF this_user_is = from_remote_user THEN  { Is this a PRIMENET login? }
   BEGIN
      writeln;                                    { Yes, remote login }
      user_duplex = duplx$ (-1);
      dummy = duplx$ (turn_echo_off);             { Turn off echo      }
      reset (INPUT, '-INTERACTIVE');              { Allow erase char   }
      write ('Please enter remote password: ');   { Prompt for passwd  }
      readln (password);                          { Read passwd        }
      dummy = duplx$ (user_duplex);               { Echo as before     }
      case$a (convert_to_upcase, password, len);  { Convert to UPCASE  }
      IF (password <> correct_passwd) THEN        { Correct passwd?    }
```

```
         BEGIN                                     { Wrong passwd.      }
            writeln;                               { Tell user          }
            writeln('Sorry, invalid password...');
            logo$$(0,0,0,0,0,code);                { and log out.       }
         END;
   END;                                            { end if passwd OK   }
   writeln;
END.                                               { end of ext. LOGIN }
```

The following is a sample LOGOUT program:

```
{ External Logout program                                          }
{ Initial coding:  Bob Eastwood 2/8/86                             }
{ This program must be installed in order for the external login   }
{ program to work properly.  It says goodbye to the user.          }

PROGRAM MAIN;

BEGIN
   writeln;
   writeln('Goodbye.');
   writeln;
END.
```

SAMPLE COMINPUT PROGRAM

The following cominput program compiles and loads the sample external login program shown above. It also copies the external login program into CMDNC0 under the name LOGIN. This final step is necessary if PRIMOS is to run the external login program whenever a user logs in.

```
    /*  extlog.comi
    /*  compile and load sample external login program
    /*  then copy to cmdnc0>login (name must not have a suffix)
    pascal extlog -64v
    seg -load
    split
    mix
    s/lo extlog 0 4000 4000
    d/li vapplb
    d/li paslib
    d/li
    map
    save
    return
    share
    ex
    quit
    copy ex4000 cmdnc0>login -nq
    co -end
```

B
PRIMOS Cold Start Messages

This appendix contains the error and program messages generated by the PRIMOS preloader and the PRIMOS initialization sequence. The name of the module generating the error is often printed in parentheses at the end of the error message. For example, the message "NRUSR INVALID (BINIT)" was produced by the module BINIT.FTN.

Starting at Rev. 20, PRIMOS was enhanced to recover from a number of configuration errors, particularly those specifying nonexistent devices and incorrect physical device numbers for the command and paging partitions.

If a configuration error causes cold start to terminate, manually boot PRIMOS (with the '100000 bit set in the BOOT option word) and use the non-shared Editor (NSED) to correct the erroneous (or missing) directive. See Chapter 10, CONFIGURATION DIRECTIVES, for details on bringing up PRIMOS without a configuration file.

For information on network initialization error messages produced by the START_NET command, see the PRIMENET Guide.

MESSAGES

- <primos error> Allocating user 1 unit tables. (BINIT)

A file system unit table could not be allocated for user 1. Contact your Customer Support Center.

- ALTDEV n conflicts with PAGDEV.

The ALTDEV and the PAGDEV can not be the same partition or refer to overlapping areas of the disk. ALTDEV is ignored.

- ALTDEV n does not point to the beginning of a valid file system partition.

The physical device number of the partition does not point to an area with a valid file system header. The partition may be a non-split paging partition which, if used previously with pre-Rev. 20 PRIMOS, has had the file system header paged over. After this message is printed, the operator is asked: "Are you SURE you want to page on ALTDEV n?" If the operator answers YES, the partition is split and used for the alternate paging partition. If the operator answers NO, the partition is left as is and ALTDEV is ignored. To eliminate this message at the next cold start, use MAKE (with the -NO_INIT and -BADSPOT_LEVEL 0 options) on the partition.

WARNING

The query is a check to prevent paging over valuable file system partitions. Before answering YES, make certain that the partition is really intended for paging.

- ALTDEV n is an old partition which is not supported.

Certain types of partitions, particularly those with 440 words per disk record, are not supported after Rev. 15. The ALTDEV directive is ignored. Convert your disk.

- ALTDEV n is not a split disk and conflicts with COMDEV.

COMDEV and ALTDEV cannot be the same unless they are part of a split partition. ALTDEV is ignored and cold start continues.

- ALTDEV n is not a valid pdev.

The physical device number (n) supplied for the alternate paging partition is not valid. This error often occurs because of an invalid device type. The device type must be '60 (pdev = nnnn6n), not '20 (floppy disk).

- ALTDEV n, partition <x>, has not previously been used for paging.

The non-split partition has never been used for paging or has been newly formatted with MAKE. This warning prevents the operator from unintentionally paging over valuable data on file system partitions. The operator is asked: "Are you SURE you want to page on ALTDEV n?". If the operator answers YES, the partition is split and becomes the alternate paging partition. (This message is not repeated unless the partition is reformatted with MAKE.) If the operator answers NO, the partition is not used for paging and you have no alternate paging partition.

- BAD AMLCLK PARAMETER (CINIT)

The AMLCLK directive specifies a baudrate value less than '35 (29 decimal) or greater than '45400 (19200 decimal).

- BAD AMLIBL PARAMETER (CINIT)

The DMC input buffer size is too small. Specify either a value of 0 or a value greater than '20.

- BAD AMLTIM PARAMETER (CINIT)

The AMLTIM directive specifies a value for disctime or gracetime that is less than the value of ticks.

- BAD CONFIG DIRECTIVE - IGNORED. directive (CINIT)

The indicated directive is unrecognized and is ignored.

- Bad disctime n(dec.) specified for AMLTIM; n(dec.) will be used. (CINIT)

The value specified for disctime was not 0 and was less than ticks. Another value has been substituted for the one that you specified.

- BAD DMQ AMLC CONFIGURATION (CINIT)

A DMQ buffer size in an AMLBUF directive is not equal to a power of 2.

- BAD FILUNT PARAMETER (CINIT)

The FILUNT directive specifies an invalid value for the first parameter (which must be 0) or for max-unit.

- Bad gracetime n(dec.) specified for AMLTIM; n(dec.) will be used. (CINIT)

The value specified for <u>gracetime</u> was not 0 and was less than <u>ticks</u>. Another value has been substituted for the one that you specified.

- Bad ICS directive: CARDS (CINIT)

The ICS CARDS directive does not specify both the device address and configuration, or specifies an invalid device address.

- Bad ICS directive: INPQSZ (CINIT)

The ICS INPQSZ directive specifies an invalid value.

- BAD ICS DIRECTIVE: INTRPT (CINIT)

The ICS INTRPT directive specifies an interrupt rate of less than '12 (10 decimal) or greater than '144 (100 decimal). Cold start continues and the interrupt rate is set to either '12 (if you specified a value less than '12) or '144 (if you specified a value greater than '144).

- Bad ICS directive: JUMPER (CINIT)

The ICS JUMPER directive specifies fewer than three speeds or an invalid speed.

- BAD LINE # IN ASRBUF COMMAND (CINIT)

An ASRBUF directive specifies a line number other than the correct value 0.

- BAD LOTLIM PARAMETER (CINIT)

The LOTLIM directive specifies fewer than 2 or more than the time allowed by LOUTQM.

- BAD LOUTQM PARAMETER (CINIT)

The LOUTQM directive specifies 0 minutes of inactivity before logging out a user. Increase this amount.

- BAD NVMFS PARAMETER (CINIT)

The NVMFS directive specifies a value larger than '400 (256 decimal) segments.

- BAD PRATIO PARAMETER (CINIT)

The PRATIO directive specifies a value greater than '12 (10 decimal) or less than 0.

- BAD PREPAG PARAMETER (CINIT)

The PREPAG directive specifies a value representing more than the number of pages of memory available for paging.

- BAD PROTOCOL SPECIFICATION IN SMLC CNTRLR DIRECTIVE: p (CINIT)

The protocol argument to the SMLC CNTRLR directive specifies p, which is an invalid protocol. The controller is disabled. See Chapter 10 of this guide for valid protocols.

- BAD RECORD ID, ASKED: m FOUND: n

The expected record address does not match the record address of the record read. Check that you entered a correct physical device number. If the device is corrupted, boot from another device or use another disk pack or reel of tape.

- BAD RWLOCK PARAMETER (CINIT)

The RWLOCK directive specifies a value other than the correct values 0, 1, or 3.

- BAD SMLC CONTROLLER MAPPING COMMAND (CINIT)

An SMLC controller mapping directive specifies an invalid or missing controller number. The correct numbers are 0 and 1.

- BAD SMLC DATASET PROCEDURE: n (CINIT)

An SMLC DSC directive specifies an invalid or missing data set procedure of n. Correct values are 1, 2, and 3.

- BAD SMLC DATASET STRAPPING ORDER: n (CINIT)

An SMLC DSC directive specifies an invalid or missing data set strapping order of n. The data set strapping order must be a value consisting of any combination of the octal values '10, '2, and '1.

- BAD SMLC LINE MAPPING COMMAND (CINIT)

An SMLC line mapping directive specifies an invalid or missing physical line number. The physical line number must be 0, 1, 2, or 3.

- BAD SMLC PARAMETER (CINIT)

The SMLC directive is invalid or does not specify the required parameters.

- BAD SMLC RECEIVER ON/OFF CONTROL: n (CINIT)

An SMLC DSC directive specifies an invalid or missing receiver on/off control value of n. Correct values are 0 and 1.

- Bad token, ignore it.

A bad parameter exists in the system configuration file (CONFIG) or the system startup file (PRIMOS.COMI or C_PRMO). The error is ignored and cold start continues.

- Bad value 'n specified for AMLCLK; directive ignored. (CINIT)

The value specified was less than '35 (29 decimal) or greater than '45400 (19200 decimal).

- Bad value 'm specified for AMLIBL; default 'n will be used. (CINIT)

The value specified was less than '16 (14 decimal) and not 0. The default buffer size of 60 (48 decimal) will be used.

- Bad value 'm specified for LOTLIM; default 'n will be used. (CINIT)

The value specified was less than or equal to 0. The default value of three minutes will be used.

- Bad value 'm specified for LOUTQM; default 'n will be used. (CINIT)

The value specified was less than or equal to 0. The default value '1750 (1000 decimal minutes, which is 16 hours and 40 minutes) will be used.

- Bad value 'm specified for NVMFS; default 'n will be used. (CINIT)

The value specified was less than 0 or greater than system limit, which is '2000 (1024 decimal). The default value of '144 (100 decimal) will be used.

- Bad value n(dec.) specified for PRATIO; default n(dec.) will be used. (CINIT)

The value specified was less than 0 or greater than '12 (10 decimal). The default value of '5 will be used.

- Bad value 'm specified for PREPAG; default 'n will be used. (CINIT)

The value specified for the number of pages was less than 0. The default value of 3 will be used.

- Bad value 'm specified for RWLOCK; default 'n will be used. (CINIT)

The value specified was not 0, 1, 3 or 5. The default value of 1 will be used.

- nK BYTES MEMORY DETERMINED BAD AND MAPPED OUT

The indicated amount of memory is bad or not present and will not be used by PRIMOS. If you have missing memory, check that the MAXPAG directive specifies the total amount of memory you would have if missing memory were included, rather than the total amount of existing memory.

- Cannot add boot device nnnnn.

The boot disk device is not a valid file system device. Boot from another disk or from tape.

- Cannot read DSKRAT of ALTDEV nnnnn.

A disk read error occurred while trying to read the record that contains the DSKRAT on ALTDEV. Cold start continues and ALTDEV is ignored. Run, from tape, either FIX_DISK (with the -FIX and -INTERACTIVE options) or MAKE on the alternate paging partition.

- Cannot read DSKRAT of COMDEV nnnnn.

A disk read error occurred while trying to read the record that contains the DSKRAT on COMDEV. Cold start continues and the operator is prompted for another physical device number for COMDEV. Run, from tape, either FIX_DISK (with the -FIX and -INTERACTIVE options) or MAKE on the command partition.

- Cannot read DSKRAT of PAGDEV nnnnn.

A disk read error occurred while trying to read the record that contains the DSKRAT on PAGDEV. Cold start continues and the operator is prompted for another physical device number for PAGDEV. Run, from tape, either FIX_DISK (with the -FIX and -INTERACTIVE options) or MAKE on the primary paging partition.

- <primos error> Can't attach to CMDNC0 (BINIT)

An error occurred while attaching to CMDNC0 on the command partition, or while establishing it as the Initial Attach Point for user 1. Run, from tape, either FIX_DISK (with the -FIX and -INTERACTIVE options) or MAKE on the command partition. If the problem persists, contact your Customer Support Center.

- Can't attach to the SAD: Not found. (nlogin)

The SAD, which contains entries used to initialize the profile of the supervisor terminal, could not be found. System defaults will be used to initialize the profile of the supervisor terminal.

- <primos error> Can't set priority ACL. (BINIT)

A priority ACL could not be set for user 1. Contact your Customer Support Center.

- <primos error> Can't start slave. (NPXON)

Remote File Access (RFA) is not enabled for this system because slave processes cannot be started.

- <primos error> Can't start system event logging. (BINIT)

System event logging cannot take place because of the indicated error. System startup proceeds normally. Fix the problem and use the command EVENT_LOG -ON at the supervisor terminal. If you cannot fix the problem, contact your Customer Support Center.

- Coldstarting PRIMOS, Please wait...

PRIMOS is cold starting.

- COMDEV n does not point to the beginning of a valid file system partition.

The physical device number of the partition does not point to an area with a valid file system header. The operator is queried for another physical device number for COMDEV.

- COMDEV n is an old partition which is not supported.

Certain types of partitions, particularly those with 440 words per disk record, are not supported after Rev. 15. The operator is prompted for another physical device number for COMDEV. Convert your disk.

- COMDEV n is not a valid pdev.

The physical device number (n) supplied for the command partition is not valid. This error often occurs because of an invalid device type. The device type must be '60 (pdev = nnnn6n), not '20 (floppy disk).

- <primos error> config-file (CINIT)

After initialization, PRIMOS is unable to open config-file for reading to complete system configuration. Check that the filename specified in the CONFIG -DATA command (in the system startup file PRIMOS.COMI or C_PRMO) identifies the proper system configuration file. Check also that the configuration file exists in CMDNC0.

- Controller codes: nnnn, nnnn
  Error while loading device oo (PCCBS)
  Controller returned p words of (hex) status: qqqq qqqq qqqq qqqq
  ICS device oo: boot failed (BTPCC).
  ICS cold start initialization failure.

The ICS2 failed to pass its self-verify test. This hardware problem could be caused either by the controller itself or by the LACs. nnnn is the error code generated by the controller, oo is the octal device address, p is the number of words returned (up to four words may be returned), and qqqq is the word status. Contact your Customer Support Center.

- DEFAULT TREENAME NOT FOUND

The default PRIMOS runfile was not found. (The default pathname is stored in MFD>BOOT_RUN_FILE_TREENAME. If this file is not found, PRIRUN>PRIMOS.SAVE is assumed.) If booting from disk, you probably either set the '4000 option word or entered a carriage return at the RUNFILE TREENAME= prompt. If booting from tape, you either did not save the file correctly or the file is not on the tape. In either case, the operator is queried again for a filename. Specify another filename, such as PRIMOS.SAVE or MFD>PRIRUN>PRIMOS.SAVE.

- DISK ERROR, STATUS: n

A disk I/O error occurred. The status word(s) are displayed. Make sure that the device exists, that it is started, and that it is online. If booting from disk, check that you have used the correct drive and controller addresses.

- EMPTY FILE

The file to be booted contains fewer than eleven words and is therefore considered empty and invalid.

- Error accessing CONFIG file.

An error was encountered while trying to read the configuration file. Cold start continues and the operator is prompted for the COMDEV, PAGDEV, and NTUSR parameters.

- ERROR: BADSPX

The BADSPT file on the primary or alternate paging partition could not be opened. Use FIX_DISK to repair the partition. If the condition recurs, use MAKE to recreate the partition with a new BADSPT file.

- Error: ICS cold start initialization failure (COMINI)

The ICS communication controller was not initialized. Three possible causes are a nonexistent DOWN_LINE_LOAD* directory, missing downline load files in the DOWN_LINE_LOAD* directory, and ICS initialization errors. Each cause produces a halt. To correct the problem, locate and install the proper ICS downline load files. To bring up PRIMOS to install these files, reboot PRIMOS with the '100000 switch (bit 1) set in the BOOT command option word. This option word skips initialization of communication devices, thus allowing the system to successfully cold start so you can install the proper files from the system console.

- Error in badspot file format. (BADSP$)

The BADSPT file on the primary or alternate paging partition could not be read. Use FIX_DISK to repair the partition. If the condition recurs, use MAKE to recreate the partition with a new BADSPT file.

- Error pre-paging Primos files.

An unrecoverable disk read or write error occurred while reading the various PRnnnn files that comprise PRIMOS. The cold start halts. Restore the proper file. Reboot from disk or boot from tape.

- Error pre-paging PRIMOS record. (TPIOS)

The preloader is unable to use the primary or alternate paging partition (PAGDEV or ALTDEV). If you have specified ALTDEV in the system configuration file, comment out the directive (place /* in front of it) and attempt to restart the system. If the problem recurs, change the PAGDEV directive to specify the alternate paging partition, and attempt another restart. While running PRIMOS, run MAKE on the other paging partition to map out badspots.

If you use only one paging partition (PAGDEV), run standalone MAKE to rebuild the badspot file for the paging partition.

- Error reading PRnnnn.

A file system error was encountered while reading the indicated PRIMOS PRnnnn file. Restore the file from a backup tape and run FIX_DISK. In the meantime, boot PRIMOS from tape.

- Error while loading device oo (PCCBS)
  ICS2 controller has not responded to self test.
  Controller returned p words of (hex) status: qqqq qqqq qqqq qqqq
  ICS device oo: boot failed (BTPCC).
  ICS cold start initialization failure.

The ICS2 failed its self-verify test. Someone has probably renamed or tampered with the downline load files. oo is the device address of the controller, p is the number of words returned, and qqqq is the word status. Restore the DLL directory from backup tapes and try again. If this fails or if you have no backup tapes, contact your Customer Support Center.

- Error writing DSKRAT on ALTDEV n.

A write error occurred while attempting to update the DSKRAT. Check that the disk drive is not write-protected. ALTDEV is ignored.

- Error writing DSKRAT on PAGDEV n.

A write error occurred while attempting to update the DSKRAT. Check that the disk drive is not write-protected. The operator is prompted for another physical device number for COMDEV.

- Excessive value 'm specified for PREPAG; maximum 'n will be used. (CINIT)

The value specified was greater than the system's limit. The number of memory pages cannot exceed the number of memory pages available for paging.

- EXPECTED FILE MARK NOT FOUND.

The tape file mark was not found when booting PRIMOS from magnetic tape. Cold start halts. Try another tape.

- FILE NOT FOUND

The boot runfile was not found. Check that you included the .SAVE suffix. If you are booting from tape and the tape was made by saving the MFD, you must include MFD> in the pathname (for example, MFD>DOS>DOS.SAVE). The complete pathname of the file to be booted must be entered exactly as it was saved on the tape.

- First command MUST be CONFIG.

The first command in the system startup file (PRIMOS.COMI or C_PRMO) was not the CONFIG command. Cold start continues and the operator is queried for the COMDEV, PAGDEV, and NTUSR parameters.

- ICS cold start configuration failure.

An ICS2 is not operating properly or a LAC card is present where not expected (as defined by the ICS CARDS directive).

- ICS cold start initialization failure.

An ICS2 is not operating properly or a LAC card is present where not expected (as defined by the ICS CARDS directive).

- Illegal ALTDEV; ALTDEV ignored.

The device specified as the alternate paging partition is not a valid physical device number or conflicts with the command partition (COMDEV) or primary paging partition (PAGDEV). ALTDEV is ignored.

- Illegal COMDEV.

An invalid physical device number was given for the command partition.

- Illegal paging device size for ALTDEV n ignored.

The records argument of the ALTDEV directive has a value of zero or a negative number. The argument is ignored and the entire available space for paging on ALTDEV is used for paging.

- Illegal paging device size for PAGDEV n ignored.

The records argument of the PAGDEV directive has a value of zero or a negative number. The argument is ignored and the entire available space for paging on PAGDEV is used for paging.

- ILLEGAL TREENAME

The pathname of the boot runfile contains an invalid character or begins with a partition name. Possible errors include using a left angle-bracket (<) in the pathname (partition names are not allowed), beginning or ending the pathname with one or more right angle-brackets (>), and beginning a filename or a UFD name with a number. The invalid character is pointed out.

- Inconsistent cold start configuration for ICS2 device dd:

The message is followed by one or both of the following lines:

```
an async line card has been found where not expected in slot ss.

slot ss is empty where an async line card was expected.
```

At cold start, the ICS2 controller found that the actual Line Adapter Card (LAC) configuration differs from that specified in the ICS CARDS directive. Cold start continues as if the discrepancy does not exist. Contact your Customer Support Center.

- Inconsistent configuration for ICS2 device dd:

The message is followed by one or more of the following lines:

an async line card has been inserted into slot ss.

the async line card in slot ss has been removed or is now inoperable.

the async line card in slot ss is now inoperable.

At warm start, the ICS2 controller found that the actual Line Adapter Card (LAC) configuration differs from that specified in the ICS CARDS directive. Warm start continues as if the discrepancy does not exist. Contact your Customer Support Center.

- Initializing profile data for the supervisor from the SAD.

The SAD is being scanned for the entry under SYSTEM, the user name of the supervisor terminal. If the entry exists, data are read to initialize the profile of the supervisor terminal.

- INPUT BUFFERS TOO LARGE (AMINIT)

The DMC buffer size is too large. Reconfigure within the permissible range using the AMLIBL directive.

- Insufficient paging records available to page Primos on PAGDEV n.

The primary paging partition does not have enough records for paging purposes. The operator is prompted for another physical device number for PAGDEV. Increase the size of the primary paging partition.

- INVALID R-VEC: SA: m, EA: n, PC: o

The file to be booted contains an invalid RVEC parameter. For an RVEC to be considered valid, the 16 bit, unsigned starting address (SA) must be lower than the ending address (EA) and the starting PC value must lie somewhere between the SA and the EA. Possible causes of this error are attempts at booting DOS>*DOS64 or booting an EPF (.RUN file) instead of a .SAVE file.

- LOGREC config directive no longer sets a quota on the system event logging file.
  Please use 'SET_QUOTA'. (CINIT)

Prior to Rev. 19, the system event logging file could be given a quota size with the LOGREC configuration directive. Since Rev. 19, the size of this file is controlled by setting a quota on its directory with the SET_QUOTA command. To enable system event logging, set the value for LOGREC to 0 or omit the directive from the configuration file.

- LOGREC* ufd does not exist. Can't start system event logging. (BINIT)

The LOGREC* UFD does not exist on the command partition. System startup proceeds normally. Create the UFD LOGREG*, set access on it as described in Chapter 8, and issue the command EVENT_LOG -ON at the supervisor terminal.

- MACHINE CHECK: DSWPB: m DSWRMA: n DSWSTAT: o

A machine check was generated. The diagnostic status words (DSW) give the location of the check, the instruction that caused the check, and the type of the check. Issue a RUN command to step past this error. If the RUN command does not solve the problem, reboot with the '1000 switch (bit 7) set in the BOOT command option word so that the boot does not run in machine-check mode.

- MEMORY TEST MISMATCH, LOCATION: n

The memory location displayed generated an error during memory test. Contact your Customer Service Representative.

- Missing COMDEV.

The configuration file does not contain the COMDEV directive. The operator is queried for the physical device number for COMDEV.

- Missing "GO" command.

The configuration file does not include the required GO directive at the end. The operator is queried for the COMDEV, PAGDEV, and NTUSR parameters.

- Missing NTUSR.

The configuration file does not contain the NTUSR directive. The operator is queried for the value of NTUSR.

- Missing PAGDEV.

The configuration file does not contain the PAGDEV directive. The operator is queried for the physical device number for PAGDEV.

- MT# OR OCTAL ONLY

The operator made an error while entering the physical device number (pdev). The number can be an octal number only. Reenter a correct physical device number or tape device number (MTn).

- NETREC config directive no longer sets a quota on the network event logging file.
  Please use 'SET_QUOTA'. (CINIT)

Prior to Rev. 19, the network event logging file could be given a quota size using the NETREC configuration directive. Since Rev. 19, the size of this file is controlled by setting a quota on its directory with the SET_QUOTA command. To enable network event logging, set the value for NETREC to 0 or omit the directive from the configuration file.

- No available CMDNC0 on n.

The command directory CMDNC0, which must exist on the boot device, could not be found. The operator is queried for another physical device number for COMDEV.

- No room. AMLBUF (TFLADJ)

The total size of the terminal I/O buffers (AMLBUF directives and ASRBUF directive) exceeds 256 kilobytes (2 segments).

- NON-ZERO T$MT RETURN CODE: n.

An unexpected error was returned by the tape driver T$MT. Correct the error and reboot.

- NOT A MAGSAV FORMAT TAPE

You attempted to boot from a non-MAGSAV tape. Rev. 20 BOOT is present only on tapes written with Rev. 20 MAGSAV without the -REV19 option.

- NOT A MINIMAL CONFIGURATION, NEED AT LEAST 512K BYTES.

A missing memory module check was generated. The minimum physical memory configuration required for Rev. 20 is 512 kilobytes. Because the boot does not reference outside this range, a missing memory check is logically due to insufficient memory or faulty hardware. This error causes a halt, but the operator can enter RUN at the VCP CP> prompt and continue to boot a T&M.

Note

> Whether there are 512 kilobytes or more of physical memory, the first 512 kilobytes must be contiguous starting at location 0. Improperly installed memory boards of different types may cause a hole in the address space that can produce this message.

- NOT ENOUGH PHYSICAL MEMORY FOR n (decimal) USERS (BINIT)

The system does not have enough physical memory for the specified configuration.

- Not found. Can't attach to PRIMENET* (NPXON)

Remote File Access (RFA) is not enabled for this system because the PRIMENET* directory cannot be accessed.

- Not found. SLAVE.COMI; Can't start slave. (NPXON)

Remote File Access (RFA) is not enabled for this system because the file PRIMENET*>SLAVE.COMI does not exist.

- NRUSR INVALID (BINIT)

The number of remote users specified by an NRUSR directive exceeds '77 (63 decimal), the maximum number of configurable remote users.

- NSEG specified is too large, defaults to maximum value of n. (CINIT)

The NSEG directive specifies more segments than PRIMOS supports. The maximum value, which is printed in decimal, is used instead.

- NSEG specified is too small, default value is used. (CINIT)

The NSEG directive specifies fewer segments than PRIMOS supports. The default value, '1776 (1022 decimal), is used instead.

- NSLUSR is too big (NPXON)
  NSLUSR defaults to its maximum value: 63.

The number of slave users specified by an NSLUSR directive exceeds the maximum number of configurable slave users ('77, decimal 63). Cold start continues, with the system configured for 63 slave users.

- NTUSR+NAMLC GREATER THAN 255 (AINIT)

The number of terminal buffers and assigned buffers exceeds the maximum number of configurable buffers ('377, 255 decimal).

- NTUSR+NPUSR+NRUSR+NSLUSR GREATER THAN 255 (AINIT)

The sum of terminal, phantom, remote, and slave users exceeds the maximum number of configurable users ('377, 255 decimal).

- NUSEG HAS BEEN REPLACED BY EDIT PROFILE SUBCOMMANDS. (CINIT)

The obsolete NUSEG directive is in the configuration file. NUSEG is ignored and cold start continues. Delete NUSEG from the file.

- PAGDEV n does not point to the beginning of a valid file system partition.

The physical device number of the partition does not point to an area with a valid file system header. The partition may be a non-split paging partition which, if used previously with pre-Rev. 20 PRIMOS, has had the file system header paged over. After this message is printed, the operator is asked: "Are you SURE you want to page on PAGDEV n?" If the operator answers YES, the partition is split and used for the primary paging partition. If the operator answers NO, the partition is left as is and the operator is queried for another physical device number for PAGDEV. To eliminate this message at the next cold start, use MAKE (with the -NO_INIT and -BADSPOT_LEVEL 0 options) on the partition.

WARNING

The query is a check to help prevent paging over a valuable file system partition. Before answering YES, make sure that the partition is definitely intended for paging.

- PAGDEV n is an old partition which is not supported.

Certain types of partitions, particularly those with 440 words per disk record, are not supported after Rev. 15. The operator is prompted for another physical device number for PAGDEV. Convert your disk.

- PAGDEV n is not a split disk and conflicts with COMDEV.

COMDEV and PAGDEV cannot be the same unless they are part of a split partition. The operator is prompted for another physical device number for PAGDEV.

- PAGDEV n is not a valid pdev.

The physical device number supplied for the primary paging partition is not valid. This error often occurs because of an invalid device type. The device type must be '60 (pdev = nnnn6n), not '20 (floppy disk).

- PAGDEV n, partition <x>, has not previously been used for paging.

The non-split partition has never been used for paging or has been newly formatted with MAKE. This warning prevents the operator from unintentionally paging over valuable data on file system partitions. The operator is asked: "Are you SURE you want to page on PAGDEV n?". If the operator answers YES, the partition is split and becomes the primary paging partition. (This message is not repeated unless the partition is reformatted with MAKE.) If the operator answers NO, the partition is not used for paging and the operator is asked for another physical device number for PAGDEV.

- PARITY ERROR: DSWPB: m DSWRMA: n DSWSTAT: o

A memory parity check was generated. The diagnostic status words (DSW) give the location of the check, the instruction that caused the check, and the type of the check. Issue a RUN command to step past this error. If the RUN command does not solve the problem, reboot with the '1000 switch (bit 7) set in the BOOT command option word so that the boot does not run in machine-check mode.

- <primos error> PRnnnn (PRIMOS)

A file system error was encountered by the preloader while attempting to open or read the indicated PRIMOS PRnnnn file. Restore the file from a backup tape and run FIX_DISK. In the meantime, boot PRIMOS from tape.

- PRnnnn does NOT exist.

A file system error was encountered by the preloader while attempting to open or read the indicated PRIMOS PRnnnn file. Cold start halts. Restore the PRIRUN directory from a backup tape and run FIX_DISK, or boot from tape.

- PRIMOS.COMI not found.

The system startup file (PRIMOS.COMI or C_PRMO) does not exist in CMDNC0. Cold start continues and the operator is queried for the COMDEV, PAGDEV, and NTUSR parameters. Before the next cold start, install PRIMOS.COMI in CMDNC0.

- Profile data cannot be initialized from the SAD for the supervisor. System defaults are being used.

The SAD cannot be read, the SAD does not exist, or there is no entry in the SAD for the supervisor terminal under the entry SYSTEM. The profile of SYSTEM will be initialized with default attributes.

- PROTOCOL IN SMLC DIRECTIVE TOO LONG; FIRST SIX CHARACTERS: pppppp (CINIT)

The protocol argument to the SMLC CNTRLR directive specifies a protocol that is too long. The controller is disabled. See Chapter 10 of this guide for valid protocols.

- <primos error> READING CONFIG FILE (PRIMOS)

An error occurred while reading the system configuration file. Boot PRIMOS manually (by setting the '100000 bit in the BOOT command option word) and run FIX_DISK on the command partition. If necessary, build a new configuration file.

- RESTART PLEASE

This message appears following any error message printed by the PRIMOS initialization logic. The system halts. Correct the offending directive in the configuration file and cold start the system.

- 2ND INSTRUCTION STREAM DID NOT START.

The slave processor of your Prime 850 did not start. Press the MODE SELECT button on the system status panel to disable the slave processor and restart the system. (This extinguishes the MULTI STREAM light and places the system in single-stream mode.) If the system fails to verify, press the ISU button to select the other CPU. In either case, contact your Customer Support Center.

- Seek error on ALTDEV n. Disk not ready.

The physical device number of the ALTDEV directive points to a nonexistent controller, a nonexistent disk drive, or a non-operating disk drive. ALTDEV is ignored. Check that the specified physical device number is correct and that the disk pack containing the alternate paging partition was not moved to a different disk drive or a different controller. If the problem persists, contact your Customer Support Center.

- Seek error on COMDEV n. Disk not ready.

The physical device number of the COMDEV directive points to a nonexistent controller, a nonexistent disk drive, or a non-operating disk drive. Check that the specified physical device number is correct and that the disk pack containing the command partition was not moved to a different disk drive or a different controller. The operator is queried for another physical device number for COMDEV. If the problem persists, contact your Customer Support Center.

- Seek error on PAGDEV n. Disk not ready.

The physical device number of the PAGDEV directive points to a nonexistent controller, a nonexistent disk drive, or a non-operating disk drive. Check that the specified physical device number is correct and that the disk pack containing the primary paging partition was not moved to a different disk drive or a different controller. The operator is queried for another physical device number for PAGDEV. If the problem persists, contact your Customer Support Center.

- <primos error > SHRINKING (TFLADJ)

An error occurred while deleting unused terminal buffers to free up memory.

- SMLC CTRLR # OUT OF RANGE (CINIT)

An SMLC directive specifies an invalid controller number.

- SMLC LINE # OUT OF RANGE: n (CINIT)

An SMLC directive specifies an invalid logical line number. The logical line number must be between 0 and 7, inclusive.

- System NOT configured with maximum possible memory:
  only using mK BYTES, when nK BYTES are available.

The value for the MAXPAG directive results in the system using less than the total amount of available physical memory. This message is only a warning, and the MAXPAG directive is obeyed.

- TAPE ERROR, STATUS: n.

A tape I/O error occurred. The status words are displayed. Check that the device is online and that the tape is mounted. Correct the error and reboot.

- The NET config directive no longer enables the network.

  Please use the START_NET command. (CINIT)

The obsolete NET directive is in the configuration file. NET is ignored and cold start continues. When the system comes up, start the network with the START_NET command. Delete the NET directive from the configuration file.

- The ALTDEV pdev of n should point to a y head partition,
  but points to a z head partition.

The number of heads specified by the physical device number in the ALTDEV directive does not agree with the number of heads in the partition according to the DSKRAT written in record 1 of the indicated partition. Check that the physical device number is correct. ALTDEV is ignored.

- The COMDEV pdev of n should point to a y head partition,
  but points to a z head partition.

The number of heads specified by the physical device number in the COMDEV directive does not agree with the number of heads in the partition according to the DSKRAT written in record 1 of the indicated partition. Check that the physical device number is correct. Cold start continues and the operator is prompted for another physical device number for COMDEV.

- The number of async lines (m) present exceeds the allowable maximum of n.

More AMLC, ICS1, and ICS2 controllers (including ICS2 LAC cards) are present than are supported by PRIMOS. If n is less than 255, edit your configuration file and increase NTUSR. If n is greater than 255, contact your Customer Support Center for assistance on total number of lines.

- The PAGDEV pdev of n should point to a y head partition, but points to a z head partition.

The number of heads specified by the physical device number in the PAGDEV directive does not agree with the number of heads in the partition according to the DSKRAT written in record 1 of the indicated partition. Check that the physical device number is correct. Cold start continues and the operator is prompted for another physical device number for PAGDEV.

- TOO MANY CHARACTERS INPUT, START AGAIN

The pathname you entered for the boot runfile contains more than 128 characters.

- TREENAME NOT A SAM FILE

The file whose name was provided to the boot is not a SAM file but is a DAM or CAM file, a UFD, or a segment directory. The operator is queried for another filename.

- UNABLE TO INITIALIZE DMQ AMLC (AINIT)

The total size of the DMQ buffer sizes specified exceeds 64,000 halfwords.

- Unknown ICS directive: keyword ignored.

An ICS directive specifies an unrecognized keyword. Valid ICS keywords are CARDS, INPQSZ, INTRPT, and JUMPER. The directive is ignored and cold start continues.

- Virtual memory inconsistency (INIPAG) probable cause is invalid MAPGEN input or operation.

Contact your Customer Support Center.

-
  ```
  *****  WARNING  ******
  2ND INSTRUCTION STREAM NOT IN USE.
  ```

Your Prime 850 is running in single-stream mode (the MULTI STREAM light on the system status panel is not lit). The system continues running, but with lower throughput than in multi-stream mode.

> WARNING
>
> Do not press the MODE SELECT button or the ISU SELECT button on the system status panel while PRIMOS is running.

-
  ```
  Warning: BAD LINE # IN AMLBUF COMMAND. (CINIT)
  Must be greater than or equal to 0 and less than
  or equal to n(dec).
  AMLBUF directive ignored.
  Directive in error is: string
  ```

The specified line number is either less than 0 or greater than the maximum number of lines that PRIMOS currently supports, which is n (decimal). string is the entire AMLBUF directive as it appears in the configuration file. The directive is ignored and cold start continues.

-
  ```
  Warning: BAD LINE # IN AMLBUF COMMAND. (CINIT)
  Must be greater than or equal to 0 and less than
  or equal to NTUSR+NRUSR+NAMLC-2 to set I/O buffers.
  AMLBUF directive to set I/O buffers ignored.
  Directive in error is: string
  ```

The specified line number is either less than 0 or greater than the the sum of the NTUSR, NRUSR, and NAMLC directives minus two. string is the entire directive as it appears in the configuration file. The directive is ignored and cold start continues.

-
  ```
  Warning:  ICS JUMPER directive will be phased out.
  Please use ASYNC JUMPER instead.  (CINIT)
  ```

The configuration file contains the ICS JUMPER directive, which will be phased out at a future revision of PRIMOS. The value specified by ICS JUMPER is configured and cold start continues. To prevent this message, replace the ICS JUMPER directive with the ASYNC JUMPER directive.

- Warning: Input buffer size greater than maximum of 4095(dec).
  Maximum size will be used. (CINIT)

The specified size of the input buffer for the AMLBUF or ASRBUF directive is greater than the maximum of '7777 (4095 decimal). The maximum value is used.

- Warning: Input buffer size less than 0(dec).
  Current size of n(dec) will be used. (CINIT)

The specified size of the input buffer for the AMLBUF or ASRBUF directive is less than 0. The value n (which is usually the default of '200, 128 decimal) is used.

- Warning: Invalid attempt to set input/output buffer size in
  REMBUF range. Input/output buffer size ignored. (CINIT)

The AMLBUF or ASRBUF directive attempted to set the size of a buffer for a line for a remote user. The directive is ignored. To change the size of these buffers, use the REMBUF directive.

- Warning: Invalid DMQ buffer size specified.
  It must be one of the following: 16, 32, 64, 128, 256, 512,
  1024(dec). Default size will be used. (CINIT)

The specified size of the DMQ buffer for an AMLBUF directive is not a valid value. The default value of '40 (32 decimal) is used.

- Warning: Output buffer size greater than maximum of 4095(dec).
  Maximum size will be used. (CINIT)

The specified size of the output buffer for the AMLBUF or ASRBUF directive is greater than the maximum of '7777 (4095 decimal). The maximum value is used.

- Warning: Output buffer size less than minimum of 50(dec).
  Minimum size will be used. (CINIT)

The specified size of the output buffer for the AMLBUF or ASRBUF directive is less than the minimum of '62 (50 decimal). The minimum value of '62 is used.

- Warning: Output buffer size less than 0(dec).
  Current size of n(dec) will be used. (CINIT)

The specified size of the output buffer for the AMLBUF or ASRBUF directive is less than 0. The value n (which is usually the default of '200, 128 decimal) is used.

- WARNING - m SEGMENTS MAY NOT BE ENOUGH FOR n USERS

The NSEG directive specifies fewer segments than the expected minimum of three per configured users. Both values are printed in octal. n is NTUSR + NRUSR + NSLUSR + NPUSR. Users probably will receive the error condition NO_AVAIL_SEGS$ as the system use becomes high. Increase the value specified by NSEG.

C
EDIT_PROFILE Messages

This appendix lists the error and information messages displayed by EDIT_PROFILE. Following each message is an explanation of the message. Variable names in the messages are enclosed in left and right angle-brackets (for example, <user_id>).

A bracketed word at the end of each explanation indicates the message is in one of the following categories:

COMMAND The current command is aborted and the user is returned to the EDIT_PROFILE > prompt.

FATAL EDIT_PROFILE is aborted and the user is returned to PRIMOS. Fatal error messages are usually preceded by a standard PRIMOS error message.

INIT An error occurred while processing the PRIMOS command line invoking EDIT_PROFILE. The user is returned to PRIMOS command level.

NOTICE The message is only advisory or informative. Execution of the command continues.

RETRY Data of an invalid format has been entered. Correct data must be entered before execution can continue.

INITIALIZATION ERRORS

- <primos error> Can't inhibit interrupts

The call to PRIMOS that disables external interrupts during EDIT_PROFILE initialization failed. Report this error to your Customer Support Center because it indicates a serious problem with either PRIMOS or EDIT_PROFILE. [FATAL]

- Can't read the SAD: bad version number.

This error was caused by one of the following two things:

- The SAD was created by a later version of EDIT_PROFILE. For example, you are using 19.2 EDIT_PROFILE on a SAD created with 20.0 EDIT_PROFILE. You must use a version of EDIT_PROFILE at least as recent as the version that built the SAD.

- The UVF may have been damaged. Restore the SAD from backups or rebuild it.

In both cases, EDIT_PROFILE aborts. [FATAL]

- <primos error> Can't read user ID

The user ID of the user running EDIT_PROFILE could not be retrieved from PRIMOS. Report this error to your System Analyst because it indicates a serious problem with either PRIMOS or EDIT_PROFILE. [FATAL]

- <filename> created at <datetime>.

When in Initialization mode, EDIT_PROFILE informs you as it creates the files directly contained in the SAD. [NOTICE]

- *** Creating project "DEFAULT".

When you create a password SAD, project DEFAULT is always created automatically. EDIT_PROFILE informs you of this fact. [NOTICE]

- Directory pathname too long.

The pathname of the SAD's parent directory that you supplied to EDIT_PROFILE was longer than the limit of 80 characters. The limit ensures that the longest subtree name in the SAD can be appended to the parent tree within the 128-character limit for pathnames set for PRIMOS. [INIT]

- EDIT_PROFILE is in use. Please try again in a few minutes.

Another user is running EDIT_PROFILE in System Administrator mode. To prevent conflicting updates, only one user is allowed to run in System Administrator mode at a time. If no other users are authorized to use EDIT_PROFILE and you are logged in at only one terminal, this message can indicate a breach of security. [FATAL]

- Insufficient access rights. <sad_pathname>

You are not authorized to use EDIT_PROFILE on the specified SAD. [INIT]

- Parent directory is not an ACL directory.

You attempted to create a SAD in a non-ACL directory. You can create a non-ACL SAD only from the supervisor terminal on the MFD of the command device. [FATAL]

- Parent pathname may not be used with -MFD_PASSWD option.

The -MFD_PASSWD option specifies the MFD owner password when the SAD resides in a password MFD. Because test SADs may reside only in ACL directories, the combination of -MFD_PASSWD and a parent pathname is inconsistent. [INIT]

- *** Protection in the SAD has been damaged ***
  Do you want EDIT_PROFILE to fix it?

The SAD and UVF can be accessed, but the Master Group File (MGF) and/or the Master Project File (MPF) cannot. The probable cause of this error is that the ACLs protecting the files and directories of the SAD have been damaged or changed. If you answer YES to the query, EDIT_PROFILE resets the proper protection on the SAD and continues execution. If you answer NO, EDIT_PROFILE aborts and returns you to PRIMOS. [NOTICE]

- *** Read/write locks in the SAD have been damaged ***
  Do you want EDIT_PROFILE to reset them?

The UVF and MPF can be accessed, but a file-in-use error was returned on the MGF. The read/write locks in the SAD have been changed from the settings initially made by EDIT_PROFILE, most likely because the SAD was copied without the -COPY_ALL option. If you answer YES to the query, EDIT_PROFILE resets the read/write locks and reinitializes. If you answer NO, EDIT_PROFILE aborts and returns you to PRIMOS. [NOTICE]

- SAD does not exist. Create it?

No SAD exists in the current directory. (If you used the EDIT_PROFILE command without an argument, the current directory by default is the MFD of logical device zero.) Answer YES to create a SAD. Answer NO to end EDIT_PROFILE and return to PRIMOS. [NOTICE]

- *** SAD is either not properly set up or has been damaged ***

The SAD was found but the User Validation File (UVF) cannot be accessed. Possible causes of this error include the following:

- The UVF was inadvertently deleted.
- The partition on which the SAD resides is damaged.
- A previous initialization of the SAD was aborted.
- An incomplete MAGRST of the SAD was done.
- An incomplete COPY of the SAD was done.

The message is always followed by the advisory message "Restore from backup or delete and re-initialize." To solve the problem, either restore a good copy of the SAD from a backup disk or tape, or delete the damaged SAD and create a new one. [FATAL]

- Size must be a number between zero and 28004.

You entered a negative number or a number greater than 28,004 at the "Projected number of users:" prompt. [RETRY]

- System administrator = "<sa_name>".

When a SAD is created from a terminal other than the supervisor terminal, the user running EDIT_PROFILE automatically becomes the System Administrator (because of ACLs). EDIT_PROFILE informs you that it has set the System Administrator in this SAD to be the name specified. [NOTICE]

- Warning: security and project support cannot be provided without ACLs.

Restrictions will apply on the password SAD that you created at the supervisor terminal. [NOTICE]

- <primos error> When adding Priority ACL.

A priority ACL could not be set for the command device. (When EDIT_PROFILE is run in Initialization mode at the supervisor terminal, it attempts to put a priority ACL on the partition to facilitate creation of ACLs in which user SYSTEM might not be found.) Report this error to your Customer Support Center because it indicates a serious problem with either PRIMOS or EDIT_PROFILE. [FATAL]

GENERAL ERRORS

- Cannot support names of depth greater than 16.

The maximum pathname depth for an initial attach point is 16 levels. Supply a new Initial Attach Point of 16 or fewer levels. [RETRY]

- Can't read project <project_id>: bad version number.

This message can indicate one of the following three errors:

  - This project was built with a version of EDIT_PROFILE that was later than the current version. (For example, the project was built with a Rev. 20.0 version of EDIT_PROFILE and your current version is Rev. 19.2.) You must use a version of EDIT_PROFILE at least as recent as the one that built the project.
  - The PVF is damaged. Either restore the SAD from backups or delete and rebuild the project.
  - Rev. 19.0 EDIT_PROFILE generated projects with invalid version numbers. These projects must be rebuilt (using the REBUILD command with the -PROJECT option) with Rev. 19.1 EDIT_PROFILE before they can be read with Rev. 20.0 EDIT_PROFILE.

In these cases, you return to the EDIT_PROFILE > prompt. [COMMAND]

- Command aborted; type "QUIT" to exit.

You used CONTROL-P to abort the current command. You are returned to the EDIT_PROFILE > prompt, where you can either continue the EDIT_PROFILE session or enter the QUIT command to return to PRIMOS. [COMMAND]

- Duplication of options in command.

You used the same command option more than once. All EDIT_PROFILE commands allow only one use of each option. The duplicated option is indicated by a caret (^). [COMMAND]

- *** EDIT_PROFILE system error: <error> when parsing command.

Report this error to your System Anaylst because it indicates an EDIT_PROFILE programming error. [FATAL]

- *** Group <group_name> not legal for this project.

When assigning a project-based group to a user (with ADD_USER or CHANGE_USER) or to a project profile (with ADD_PROJECT or CHANGE_PROJECT), that group was not found in the MPP for that project. The group is not assigned. [NOTICE]

- Illegal <object_type> "<name>".

name is not a valid object of type object_type. The valid types are user ID, password, group name, or project ID. EDIT_PROFILE continues to prompt you until you enter a valid type. [RETRY]

- Improper data format in command.

You used an incorrect format for an argument of a command or option. For example, the argument is a user or project ID that is longer than 32 characters or that contains an illegal character. The erroneous object is indicated by a caret (^) on the next line. [COMMAND]

- Incorrect format: "<option1>" and "<option2>" options are exclusive.

You used two command options that cannot be given together. [COMMAND]

- Incorrect format: "<option_name>" option requires an argument.

You used an option that takes an argument, but you supplied no argument. Reenter the command either without the option or with the required argument. [COMMAND]

- Incorrect format: No options allowed without <object_type>.

All commands that take objects require that the object be supplied if any options are given. You used one or more options, but supplied no object. Either use the command with no options (in most cases EDIT_PROFILE prompts for them), or supply an object. [COMMAND]

- *** Input truncated to 256 characters.

The command line or response contained more than 256 characters. All characters past the 256th are ignored. [NOTICE]

- "<project_id>" is not a valid project.

The requested project does not exist or, in Project Administrator mode, is not under the jurisdiction of the Project Administrator. [COMMAND]

- *** New <object> added to <location>: "<name>".

A new object of the given type was added to the data bases. object is PROJECT or GROUP. location is SYSTEM or PROJECT. name is the name of the object being added. The message allows you to check your input, in case you made a typographical error and inadvertently created a new group or project that no one can use. (For example, if you intended to add the group .OPSYS to a user's list and instead typed .OPSSS, you would get the message "*** New group added to system: .OPSSS". You would then add .OPSYS and delete .OPSSS.) [NOTICE]

- Pathname must be fully qualified.

When entering an Initial Attach Point, you did not supply an absolute pathname (that is, a pathname that includes the partition name). EDIT_PROFILE continues to prompt you until you enter the correct pathname format. [RETRY]

- Pathname must have at least one directory level.

When entering an Initial Attach Point, you supplied only the partition name. EDIT_PROFILE continues to prompt you until you include at least one directory name in the pathname. [RETRY]

- *** Project Data File overflow

You attempted to expand the Project Data File (PDF) for the project in which you were working to more than 64,000 entries. The command is aborted. Rebuild the project to delete any inactive entries from the PDF. If that does not solve the problem, break the project into two or more projects. [COMMAND]

- Token too long; truncated to "<token>".

You entered a token longer than 32 characters. Tokens (individual items) in a command line may not be more than 32 characters long. The value of the truncated token is displayed. This error is usually caused by a skipped blank or extra character instead of a blank between tokens. Further errors may result because of the truncation. [NOTICE]

- Too many objects specified in command.

You used more objects than the command expected. All EDIT_PROFILE commands take, at most, one object. Perhaps you left off the hyphen from an option name. The excess object is indicated on the following line by a caret (^). [COMMAND]

- Unrecognizable command "<command>".

You either issued a command that is restricted to the System Administrator while you were in Project Administrator mode or a command that is completely unknown to EDIT_PROFILE. [COMMAND]

- Unrecognizable option in command.

You either used an incorrect command option or an option that is restricted to the System Administrator while you were in Project Administrator mode. The command line option list is repeated and the erroneous option is indicated by a caret (^) on the next line. [COMMAND]

- User already belongs to 16 groups.

You used the ADD_PROJECT or CHANGE_PROJECT command and the Project Administrator you specified is a member of 16 system groups, but not a member of the .PROJECT_ADMINISTRATORS$ group. Either delete one or more of the new Administrator's groups, or choose another Administrator. [COMMAND]

- User <user_id> isn't registered, do you want to register <user_id>?

You used the ADD_PROJECT or CHANGE_PROJECT command and the Project Administrator you specified is not in the SAD. If you answer YES, you enter the ADD_USER dialog to create the new Administrator's entry. If you answer NO, the command is aborted. [COMMAND]

ADD_PROJECT MESSAGES

- *** Can't find like reference "<project_id>".

You used the -LIKE option, but the project whose attributes were to be copied does not exist. [COMMAND]

- *** Project "<project_id>" already exists. Must use Delete or Change.

You used the command for an existing project. Either use CHANGE_PROJECT to change the attributes of the existing project, or use DELETE_PROJECT to delete the existing project and then ADD_PROJECT to create a new project with the old name. [COMMAND]

- Project "<project_id>" created.

The command executed successfully. [NOTICE]

- Projects not supported in non-ACL systems.

You attempted to create a project in a password SAD. You must convert the SAD to ACL protection (with SET_DEFAULT_PROTECTION) before you can use ADD_PROJECT. [COMMAND]

ADD_USER MESSAGES

- *** Can't find like reference "<user_id>".

You used the -LIKE option, but the user whose attributes were to be copied does not exist. If the -PROJECT or -DEFAULT options were given, this may mean that the referenced user is either not in the UVF, or is not in the PVF of the specified project. [COMMAND]

- User "<user_id>" added to project "<project_id>".

The user was successfully added to the specified project. [NOTICE]

- User "<user_id>" added to system.

The user was successfully added to the UVF. If only project DEFAULT exists, the user was also added to its PVF. [NOTICE]

- *** User "<user_id>" already in project "<project_id>". Must use Delete or Change.

You attempted to add a user to a project, but the user is already in the project. Either use CHANGE_USER to change the attributes of the existing user, or use DELETE_USER to delete the user and then use ADD_USER to add a new user with the old user ID. [COMMAND]

- *** User "<user_id>" already on system. Must use Delete or Change.

You attempted to create a user ID, but a user with that ID is already on the system. If the existing user is no longer using the system, use DELETE_USER to remove that user and then use ADD_USER to add the new user. If the existing user is still using the system, use a different user ID for the new user. [COMMAND]

- Verify_ns option may only be used by true SA; ignored.

The -VERIFY_NS option was used by a Project Administrator or in a test SAD. Because the -VERIFY_NS option opens SADs on remote systems, the option can be used only by the System Administrator as known to PRIMOS. The option is ignored and execution continues. [NOTICE]

- Warning: all users must have an initial attach point.

You did not specify an Initial Attach Point for the user being added. All users must have an Initial Attach Point to log in. If the project profile of the project in which the warning occurred has an Initial Attach Point, the message may be ignored. If it does not, the user must be given an Initial Attach Point to log in to that project. [NOTICE]

- Warning: Project "<project_id>" is overloaded.

The Project Validation File is more than 75% full, or the number of overflow entries in the PVF is more than 10% of the total number of entries. For maximum efficiency, the PVF should be rebuilt. If the -NO_QUERY option was not given, EDIT_PROFILE asks if the PVF should be rebuilt. This warning is not given if the PVF is already at the maximum size. [NOTICE]

- Warning: User "<user_id>" found on system(s): <list>

You used the -VERIFY_NS option and the user was found on at least one other system in the naming sphere. The message lists all the systems on which the user was found. [NOTICE]

- Warning: User validation file is overloaded.

The User Validation File is more than 75% full, or the number of overflow entries in the UVF is more than 10% of the total number of entries. For maximum efficiency, the UVF should be rebuilt. If the -NO_QUERY option was not given, EDIT_PROFILE asks if the UVF should be rebuilt. This warning is not given if the UVF is already at the maximum size. [NOTICE]

CHANGE_PROJECT MESSAGES

- Only one administrator allowed in non-ACL systems.

You used the -CHANGE_PA option on a password system. On non-ACL systems, the Project Administrator for project DEFAULT must always be the System Administrator. [COMMAND]

- Project "<project_id>" is being modified. Please try again in a few minutes.

Another user is using EDIT_PROFILE on the specified project. If only one person has access to this project, this message may indicate a breach of security. [COMMAND]

- Project "<project_id>" updated <date/time>.

The command executed successfully. [NOTICE]

CHANGE_SYSTEM_ADMINISTRATOR MESSAGES

- <primos error> Calling Chg$sa

The call to the PRIMOS routine CHG$SA to change the System Administrator's name has failed. This is a serious error and should be reported to your Customer Support Center. [FATAL]

- <primos error> Can't set priority ACL.

An error occurred while PRIMOS attempted to set a priority ACL. EDIT_PROFILE uses this priority ACL to ensure access to the SAD during the changeover from the old System Administrator to the new one. Report this error to your Customer Support Center because it indicates a serious problem in PRIMOS. [FATAL]

- Change_sa command may not be used on test SADs.

The CHANGE_SYSTEM_ADMINISTRATOR command is valid only when operating on the SAD in the MFD of the command partition, because that is the only case in which the copy of the System Administrator's name in PRIMOS may be changed. [COMMAND]

- *** Mandatory exit from EDIT_PROFILE ***

The CHANGE_SYSTEM_ADMINISTRATOR command executed successfully. EDIT_PROFILE terminates because the old System Administrator no longer has access to files in the SAD. [FATAL]

- *** New administrator not found on system.

The new System Administrator has no entry on the system. This message is followed by the prompt "Create entry:". EDIT_PROFILE then automatically enters the ADD_USER dialog to allow you to create an entry for the new System Administrator. [NOTICE]

- New administrator's name same as old one!

The name given as that of the new System Administrator is the name of the existing System Administrator. The command is ignored. [COMMAND]

- *** System administrator name is not known by PRIMOS.

PRIMOS normally holds the System Administrator's name in its internal data base. When the SAD is first created, however, the name is not read by PRIMOS until the system has been rebooted. Because PRIMOS allows the System Administrator's name to be changed only by the current System Administrator, the CHANGE_SYSTEM_ADMINISTRATOR command may be used only after that name is established. This message is followed by the advisory message "System must be re-booted before Change_sa command may be used." [COMMAND]

CHANGE_USER MESSAGES

- Options "-add" or "-delete" may be put only at the beginning of the command.

The -ADD or -DELETE option was given as part of the new group list, but the list began with a group name. If either -ADD or -DELETE is given, one of these options must be the first item in the list. [COMMAND]

- *** User "<user_id>" not found in project "<project_id>".

The user ID whose project-based attributes were to be changed does not have an entry in the specified project. Check for misspellings of both the user ID and the project ID. [COMMAND]

- *** User "<user_id>" not found on system.

The user ID whose attributes were to be changed does not exist. Check for possible misspellings. [COMMAND]

- Warning: all users must have an initial attach point.

You did not specify an Initial Attach Point for the user or you removed the user's existing Initial Attach Point. All users must have an Initial Attach Point to log in. If the project profile of the project in which the warning occurred has an Initial Attach Point, the message may be ignored. If it does not, the user must be given an Initial Attach Point to log in to that project. [NOTICE]

- User "<user_id>" updated <date/time>.

The command executed successfully. [NOTICE]

DELETE_PROJECT MESSAGES

- *** Can't delete DEFAULT unless other projects exist.

Project DEFAULT cannot be deleted if it is the only project on the system. [COMMAND]

- *** Can't delete "<filename>": <primos error>.

The specified file could not be deleted. Execution continues, but you should probably delete the file later with the DELETE command. [NOTICE]

- (<count> default projects reset.)

count users had the deleted project as their default login project. Because that project no longer exists, these users now have no default login project, and thus must always supply a project ID when they log in. [NOTICE]

- *** Project "<project_id>" deleted <date/time>.

The command executed successfully. [NOTICE]

DELETE_USER MESSAGES

- *** Can't delete System Administrator!

The System Administrator must always have an entry in the UVF. An attempt to delete this entry has been rejected. [COMMAND]

- (Project "<project_id>" is now empty.)

The user who was deleted was the last user in the specified project. That project's PVF now contains no entries. [NOTICE]

- PROJECT option not available when only DEFAULT project present.

If there is only one project on the system, a user may not be deleted only from that project. To remove the user from the system, use the DELETE_USER command without any options. [COMMAND]

- User "<user_id>" deleted from project "<project_id>".

The user was successfully deleted from the PVF of project_id. If the -PROJECT option was not given, this message is displayed for each project from which the user was deleted. [NOTICE]

- User "<user_id>" deleted from system <date/time>.

The user was successfully removed from the UVF. [NOTICE]

- *** User "<user_id>" not found in project "<project_id>".

The user you attempted to delete has no entry in the project. Check for misspellings of the user ID and the project ID. [COMMAND]

- *** User "<user_id>" not found on system.

The user you attempted to delete has no entry in the UVF. The command continues to delete the user from all projects. [NOTICE]

DETACH_PROJECT MESSAGES

- "<project_id>" is not the current project.

The specified project ID does not match the name of the current project. Check for a misspelling of the project ID. [COMMAND]

COMMAND ENVIRONMENT MESSAGES

- Attribute limits should be set up first.

You attempted to set up a user's individual command environment limits without first setting up the project attributes. [COMMAND]

- Attribute should be numeric.

You entered a value that was not a positive number when prompted for EPF attributes. [RETRY]

- ***EPF Attributes are not set

You have not defined the EPF attributes for a user or a project. [NOTICE]

- Invalid number of <resource-units>:

You specified a number that was too small or too large as the attribute for a project or a user. resource_units is one of the following: command levels, program invocations per level, dynamic segments, or static segments. [RETRY]

- Number of <resource_units> exceeds the limit:

The number of resource_units for the project is greater than the maximum number allowed by the Project Administrator for the project. resource_units is one of the following: command levels, program invocations per level, dynamic segments, or static segments. [RETRY]

- The sum of dynamic and static segments exceeds current system limit.

The sum of the values for the dynamic and static segments for the project you are defining is greater than 512. [RETRY]

- User attribute may not be null.

A value must be entered. After this message appears, the prompt is repeated. [RETRY]

- User attribute should be numeric.

A non-numeric value cannot be entered as a limit. After this message appears, the prompt is repeated. [RETRY]

LIST_PROJECT MESSAGES

- *** User "<user_id>" not found in project "<project_id>".

The user specified in the -USER option does not exist in the project. [COMMAND]

- Can't open "<filename>" for output. Please try again.

The file specified in the -OUTPUT option cannot be opened for output. Reissue the command, making sure that you do not make a typographical error and that you supply a pathname (not a simple filename) with the -OUTPUT option. (If you supply a simple filename with the -OUTPUT option in Project Administrator mode, the output file cannot be opened because of insufficient access rights on the SAD.) [COMMAND]

LIST_SYSTEM MESSAGES

- Can't open "<filename>" for output. Please try again.

The file specified in the -OUTPUT option could not be opened. Check for typographical errors and try again. [COMMAND]

- GROUPS option not supported in non-ACL SADs.

The -GROUPS option is illegal in a password SAD because there are no ACLs and thus no ACL groups. [COMMAND]

LIST_USER MESSAGES

- *** User "<user_id>" not found in project "<project_id>".

The specified user does not have an entry in the specified PVF. [NOTICE]

- *** User "<user_id>" not found on system.

The specified user does not have an entry in the UVF. If the -PROJECT or -ALL option was given, the command continues to search for the user in the PVF. [NOTICE]

MINIMUM_PASSWORD_LENGTH MESSAGE

- Illegal password "<passwd>".

You attempted to give a user a password that contains fewer characters than the length specified by the MINIMUM_PASSWORD_LENGTH command.

NO_NULL_PASSWORD MESSAGE

- Warning: the following users currently have null passwords: <list>

You issued the command either with no options or with the -ON option. Users who are in violation of the new standard are listed. [NOTICE]

REBUILD MESSAGES

- *** <file> backed up into file "<file>.OLD" <date/time>.

When a rebuild takes place, EDIT_PROFILE informs you of the names of the backup files it creates as each file is backed up. [NOTICE]

- Duplicate entry for user "<user_id>" (entry <number>).

Contact your Customer Support Center because a serious error in the SAD data base has been found. The rebuild is aborted, and the original UVF, MPF, and MGF are replaced by their backups. This message is preceded by the warning: "*** EDIT_PROFILE system error! ***". [COMMAND]

- <primos error> when copying files.

The specified error occurred while copying to or from the backup files used during the rebuild. If the message occurs before all the "File xxx backed up..." messages appear, the original files are still in a consistent state. If it occurs after all the initial copies are done, restore the files in question from their backup copies. The cause of this problem is often a serious physical disk or hardware error, and if so it should be reported to your Customer Support Center. [FATAL]

- The following project-id's have been removed from the MPF: <list>

During a system rebuild, inactive entries are removed from the MPF and MGF. Projects that are no longer valid are listed. [NOTICE]

- *** Rebuild complete <date/time>! ***

The rebuild executed successfully. [NOTICE]

SET_DEFAULT_PROTECTION MESSAGES

- <primos error> Converting MFD.

The error occurred while attempting to put an ACL on the MFD. Report this to your Customer Support Center because it indicates a serious error in PRIMOS or EDIT_PROFILE. [FATAL]

- Master Group File created <date/time>

The -CONVERT option was used, which creates a Master Group File (MGF) that holds the names of all ACL groups that are valid on the system. [NOTICE]

VERIFY_USER MESSAGES

- <primos error> in X$stat call.

EDIT_PROFILE could not gather information about the PRIMENET network. This generally indicates a serious problem with PRIMENET and, if repeated, should be reported to your Customer Support Center. [FATAL]

- No room. Too many nodes in network.

Your network has more than 256 nodes configured. EDIT_PROFILE aborts because it probably suffered damage to its stack while attempting to get information on the network. EDIT_PROFILE probably cannot terminate its run successfully after this message. To solve this problem, reduce the number of nodes configured in your network. [FATAL]

- Only true SA may use Verify_user command.

The command was used by a Project Administrator or in a test SAD. Because the VERIFY_USER command opens SADs on remote systems, the command can be used only by the System Administrator as known to PRIMOS. [COMMAND]

- User id and -ALL option are exclusive.

Either a user ID or the -ALL option may be given, but not both. [COMMAND]

- Warning: user "<user_id>" found on system(s): <list>

The specified user ID was found on at least one other system. All systems on which the ID was found are listed. [NOTICE]

D
Obsolete and Rarely Used Commands and Directives

This appendix describes the AMLC command, an octal method of configuring asynchronous lines; the LOOK command, a debugging tool; and the configuration directives FILUNT, PREPAG, RWLOCK, AND VPSD.

OBSOLETE COMMANDS

▶ 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 ALMC 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. The command defines the following:

- The protocol for the line
- The use of Auto Speed Detect (ASD) by the line
- The line speed, bit pattern, and parity for the line
- The initial terminal characteristics of the line
- The user number associated with the line (or the AMLC command defines the line as assignable)

The format of the AMLC command is as follows:

```
    AMLC [protocol] line [[configuration [lword]]
```

All numbers used in the AMLC command must be in octal format. The values and meanings of the arguments are explained in the following sections.

The protocol Argument

The values for the protocol argument are TRAN (the default), TTY, TT8BIT, TTYUPC, TTYNOP, and ASD. The basis for selection of the protocol is discussed below.

Protocols for older model AMLC boards (model 5054) are discussed later in this appendix.

TRAN: TRAN, the transparent protocol, is usually used by lines connected to peripheral devices or to other computers. With transparent protocol, no input is echoed, no response is made to the input of a line feed or carriage return, and no transformation of carriage return to line feed is made. CONTROL-P has no special meaning under this protocol and is passed through to the program.

If you do not specify a protocol when using the AMLC or ASSIGN AMLC commands, TRAN is assigned by the operating system.

TTY: TTY, the terminal protocol, is the protocol assigned at cold start to lines controlling interactive terminals. With TTY protocol, all input from the terminal is echoed if the line is set for full duplex; a carriage return and a line feed are echoed following carriage return. The high order bit (the ASCII code parity bit) of each character input from the terminal is forced on. CONTROL-P and BREAK are interpreted as a QUIT command if the terminal is connected to the system as a user terminal.

If the terminal is connected to the PRIMOS operating system as an assigned line, CONTROL-P, BREAK, and line feed input from the terminal are ignored and discarded. A carriage return input from the terminal is transmitted as a new line (or line feed) to the program requesting input. If the associated input buffer becomes full, input is no longer echoed and any additional characters that are typed are lost.

TT8BIT: TT8BIT behaves in the same manner as the TTY protocol except that the high order bit (ASCII parity bit) is not forced on for each character input from the terminal. All control characters are handled in the same manner as the TTY protocol.

Use this protocol only if you have an Arabic DM5E/PLUS terminal.

To use this protocol, you must set bit 13 of the line configuration word to 0 to enable parity and bit 14 to 0 to select odd parity, as shown in the example below:

```
    AMLC TT8BIT 5 2403 /* Line 5:  odd parity enabled, 9600 baud
```

TTYUPC: TTYUPC, the uppercase translating protocol, is used to avoid sending lowercase output to terminals or peripheral devices that cannot print lowercase characters. This is the only difference between TTY and TTYUPC protocols.

TTYNOP: TTYNOP configures the asynchronous DIM to ignore all traffic on the line. If you have a noisy line to which no device is connected, use this protocol to ensure that the CPU does not waste time trying to interpret any noise coming from this line as commands that it must process.

ASD: ASD (Auto Speed Detect) allows PRIMOS to detect automatically the transmission speed of a user terminal. Depending on how you configure the line, PRIMOS can detect baud rates of 110, 300, 600, 1200, 2400, 4800, 9600, and 19200 bits per second. For details on setting and using ASD, see the section Enabling Auto Speed Detect in Chapter 11, CONFIGURING ASYNCHRONOUS LINES.

Protocols For Older Model AMLC Boards

If you have lines attached to an older model 5054 AMLC board (also known as the DMT AMLC board), use TTYHS, TRANHS, or TTYHUP as the value for the AMLC command's *protocol* argument. TTYHS, TRANHS, and TTYHUP are referred to as high-speed protocols.

Do not use these protocols on the following lines:

- Lines attached to Model 5154 AMLC boards (also known as QAMLC or DMQ AMLC boards) or to ICS controllers

- Lines that normally have their character-time-interrupt flag always set (for example, the last line on the last AMLC)

Depending on the baud rate and the number of lines in the group, lines using high-speed protocols can greatly increase the system overhead.

TTYHS and TRANHS: TTYHS and TRANHS are used by lines connected to peripheral devices that can run at greater than 1200 baud, which is the standard terminal speed. For example, lines using the high-speed protocols can run at 9600 baud.

TTYHS and TRANHS are the same as the protocols described in Chapter 11, with one exception: For output only, the line's character time interrupt flag is set when the output buffer contains more than 40 characters, and it remains set until the output buffer contains fewer than 40 characters. The protocols have a burst-mode effect on the output device.

TTYHUP: TTYHUP, the high-speed translating protocol, is used to avoid sending lowercase output to terminals or peripheral devices that cannot print lowercase characters.

The line Argument

The line number is an octal number from 0 to the highest line number present in a system. The maximum value is '377 (255 decimal). These lines may exist on AMLC or ICS controllers. No more than 128 lines may be on AMLC controllers. For ICS2 and ICS3 controllers, the async line adapter cards may be distributed over a maximum of four ICS2s, even if the total number of ICS2 lines is less than 256. ICS lines are allocated line numbers after AMLC lines are allocated.

The configuration Argument

The configuration argument, which sets the line configuration, is an octal number that corresponds to the bit pattern illustrated below. Some commonly selected values (shown below) are used for data sets with parity disabled and 8-bit character length. The bit pattern is read from left to right.

Note that the baud rate values can be incorrect if the default controller speeds have been changed by hardware jumpers on the AMLC or by the ASYNC JUMPER configuration directive on the ICS.

```
 -----------------------------------------------------------------
 |  |           |           |           |           |           |
 -----------------------------------------------------------------
   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16
```

Config	Baud Rate
2033	110
2113	134.5
2213	300
2313	1200 (default)
2413	9600 (1)
2513	75 (2)
2613	150 (2)
2713	1800 (2)

(1) Default for 2413 if AMLCLK is not set.

(2) Assigned by hardware jumper on AMLC or by the ASYNC JUMPER configuration directive. The values are the defaults as supplied by Prime.

The configuration number is constructed in the following way. (The bit count is read from left to right, beginning at 1).

Bits	Meaning
1,2,3,4	Line number on controller. Set by PRIMOS. Leave as 0.
5	Reserved. Set to 0.
6	0: Data set control off. 1: Data set control on. Required for modems and port selectors, ignored by terminals. (Default)
7	0: Do not loop line. (Default) 1: Loop line. (Used only for testing purposes.)
8,9,10	Asynchronous line speed (bits per second)

000	110
001	134.5
010	300
011	1200 (Default)
100	Programmable clock. (See AMLCLK directive.)
101	75 *
110	150 *
111	1800 *

* Assigned by hardware jumper on AMLC board or by ASYNC JUMPER directive.

Bits	Meaning
11	0: Disable Reverse Flow Control. (Default) 1: Enable Reverse Flow Control.
12	0: 1 stop bit. (Default) 1: 2 stop bits.
13	0: Enable parity. 1: Disable parity. (Default)
14	0: Odd parity. (Default) 1: Even parity.
15,16	Character length: 00 5 bits 10 6 bits 01 7 bits 11 8 bits (Default)

The lword Argument

The lword argument is an octal number that corresponds to a 16-bit halfword constructed as follows:

Bits			Meaning
1	Reset	(0)	Full-duplex (Default)
	Set	(1)	Half-duplex
2	Reset	(0)	Echo LINE FEED for RETURN. (Default)
	Set	(1)	Do not echo LINE FEED for RETURN. (Meaningful only if bit 1 is set.)
3	Reset	(0)	Do not recognize XOFF and XON.
	Set	(1)	Recognize XOFF (CONTROL-S, '223) and XON (CONTROL-Q, '221).
4	Reset	(0)	Terminal output not currently blocked.
	Set	(1)	Terminal output currently suspended (XOFF seen).
5	Set	(1)	Use data set sense.

Bits		Meaning
6	Reset (0)	If data set sense is on, treat as if XOFF was received. If data set sense is off, treat as if XON was received. Examined only if bit 5 is set.
	Set (1)	If data set sense (carrier detect signal) is off, treat as if XOFF was received. If data set sense is on, treat as if XON was received.
7	Set (1)	Enable error detection, send NAK, '225 if parity or overflow sensed.
8	—	Reserved. Set to 0.
9-16	—	User number. (Normally line number plus 2.) 0 means the line is assignable. (The system's default formula for determining the user number is usernumber = linenumber + 2.)

Caution

If you do not follow the system's default formula for the user number, two separate AMLBUF directives are required to set the size of the DMQ buffer and the input and output ring buffers for terminal users.

The user number is the number displayed by the STATUS USERS command or after login or logout. (The STATUS, LOGIN, and LOGOUT commands display this number as a decimal value, but its octal equivalent must be used in the AMLC command.) If the rightmost eight bits (9-16) of lword are zero, the line is not associated with any user space and is available to be assigned if the configuration directive NAMLC is greater than 0.

For example, to set line 3 as an assignable line using the transparent protocol and having a baud rate of 9600, the command line is as follows:

```
AMLC TRAN 3 2413 0
```

To reset line 3 as a line that can be used for logging in, with TTY protocol, a baud rate of 1200, and connected to user 5, the command line is as follows:

```
AMLC TTY 3 2313 5
```

Users can also issue the TERM command to set their terminal characteristics, such as specifying the duplex mode or disabling the quit or break character (CONTROL-P).

Data Set Sense: Bits 5 and 6 of the lword are used to support devices that toggle an RS-232-C pin to signal when they are busy/ready, rather than using XON/XOFF. RS-232-C pin 8, data carrier detect (DCD) is used for the signal.

Bit 5 of the lword indicates that pin 8 should be interrogated before performing an output operation. If pin 8 busy is detected, PRIMOS responds as if an XOFF was received for that line. When pin 8 goes to the ready state, it is flagged as if an XON was received, and output resumes. A device can signal busy by causing pin 8 to be raised or lowered.

Bit 6 is checked only if bit 5 is set. Bit 6 can be set to interrogate pin 8 either way. For example, if the device signals busy as pin 8 high (1), the lword bit setting is the following:

Bit 5 = 1 (Use data set sense.)

Bit 6 = 1 (If pin 8 is low, interpret as XOFF. If pin 8 is high, interpret as XON.)

On some devices, pins other than pin 8 may be used. If this is the case, the cabling should be arranged so that the data set sense signal is wired into carrier detect for the controller.

Data set sense is also referred to as buffered protocol or reverse channel protocol.

Error Detection: Setting bit 7 of the lword enables a limited set of error detection. When this bit is set, PRIMOS checks for overflow of the user input buffer, and for parity errors. If PRIMOS finds an error in an incoming character, it replaces that character with an '225 (ASCII NAK). If the user input buffer is full, an ASCII NAK is placed in the input buffer. PRIMOS always reserves one position in the input buffer in order to do this.

Reverse Flow Control

Reverse Flow Control allows XOFF characters to be sent to devices to indicate that the PRIMOS input queue or the user input buffer is full and cannot receive any more data. An XON character is sent to the device to indicate that transmission can resume when the queue is no longer full. Only devices that can interpret these characters (such as PT45, PST 100, and PT200) should use Reverse Flow Control.

To enable Reverse Flow Control, set bit 11 of the configuration word of the AMLC command, as in the following example.

```
    AMLC TTY 10 2453 /* Reverse Flow Control enabled at 9600 baud
```

Assigning and Unassigning Asynchronous Lines

The AMLC command configures assigned asynchronous lines as well as terminal lines. After the system is running, users can assign such asynchronous lines by using the following command:

```
    ASSIGN  AMLC  [protocol]  line  [configuration  [lword]]
```

The ASSIGN AMLC command associates the assigned line with the first free pair of input and output ring buffers in the pool of assigned line buffers.

Users can unassign the AMLC lines by using the following command:

```
    UNASSIGN AMLC line
```

You can issue the UNASSIGN command from the supervisor terminal to unassign any line or device, regardless of who assigned it or from which terminal it was assigned.

An UNASSIGN command followed by an ASSIGN command can reassign any device with optional line configuration data. The ASSIGN AMLC command cannot be issued twice in succession to perform an implicit UNASSIGN. To change a line from an assigned line to a login line, you must first issue the UNASSIGN command.

When assigning or unassigning asynchronous lines, the parameters used in the command line are the same as those used with the AMLC command.

The LOOK Command

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 (the Ring 0 stack segment for the user).
access	Access rights to be granted (as in the SHARE command). The default is '200 (read only).
mapseg	Segment of user 1's address space into which the specified segment is to be mapped. The default is '4001.

Caution

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, PREPAG, RWLOCK, and VPSD configuration directives are not obsolete, but their use should be avoided. These directives are discussed next.

▶ FILUNT

The FILUNT directive (also known as per-user file units) defines the maximum number of file units available to each user. The default value for FILUNT is '77772 (32,762 decimal). You should omit this directive from the configuration file and use the default number of file units.

▶ PREPAG

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 to PREPAG 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

The default file system read/write lock is set to allow _n_ readers or 1 writer. You can use the RWLOCK directive to change this setting. Specifying a value of 0 to RWLOCK allows 1 reader or 1 writer (the writer has exclusive control). Specifying a value of 3 allows _n_ readers and 1 writer.

You should retain the default setting because a number of utilities and subsystems do not work under the other settings. If it is necessary to change the read/write lock of any file or set of files, use the PRIMOS RWLOCK command rather than the RWLOCK directive.

▶ VPSD

Prime's assembly language debugger (also known as the kernal debugger) can be wired into PRIMOS at system startup with the VPSD directive. This debugger is useful for debugging the operating system but not for specifying 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 NET and NUSEG directives do not work for Rev. 19.4 systems.

Note

> At Rev. 20, the ASYNC JUMPER directive is a synonym for the ICS JUMPER directive and the SYNC directives are synonyms for the SMLC directives. For details, see Chapter 10, CONFIGURATION DIRECTIVES.

▶ NET ON

As of PRIMOS Rev. 19.3, the NET directive no longer starts up the PRIMENET network. Use the START_NET command to start up PRIMENET. For details on START_NET, see the PRIMENET Guide.

If the NET ON command is included in a Rev. 20.2 configuration file, the following warning message is displayed during cold start:

```
The NET config directive no longer enables the network.  Please
use the START_NET command.  (CINIT)
```

Cold start continues after the message is displayed.

▶ NUSEG number

Prior to PRIMOS Rev. 19.4 the NUSEG directive set the size of the virtual address space for each user. Since Rev. 19.4, the System Administrator uses EDIT_PROFILE to set the number of static and dynamic segments for each user.

If the NUSEG directive is in the configuration file at cold start, the following warning message is displayed:

```
NUSEG HAS BEEN REPLACED BY EDIT PROFILE SUBCOMMANDS.  (CINIT)
```

Cold start continues after the message is displayed.

E

Determining Physical Line Numbers

This appendix describes the procedure for tracing your asynchronous communication lines back to the controller. It explains how to calculate a physical line number from the controller's device address, offset, and jack number.

DETERMINING LINE NUMBERS

Use the following procedure to find the line number for a particular asynchronous line:

1. Use the STATUS COMM command to determine the device addresses for all the AMLC and ICS controller boards in the system.

2. Trace the target line to the controller that it is plugged into (the target controller). The line is connected to a cable assembly, which consists of four cables. The four cables terminate at one end in a single connector, marked C1. Each of the other ends has a nine-pin connector labeled J1 to J4. The cable assembly, in turn, is plugged into one of one, two, or four ports on the target controller.

3. Determine the type of the target controller (AMLC, ICS1, ICS2, or ICS3).

4. Find out the device address of the target controller. Examples are 10, '36, and '54.

5. Determine which port the C1 cable connector is plugged into.

6. Find out the jack number into which the target line is plugged. The four cables on a given cable assembly are labeled J1, J2, J3, and J4. The labels are located near the point at which the target line connects to the cable.

7. Use one of the two procedures described below to determine the target line number. For ICS controllers, see the section entitled Lines Attached to ICS Controllers. For AMLC controllers, see the section entitled Lines Attached to AMLC Controllers.

Lines Attached to ICS Controllers

To determine the line number for a line connected to an ICS controller, use the following procedure:

1. Determine the offset of the first ICS controller on the system.

2. Determine the offset of the target controller.

3. Determine the offset of the line on the target controller.

4. Add the three numbers obtained from Steps 1-3. The line number is the sum of these three numbers.

This procedure is explained in the following paragraphs. For more information on ICS controllers, see the ICS User's Guide.

Finding the Offset of the First ICS Controller: AMLC controllers are always configured before ICS controllers. Therefore, the offset of the first ICS controller depends on whether the system has AMLC controllers.

- If the system has no AMLC boards, the offset of the first ICS controller is 0 (zero).

- If the system has AMLCs, the offset of the first ICS controller is the highest available AMLC line number rounded up to the next modulus 16 boundary. The offset is always represented in octal, is always evenly divisible by 16 (20 octal), and always ends in 0. For example, octal offsets could be 0, 20, 40, 60, and 100.

Finding the Offset of the Target Controller: The offset of the target ICS controller depends on the configuration of the lines and controllers. However, with AMLC controllers, the offset depends only on the device address.

Use the STATUS COMM command to find the device addresses of all the ICS controllers. The address priority for all ICS controllers is '10, '11, '36, and '37. ICS2 and ICS3 controllers are usually assigned a device address of '10 or '11 by the manufacturer. When you have ICS2 or ICS3 controllers as well as an ICS1, the ICS1 controller has a lower priority and will be assigned a device address of '36 or '37. If you have only ICS1 controllers, all device addresses ('36,'37, '10, or '11) are valid.

Check off the device addresses of the ICS controllers on the list in Table E-1. Do not, however, check off AMLC device addresses on the list.

The ICS controller indicated by the first check in the list has a target offset of '0. The target offset of an additional ICS controller depends on the number of lines configured for those above it on the list.

Number of Lines Per Controller: The number of lines on an ICS controller depends on the model.

An ICS1 controller is always configured for eight asynchronous lines.

Use the following procedure to determine how many lines are configured for ICS2 and ICS3 controllers:

1. Determine how many Line Adapter Cards (LACs) are on the controller.

2. If the number from Step 1 is odd, add one to it.

3. Multiply the result by 4. Note this number in the Number of Lines column of the controller checklist in Table E-1.

4. Add the number of lines configured for the first ICS to its offset (zero). The result gives the offset of the second ICS. If the system has a third ICS, its offset is the sum of the number of lines configured for the second ICS and the offset of the second ICS, and so on.

Table E-1
ICS Controller Checklist

Present	Address	Offset	Number of Lines
[ ]	'10	[ ]	[ ]
[ ]	'11	[ ]	[ ]
[ ]	'36	[ ]	[ ]
[ ]	'37	[ ]	[ ]
[ ]	'56	[ ]	[ ]
[ ]	'51	[ ]	[ ]
[ ]	'50	[ ]	[ ]
[ ]	'32	[ ]	[ ]
[ ]	'17	[ ]	[ ]
[ ]	'16	[ ]	[ ]
[ ]	'15	[ ]	[ ]
[ ]	'35	[ ]	[ ]
[ ]	'52	[ ]	[ ]
[ ]	'53	[ ]	[ ]
[ ]	'54	[ ]	[ ]

Finding the Line Number on the Target Controller: Determining the line number on the target controller depends on the model.

For an ICS1 controller, the line number depends on the following:

- The port to which the line is connected on the ICS1 controller
- The cable connector into which the line is plugged

An ICS1 controller has three ports. The leftmost port is for synchronous communication only, so it is not discussed here. The center and rightmost ports are the asynchronous ports. The rightmost port contains controller line numbers 0 to 3, and the middle port has board line numbers 4 to 7. (Note that this is the reverse of the AMLC board.)

Use the following list to determine the ICS1 board line number.

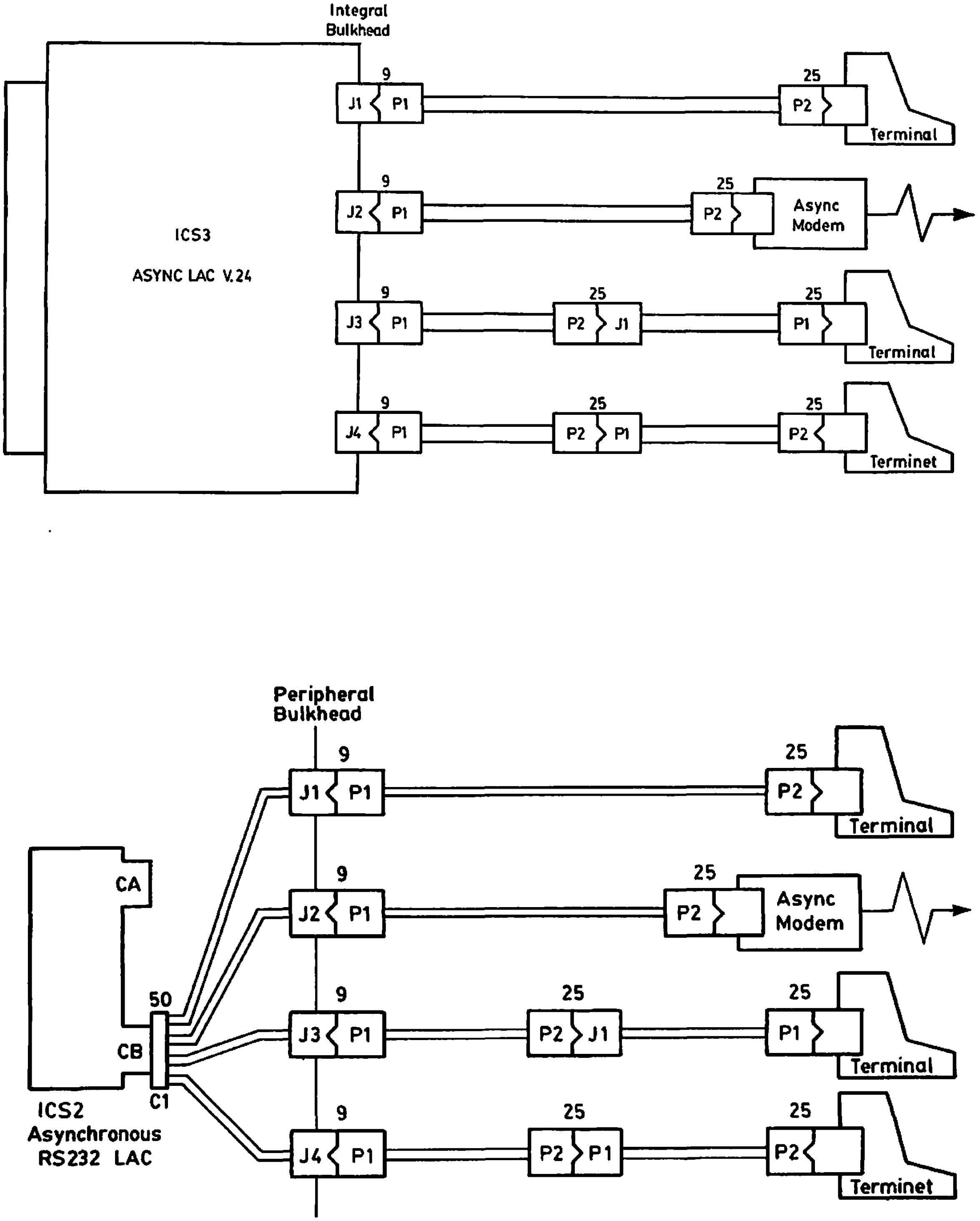
Port	J1	J2	J3	J4
rightmost	0	1	2	3
middle	4	5	6	7

For ICS2 and ICS3 controllers, the board line number depends on the following:

- The LAC to which the line is connected on the ICS2 or ICS3
- The jack number of that particular line on the LAC

ICS2 and ICS3 controllers support up to 16 LACs. Figure E-1 shows the jacks and cable connections for ICS2 and ICS3 asynchronous LACs. The LACs are numbered from right to left as viewed from the back of the ICS2. The rightmost cable leads to a buffer card. The number of other cables depends on how many LACs there are on the controller. Each of the next eight cables leads to a LAC. If the system has more than eight LACs, the next cable leads to a buffer card, and the following cables lead to LACs.

Determine how many LACs are to the right of the LAC to which the line is connected, and to which of the four jacks the line is connected. Now multiply the number of LACS by four, add the jack number, and subtract one. The result is the board line number.



Cable Connections for ICS2 and ICS3 Asynchronous LACs
Figure E-1

Calculating Line Numbers: To calculate the line number, add the ICS line offset, the target offset, and the board line number. This value is the line number in octal. You can convert this number to a decimal number with the following PRIMOS command for octal-to-decimal conversions:

```
TYPE [OCTAL octal-number]
```

Lines Attached to AMLC Controllers

A system can have a maximum of eight AMLC boards. Each board has up to four ports (C, D, E, F), proceeding from left to right as viewed from the rear of the CPU. Line numbers can be calculated with the formula given in Table E-2.

Table E-2
Determining AMLC Line Numbers

AMLC Board	Value of x =	Device Address
1	0	'54
2	1	'53
3	2	'52
4	3	'35
5	4	'15
6	5	'16
7	6	'17
8	7	'32

AMLC Port	Value of y =	Jack Number	Value of z =
C	0	J1	1
D	1	J2	2
E	2	J3	3
F	3	J4	4

To determine:	Use the default formula:
1) Physical line number of AMLC	16(x) + 4(y) + (z-1) = n
2) SET_ASYNC command line argument	Convert the physical line number n calculated above to decimal with the PRIMOS command TYPE [OCTAL n]

Example of Line Number Calculation

Suppose a system has the following controllers:

- One AMLC controller
- One ICS2 controller, at device address '10, containing seven Line Adapter Cards
- Two ICS1 controllers, at device addresses '11 and '36

To determine the number of the line connected to cable connector J3 on the middle port of the ICS1 at device address '36, proceed as follows:

1. Find the offset of the first ICS controller. Because the system has only one AMLC, this offset is '20.
2. Use the STATUS COMM command to check the device addresses of the ICS controllers, and check these off on the ICS Controller Checklist. (See the sample under Step 5 below.)
3. Determine the number of lines that are allocated for the ICS2. Because there are 7 LACs, add 1 and multiply by 4. The result is 32 in decimal. Converted to octal, the answer is '40.
4. Find the offset of the ICS1 at device address '11. This is computed by adding the offset of the first ICS (0) to the number of lines, '40.
5. Find the offset of the ICS1 at device address '36. This is the sum of the offset of the second ICS ('40) and the number of asynchronous lines ('10). (An ICS1 is always configured for '10 lines.) Fill in the appropriate information on your checklist as shown in the following example.

Completed Sample ICS Controller Checklist

Present	Address	Offset	Number of Lines
[x]	'10	[' 0]	['40]
[x]	'11	['40]	['10]
[x]	'36	['50]	['10]

6. Find the board line number. In the preceding list of ICS1 board line numbers, the line connected to J3 on the middle port is line number 6.

7. Add the ICS line offset from Step 1 ('20), the offset of the target ICS1 from Step 4 ('50), and the board line number from Step 6 (6). The result is '76.

8. Use the PRIMOS command TYPE [OCTAL 76] to convert the answer to decimal and display it on the screen. The result is 62. The line number is therefore 62.

INDEX

Index

SURVEY

READER RESPONSE FORM

DOC5037-4LA System Administrator's Guide Fourth Edition

Your feedback will help us continue to improve the quality, accuracy, and organization of our user publications.

1. How do you rate the document for overall usefulness?

   ___excellent ___very good ___good ___fair ___poor

2. Please rate the document in the following areas:

   Readability: ___hard to understand ___average ___very clear

   Technical level: ___too simple ___about right ___too technical

   Technical accuracy: ___poor ___average ___very good

   Examples: ___too many ___about right ___too few

   Illustrations: ___too many ___about right ___too few

3. What features did you find most useful? ______________________________

   ____

   ____

   ____

4. What faults or errors gave you problems? ______________________________

   ____

   ____

   ____

Name: ______________________________ Position: ____________________

Company: ____

Address: ____

____ Zip: ____________

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

First Class Permit #531 Natick, Massachusetts 01760

BUSINESS REPLY MAIL

Postage will be paid by:

Prime™

Attention: Technical Publications
Bldg 10
Prime Park, Natick, Ma. 01760

READER RESPONSE FORM

DOC5037-4LA   System Administrator's Guide   Fourth Edition

Your feedback will help us continue to improve the quality, accuracy, and organization of our user publications.

1. How do you rate the document for overall usefulness?

   ___excellent ___very good ___good ___fair ___poor

2. Please rate the document in the following areas:

   Readability: ___hard to understand ___average ___very clear

   Technical level: ___too simple ___about right ___too technical

   Technical accuracy: ___poor ___average ___very good

   Examples: ___too many ___about right ___too few

   Illustrations: ___too many ___about right ___too few

3. What features did you find most useful? ______________________________

   ____

   ____

   ____

4. What faults or errors gave you problems? ______________________________

   ____

   ____

   ____

Name: ______________________________ Position: ____________________

Company: ____

Address: ____

____Zip: ____________

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

First Class Permit #531 Natick, Massachusetts 01760

BUSINESS REPLY MAIL

Postage will be paid by:

Prime™

Attention: Technical Publications
Bldg 10
Prime Park, Natick, Ma. 01760

READER RESPONSE FORM

DOC5037-4LA   System Administrator's Guide   Fourth Edition

Your feedback will help us continue to improve the quality, accuracy, and organization of our user publications.

1. How do you rate the document for overall usefulness?

   ___excellent   ___very good   ___good   ___fair   ___poor

2. Please rate the document in the following areas:

   Readability: ___hard to understand   ___average   ___very clear

   Technical level: ___too simple   ___about right   ___too technical

   Technical accuracy: ___poor   ___average   ___very good

   Examples: ___too many   ___about right   ___too few

   Illustrations: ___too many   ___about right   ___too few

3. What features did you find most useful? ______________________________

   ____

   ____

   ____

4. What faults or errors gave you problems? ______________________________

   ____

   ____

   ____

Name: ______________________ Position: ____________________

Company: ____

Address: ____

____Zip: __________