Prime®

System Administrator's Guide Volume III System Access and Security

Revision 22.0

DOC10133-2LA

System Administrator's Guide
Volume III
System Access and Security

Second Edition

Dick Frost

This guide documents the software operation of the Prime Computer and its supporting systems and utilities as implemented at Master Disk Revision Level 22.0 (Rev. 22.0).

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

Printing History

First Edition (DOC10133-1LA) July 1987 for Revision 21.0
First Update (UPD10133-11A) October 1987 for Revision 21.0.1
Second Edition (DOC10133-2LA) October 1988 for Revision 22.0

Credits

Editorial: Roberta King
Project Support: Alan Dossett, John Seybold
Graphic Support: Mingling Chang
Illustration: Terry Bacharz
Production: Judy Gordon

How to Order Technical Documents

To order copies of documents, or to obtain a catalog and price list:

United States Customers	*International*
Call Prime Telemarketing, toll free, at 1-800-343-2533, Monday through Friday, 8:30 a.m. to 5:00 p.m. (EST).	Contact your local Prime subsidiary or distributor.

Customer Support

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

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

For other locations, contact your Prime representative.

Surveys and Correspondence

Please comment on this manual using the Reader Response Form provided in the back of this book. Address any additional comments on this or other Prime documents to:

Technical Publications Department
Prime Computer, Inc.
500 Old Connecticut Path
Framingham, MA 01701

CONTENTS

APPENDICES

ABOUT THIS BOOK

The *System Administrator's Guide, Volume III: System Access and Security* is intended to help a System Administrator

- Plan and establish proper maintenance and security procedures for equipment, media storage, and the computer room environment
- Plan and establish user environments and project environments on the system
- Plan and establish the proper access controls for the use of the system, subsystems, and peripheral devices
- Help users and operators deal with unexpected problems
- Use and tune the Security Audit facility, a separately priced facility for systems running at a C2-certified level of security

If you have administrative responsibility for a Prime system, this book is intended for you. The System Administrator may delegate many security responsibilities to a Security Administrator, but the overall responsibility for the system belongs to the System Administrator alone. The System Administrator may also designate a Project Administrator for one or more projects. All such administrative responsibilities are described here. Other administrative functions are described in

- *System Administrator's Guide, Volume I: System Configuration* (DOC10131-2LA)
- *System Administrator's Guide, Volume II: Communication Lines and Controllers* (DOC10132-2LA)

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

RELATED DOCUMENTATION

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

- *Rev. 22.0 Software Installation Guide* (IDR10176-2LA) describes how to install the PRIMOS Rev. 22.0 software, both for initial installations and for revision update installations.
- *Operator's System Overview* (DOC9298-3LA) introduces the series of operator's guides and describes computer-room operation of Prime systems.
- *Operator's Guide to System Monitoring* (DOC9299-3LA) describes how to monitor system activity and respond to system and user messages.
- *Operator's Guide to File System Maintenance* (DOC9300-4LA) 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.
- *Data Backup and Recovery Guide* (DOC10129-1LA) and its update UPD10129-11A describe how to save information on disk or tape and how to restore that information later.
- *Operator's Guide to the Batch Subsystem* (DOC9302-3LA) describes how to set up, monitor, and control the Batch subsystem.
- *Operator's Guide to System Commands* (DOC9304-4LA) details the commands used by system operators.
- *Operator's Guide to the Spooler Subsystem* (DOC9303-3LA) describes how to set up, monitor, and control the Spooler subsystem.
- *PRIMOS Commands Reference Guide* (DOC3108-6LA) is a detailed reference of user commands.
- *DSM User's Guide* (DOC10061-2LA) describes how to set up, modify, and manage the utilities provided by Distributed Systems Management.
- Network documentation available at Rev. 22.0:
  - *PRIMENET Planning and Configuration Guide* (DOC7532-4LA)
  - *Programmer's Guide to Prime Networks* (DOC10113-1LA) and its update UPD10113-11A
  - *Operator's Guide to Prime Networks* (DOC10114-1LA) and its update UPD10114-11A
  - *User's Guide to Prime Network Services* (DOC10115-1LA)
  - *NTS Planning and Configuration Guide* (DOC10159-1LA) and its update UPD10159-11A
  - *NTS User's Guide* (DOC10117-2LA)

- *ICS User's Guide* (DOC10094-1LA) and its update UPD10094-11A provide detailed information on Prime's Model 2 (ICS2) and Model 3 (ICS3) Intelligent Communications Subsystems.
- Your CPU handbook, for example, the *6550 Handbook* (DOC10161-2LA) and the *4150 Handbook* (DOC10162-2LA).
- *Site Preparation Guide* (DOC5029-2LA) and its update UPD5029-21A provide information for preparing and maintaining a system site.
- *PRIMAN User's Guide* (DOC10157-2LA) explains how to use PRIMAN™ for system monitoring.

A complete list of Prime technical documentation is available online by typing the command HELP DOCUMENTS or in hardcopy through the *Guide to Prime User Documents* (DOC6138-6PA). This guide is issued once a year. Lists of additional updated material are published quarterly in the Customer Service Newsletter.

USING THIS BOOK

The *System Administrator's Guide, Volume III: System Access and Security* documents all the security features available on the PRIMOS operating system, including the new features available at Rev. 22.0. Some features (for security audits) require separately priced software and documentation. The Security Audit facility is a requirement to maintain the system at a security level of C2 certification, as specified by the *Department of Defense Trusted Computer System Evaluation Criteria.*

One such DoD requirement is to provide a "Trusted Facilities Manual" for administrators of a system maintaining a C2-certified level of security. Such a manual must describe all security features on the system. This volume satisfies that requirement.

The *System Administrator's Guide, Volume III: System Access and Security* has three major parts to help you plan, establish, and maintain system access and security.

Part I, Introduction, briefly outlines the issues affecting system access and security.

Part II, Physical Security, describes the need for careful attention to environmental factors and for orderly procedures to maintain the security of terminals, peripherals, and storage media.

Part III, Online Access and Security, describes offline logs and online logs, audits, profiles, and commands that enable the administrator to maintain the security of online information. It provides separate directions for both C2 and non-C2 installations, wherever separate procedures are required.

Appendices provide guidelines and samples for login and logout programs, sample EDIT_PROFILE messages, and record formats for security audits.

If your site must use the fully C2-certified version of PRIMOS formally evaluated by the Department of Defense, then you must use Revision 21.0.1 of PRIMOS. Other sites that do not need to adhere to the strict C2 certification norms will find that Revision 22.0 of PRIMOS can provide all the previous C2 security enhancements plus some additional access and security enhancements.

ACCESS AND SECURITY ENHANCEMENTS AT REV. 22.0

At Rev. 22.0, there are several enhancements to access and security.

The changes generally affect passwords — their generation, duration, length, and format. See Chapter 6 for descriptions and examples of the following new functionality in EDIT_PROFILE:

- COMPUTER_GENERATED_PASSWORD subcommand
- DEFAULT_PASSWORD_LIFETIME subcommand
- MAXIMUM_PASSWORD_LENGTH subcommand
- VERIFY_PASSWORD_FORMAT subcommand
- -PASSWORD_LIFETIME option of ADD_USER subcommand
- -PASSWORD_LIFETIME option of CHANGE_USER subcommand

Device ACLs now may be set on a maximum of 960 assigned lines, whether locally assigned or NTS-assigned. The numbering assignments for the local lines must be within the range 0 through 512. The numbering assignments for the NTS lines must be within the range 1025 through 1536. You must customize the number of assigned lines to the number of processes supported by your system.

See Chapter 5 for details on AL*n* subdirectories; these subdirectories are now numbered in decimal.

See Chapter 4 for the implementation of these features in planning your system. Also see Chapter 7 for examples of user logins when password lifetimes have expired, with computer-generated passwords either enabled or disabled.

PRIME DOCUMENTATION CONVENTIONS

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

Convention	*Explanation*	*Example*
UPPERCASE	In command formats, words in uppercase bold indicate the names of commands, options, statements, and keywords. Enter them in either uppercase or lowercase.	**SLIST**
italic	In command formats, words in lowercase bold italic indicate variables for which you must substitute a suitable value. In text and in messages, variables are in non-bold lowercase italic.	**LOGIN** ***user-id*** Supply a value for *x* between 1 and 10.
Abbreviations in format statements	If a command or option has an abbreviation, the abbreviation is placed immediately below the full form.	**SET_QUOTA** **SQ**
Brackets [ ]	Brackets enclose a list of one or more optional items. Choose none, one, or several of these items.	**LD** [**-BRIEF** **-SIZE**]
Braces { }	Braces enclose a list of items. Choose one and only one of these items.	**CLOSE** {***filename*** **-ALL**}
Braces within brackets [{ }]	Braces within brackets enclose a list of items. Choose either none or only one of these items; do not choose more than one.	**BIND** [{***pathname*** ***options***}]
Parentheses ( )	In command or statement formats, you must enter parentheses exactly as shown.	**DIM** ***array*** **(*row, col*)**
Underscore in examples	In examples, user input is underscored but system prompts and output are not.	OK, RESUME MY_PROG This is the output of MY_PROG.CPL OK,
Ellipsis ...	An ellipsis indicates that you have the option of entering several items of the same kind on the command line.	**SHUTDN** ***pdev-1*** [***...pdev-n***]
Hyphen -	Wherever a hyphen appears as the first character of an option, it is a required part of that option.	**SPOOL -LIST**

Convention	*Explanation*	*Example*
Subscript	A subscript after a number indicates that the number is not in base 10. For example, a subscript 8 is used for octal numbers.	200_8
Key symbol	In examples and text, the name of a key enclosed by a rectangle indicates that you press that key.	Press Return

PART I
INTRODUCTION

1

SYSTEM ACCESS AND SECURITY ISSUES

As System Administrator you have overseen the proper installation and configuration of system hardware and software. Now you must add users to the system.

You must ensure that your user community has adequate and proper access to the system by providing them with clearly defined rights for using the system. You must also provide security for your users, so that no user abuses another's rights by taking, modifying, or destroying data. Finally, you must protect and maintain the system: the hardware, the software, the storage media, and the environment.

The chapters of this volume, summarized below, provide information and procedures for handling administrative responsibilities that involve system security and access. As System Administrator you may delegate many of these responsibilities to Security Administrators, Project Administrators, and operators. The section of this chapter entitled Administrative Responsibilities indicates the chapters these administrators should master in order to carry out their respective responsibilities.

SYSTEM ACCESS AND SECURITY OVERVIEW

Part II of this guide describes administrative responsibilities for maintaining physical security. It contains two chapters.

Chapter 2, Equipment and Environment, contains guidelines for maintaining an orderly machine room, describes how to handle and store disks and tapes, and explains how to be prepared for emergencies in the machine room.

Chapter 3, Backups, provides guidelines for scheduling backup operations, so that little or no information is lost through an accident or mistake.

Part III describes the System Administrator's role in providing online access and security. It contains eight chapters.

Chapter 4, Planning the User Environment, describes the factors a System Administrator must consider before adding a user to the system. Guidelines and examples indicate how to add a user to one or more projects, and how to define user profiles. Chapter 4 begins the description of Access Control Lists (ACLs), and describes the assignment of rights to a single user or a collection of users (an ACL group).

Chapter 5, Setting Access Rights, looks at ACLs and ACL groups as applied to the directory structure of PRIMOS and its subsystems. The chapter describes how to protect both system directories and user directories. It discusses special types of ACLs called priority ACLs and device ACLs. It also discusses other directory structure features, including the PRIMOS Search Rules facility, and the (now obsolete) use of passwords on directories.

Chapter 6, Using EDIT_PROFILE, explains how the System Administrator adds user IDs to the system. Each user ID is assigned the ACLs associated with a unique user profile and one or more project profiles. EDIT_PROFILE creates or modifies the environment for each system user.

Chapter 7, Security, describes security procedures for protecting the system itself as well as all the users granted access to it. This chapter emphasizes the protection of privileged information on the system.

Chapter 8, Adding Subsystems, provides information on other facilities you may now be interested in adding to your system.

Chapter 9, Looking After Users, provides procedures to solve user access problems before such problems become security issues.

Chapter 10, System Monitoring, provides procedural guidelines for maintaining the system hardware and software. It describes an online utility to track unusual system events and describes other commands to monitor the performance and usage of the system.

Chapter 11, Security Audits, describes how to use a separately priced Security Audit facility containing an Audit Collection facility that may be tuned to create audit trails of various actions or users on the system. It may track a particular event, particular results of events, or the actions of one or more users. The Security Audit facility also contains an Audit Reporting facility, an Audit File Backup facility, and a Crash Audit Recovery facility. The Security Audit facility is a requirement for every Prime system that is to maintain security at a C2-certified level, as specified by the Department of Defense.

Four appendices provide supplemental access and security information. Appendix A gives information and examples concerning external login and logout programs. Appendix B lists EDIT_PROFILE messages, each with a brief explanation of reasons for the message. Appendix C gives a list of record fields for security audits. Appendix D gives information on numbered semaphore ACLs, which are required on systems that must maintain a strict C2-certified level of security.

ADMINISTRATIVE RESPONSIBILITIES

The System Administrator may delegate some responsibilities to other individuals, such as

- Security Administrators
- Project Administrators
- Network Administrators
- System operators

Information for New System Administrators

If your system is newly installed, you first need the information in Chapters 4, 5, 6, and 7 to initialize an active and secure user environment.

Information for Security Administrators

A Security Administrator who is assigned maintenance and security responsibilities for equipment, environment, and storage facilities must master the information in Chapters 2, 3, 7, and 10. The System Administrator may assign only a subset of these responsibilities; nevertheless, the Security Administrator must know all of them.

A Security Administrator who is assigned responsibility for maintaining online security must master the information in Chapters 5, 7, 10, and 11. A Security Administrator acts as the System Administrator when carrying out responsibilities for online security.

Information for Project Administrators

A Project Administrator must master the information in Chapters 4, 5, and 6 to carry out the responsibilities of project management. The Project Administrator controls system access to assigned projects. The System Administrator first adds users to the system and creates the ACL groups for that project profile, but thereafter the Project Administrator controls access. The Project Administrator must know all aspects of EDIT_PROFILE and ACLs.

Information for Network Administrators

If a system is a node on a network, then the System Administrator shares some system control with a Network Administrator. Administrators of systems on the network therefore must be aware of the responsibilities of a Network Administrator. Refer to About This Book for a list of manuals that provide information on Prime networks and Distributed Systems Management (DSM).

Information for Operators

The daily operations and maintenance of the system require an operator to use the supervisor terminal. The System Administrator must therefore have complete trust in the system operator, and must make the operator aware of the need for security. The System Administrator should encourage capable operators to learn more about the system. Limiting their information about certain procedures does not guarantee security, but it does increase the probability of an accidental security or access problem.

PART II
PHYSICAL SECURITY

2

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, and the subject of this chapter, is the maintenance of the system environment and hardware.

Other maintenance tasks described elsewhere in this book include controlling interactions between users and the system (Chapters 4, 5, 6, 7, and 9), setting schedules for backups (Chapter 3), integrating and maintaining subsystems on the system (Chapter 8), monitoring system usage (Chapter 10), and maintaining an optional Security Audit facility (Chapter 11).

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 and conditions under which the hardware is 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

An important procedure for every user and operator is to maintain the confidentiality of passwords. If a password is written on the wall next to the terminal or stored in the top drawer of the desk, that password is not secure. The System Administrator should remind users about this fundamental security measure and should recommend that users change their passwords periodically.

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. A general user who is unfamiliar with machine room procedures may cause severe problems; a general user who knows the procedures still makes little or no contribution by being in the machine room. A machine room open to general users has no system security. 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. Above all, you must have operators whom you can trust with the security 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.

Supervisor Terminal and the RESUS Command: The Distributed Systems Management (DSM) facility provides the System Administrator the RESUS command to make a user terminal function as a remote supervisor terminal. While RESUS is active, the physical supervisor terminal functions solely as a printer. It merely echoes the activity of the RESUS-activated user terminal that now is associated with the User 1 process. The RESUS command is extremely powerful, requiring care in its use and the utmost security precautions.

If both original and RESUS-activated supervisor terminals are not physically identical video display terminals, then no EMACS-based activity may be performed. A RESUS-activated terminal left unattended offers complete system access to anyone with access to the terminal. The System Administrator must maintain tight security and time controls on a RESUS-activated terminal. Refer to the *DSM User's Guide* for further information on RESUS and other DSM-related commands.

User Terminals and Data Security: The System Administrator should periodically remind all users not to leave terminals unattended while they are logged in. Terminals used by the System Administrator and system operators require special attention to security measures, since these terminals provide privileged access to the system.

Other Precautions: Other procedures you may want to set up are

- Rules about who may use the machine room and such peripheral devices 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 make 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. Removable disks, in particular, are fragile. Improper handling may damage the disk, which may cause a head crash when that disk is used. Tapes and disks that have been used on the system hold information that should be either erased or protected. Store this media in a secure area.

Handle disks with care. Disks should not be carried in large piles because they 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, do not use it 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 logbook in the storage area. The logbook 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 logbook. When a tape or disk is taken from storage, make an entry in the logbook 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 provides information on the location of all storage media.

Storage media that contain confidential information should be kept in a special secure area. This area may be a locked strongbox, 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 proper 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:

- Prohibit smoking, food, and beverages in the machine room. There should be no exceptions to this rule.
- Keep the machine room free of dust and other contaminants.
- Maintain the machine room environment within the temperature and humidity specifications provided by your Customer Support Center.
- 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 for 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, with recording surfaces that 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 the section entitled System Halts. Measures to prevent data loss are discussed in Chapter 3, Backups.

Paper dust from a printer can be a major source of airborne dust. Your printers should be vacuumed at least once a day by your servicing agency or by your own personnel. As an alternative, you can put the printers in a separate 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%.

Provide preventive maintenance and service to air-conditioning systems every six months or within the manufacturer's suggested time period (whichever is shorter).

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 block 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 do serious damage 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 the system crashes repeatedly, 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 be 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, always maintain an up-to-date system logbook; be sure to perform tape dumps if you have frequent system crashes.

If system crashes occur 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 logbook 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 Support representative to discuss methods of solving 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 may 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 choose between 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 of 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 be cold started after a halt. The only exceptions are systems that have the optional Model 7300 battery backup power supply; these systems can be warmstarted after a halt caused by a power outage.

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 would also be useful if an employee were to have a heart attack.)

Alert all personnel to the danger of electric shock. Under no circumstances should they touch any internal components of the system.

3

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 *Data Backup and Recovery Guide.*

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 (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 database 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. Consider the following questions 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 you can use to answer 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 to disk, perform backups under PRIMOS, not PRIMOS II. From Rev. 20.0 onward, PRIMOS II cannot write on disks and cannot save information for access control or quotas.

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 less than one hour. Larger fixed-media disks are faster, but require more time because of the extra size.

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 770-megabyte disk may require up to 24 tapes.

You can use a disk backup without performing 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 disk backups must be weighed against their disadvantages. Disks are expensive and require special handling and storage because they have lower tolerances than tape 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 30 minutes at 6250 bpi. The restoration of the backup to disk, however, (which is rarely needed) also takes about 30 minutes, which means a total of about one hour for the backup and restoration procedures. PHYSAV physically copies the entire disk; it cannot perform incremental backups.

The three utilities for making disk-to-tape backups are MAGSAV, BACKUP, and PHYSAV. These utilities are discussed in the following paragraphs.

The MAGSAV Utility: The MAGSAV utility copies data file by file. The MAGRST utility restores data from a MAGSAV tape.

Rev. 20 or later versions of MAGRST and MAGSAV work with Rev. 19 versions as follows:

- Rev. 20 or later MAGRST reads Rev. 19 MAGSAV tapes.
- Rev. 19 MAGRST reads Rev. 20 or later MAGSAV tapes if the tapes were created with a post-Rev. 19 version of MAGSAV using the -REV19 option.
- Rev. 20 or later MAGSAV writes the Rev. 20 or later new system boot on MAGSAV tapes only if the -REV19 option is not used.

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 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 Date/Time Accessed attribute.

If you use BACKUP, create a system ACL group named .BACKUP$ and assign to it any user 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 PHYSAV Utility: Unlike BACKUP and MAGSAV, the PHYSAV utility makes an exact physical copy of the disk contents as the contents appear on the disk. You cannot use a PHYSAV tape 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 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, directory, 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. You can make incremental backups 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. It is sometimes slower to restore a complete database from them, however, because each increment must be reloaded separately.

Incremental backups may supplement full backups. For example, you can use incremental backups 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.

You should not keep the latest (most recent) backup disk or tape in exactly the same place as the originating data, but the tape should be easily and quickly accessible. This is the copy that restores data to the system in the form that requires the least updating.

You should keep the intermediate copy 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.

You should keep the oldest copy off site, preferably in a different building. The off-site copy is the copy that is least likely to be needed.

DATA ARCHIVES

You can also use backups 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 use your normal backups as archives as well as for security against data loss. In this case, you are essentially archiving your entire file system 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 three factors:

- The volatility of data on your system
- The degree of protection you need for active data
- Resource considerations for your 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 if a disk crash occurs, 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 type of media undergoing a backup. Disk-to-disk backups are faster than disk-to-tape backups.
- The accessibility of disks undergoing backup. Only the person performing the backup should have access to the disks during the backup process.
- The amount, type, and timing of the use that your system gets. 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 that hold separate sets of data kept clearly distinct from one another.

- 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.

Backup Alternatives

A major reason for having backups is to guarantee that crucial data is not totally lost if a disk undergoes a head crash. At Rev. 21.0 mirrored partitions offer an online backup to sites that can devote two disk drives for duplicate versions of the same data. For example, a paging partition and a command device partition are duplicated on two physical disks, and mirrored partitions are activated. The two disks will be updated almost simultaneously. Should one disk drive fail, the other disk drive provides continued service without a system halt or a major loss of data.

Note that mirrored partitions do not alleviate the necessity for regular backups to offline media. The individual user will still need such offline backups to restore any file that was unintentionally deleted.

PART III
ONLINE ACCESS AND SECURITY

4

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 Database, which contains information on all users and projects on your system
- Access Control Lists (ACLs), which provide security for directories and files

The chapter also shows you how you can design your database, and gives examples of setting up different kinds of databases.

USER AND PROJECT PROFILES

The System Administrator plans the User Profile Database (as explained later in this chapter) and then creates it using the EDIT_PROFILE utility (as explained in Chapter 6, Using EDIT_PROFILE). The Administrator also uses EDIT_PROFILE to keep the database up-to-date.

The database information and profiles that are created within EDIT_PROFILE are stored in the top-level directory called the **SAD**. The SAD is the System Administration Directory, and it holds user/password verification information, project profiles, the default project profile, and so forth. The System Administrator creates the SAD when using EDIT_PROFILE for the first time.

Each user must have a user profile in the User Profile Database before that user can log in to the system. A user profile contains one set of **system attributes**. It also contains one set of **project attributes** for each project to which the user belongs. During a login session, a user maintains two sets of attributes: the user's system attributes plus the user's project attributes for the project to which the user is currently connected.

The User Profile Database 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 when EDIT_PROFILE asks if you want to create project 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 sets for that project. By appointing one or more Project Administrators, the System Administrator equitably delegates the responsibilities and the work load of administration among a number of people.

Advantages of User Profiles

Some of the advantages that user profiles provide are

- A secure method of identifying and validating users
- Administrative control over users
- An interface with the Access Control List mechanism for file system protection
- The grouping of users with similar characteristics for purposes of accounting and file system control
- The creation of a unique environment for each user

The ensuing sections describe the organization of the User Profile Database and other information to be considered in the process of adding users to the system.

One of the most important user IDs that the System Administrator may choose to add to the User Profile Database is SYSTEM. By adding SYSTEM to the User Profile Database, the System Administrator may set up a customized list of system attributes for the supervisor

terminal. If the System Administrator (SA) chooses not to do this, then PRIMOS supplies a set of default attributes for the supervisor terminal, User 1.

The user that is identified as SYSTEM in the User Profile Database must, like any other user, have at least two sets of attributes: system attributes and those attributes derived from its current project.

DEFINING USER PROFILES

The following sections describe how to define both system and project attributes within a user profile.

Supervisor Terminal 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 the 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 (IAP) 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 an entry in the SAD for SYSTEM that you are using for administrative purposes, PRIMOS uses that entry to initialize profile data for User 1. The identifier SYSTEM should be reserved for User 1, but a user with the correct password for SYSTEM may log in as SYSTEM at a user terminal. 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.
```

To maintain good security, the System Administrator should require operators and administrators to use their individual user IDs whenever possible. The System Administrator should assign operators and Project Administrators to an ACL group that provides whatever access they need for administrative functions. This assignment enables them to do administrative functions at a terminal other than the supervisor terminal. The results of their administrative tasks will be logged under their own user ID. They can

thereby help to resolve any maintenance or security problem quickly. By contrast, a login user ID reserved for administrative functions reduces accountability.

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 lets you (or any other user) run EDIT_PROFILE from the supervisor terminal.

Note

It is recommended that you enter a name other than SYSTEM when EDIT_PROFILE prompts you for the name of the System Administrator. This lets you run EDIT_PROFILE from a user terminal under a name known only to yourself.

PRIMOS has default internal values that it may use to initialize the system attributes for the User 1 process associated with the supervisor terminal. You may customize these attributes within EDIT_PROFILE by making an entry for the user SYSTEM. PRIMOS will use this profile data to initialize the User 1 process. You should limit your use of this user ID. Do not use an entry named SYSTEM in the SAD for administrative purposes. Create another identifier, different from SYSTEM, for these purposes.

When you add a user to the system (using EDIT_PROFILE's ADD_USER subcommand), 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 a maximum of 16 systemwide ACL groups (optional). ACL groups are defined in the section below, Access Control Lists.

These system attributes are stored in the system database 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 database, 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 or IAP), which is the directory to which the user is attached at login.

- Membership in a maximum of 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 as many as 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 EDIT_PROFILE's LIST_USER subcommand for JILL shows <none> as her ACL groups, but the PRIMOS command LIST_GROUP 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 (IAP), usually the top-level directory 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 IAP 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. A user can log in as a project member in four ways:

- 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 **Executable Program Formats (EPFs).** 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

- 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 less than or equal to the system limits, but they 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 Prime-supplied default values are

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

You can change these system default values with the CHANGE_SYSTEM_DEFAULTS command within EDIT_PROFILE. For example, you may install EMACS on your system, at which time you may wish to increase private dynamic segments to 100.

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 4-1 lists the minimum values, maximum values, default values, and recommended values that the System Administrator can assign to each of the four attributes of a project or user.

TABLE 4-1. Command Environment Limits

Command Attribute	*Minimum Value*	*Maximum Value*	*Default Value*	*Recommended Value*
Command levels	1	100	10	10
Live invocations per level	1	100	10	10
Private dynamic segments	16	1016	64	64
Private static segments	8	1008	64	64

Notes

The sum of a user's static and dynamic private segments cannot exceed 1024. A sum greater than 192 requires an additional page of wired memory per user.

Some separately priced Prime products require more than the default number of private dynamic segments.

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 4-1 shows the different settings you might create for the first of these command environment limits: the settings for command levels. The system has two projects and three users, and different command levels may exist for each of these – for the system, for each project, and for each user. The figure shows that a user limit is constrained by a project limit, just as a project limit is constrained by a system limit.

As the figure shows, the project limit can be the same as, but not greater than, the system limit. Note, however, that a project (such as Project-B) can have both its default and its limit greater than the system default. Similarly, a user's command levels in the project may be the same or lower than the project limit, but they cannot be greater than the project limit. Note likewise that a user limit can exceed the project default (such as User-A in Project-A).

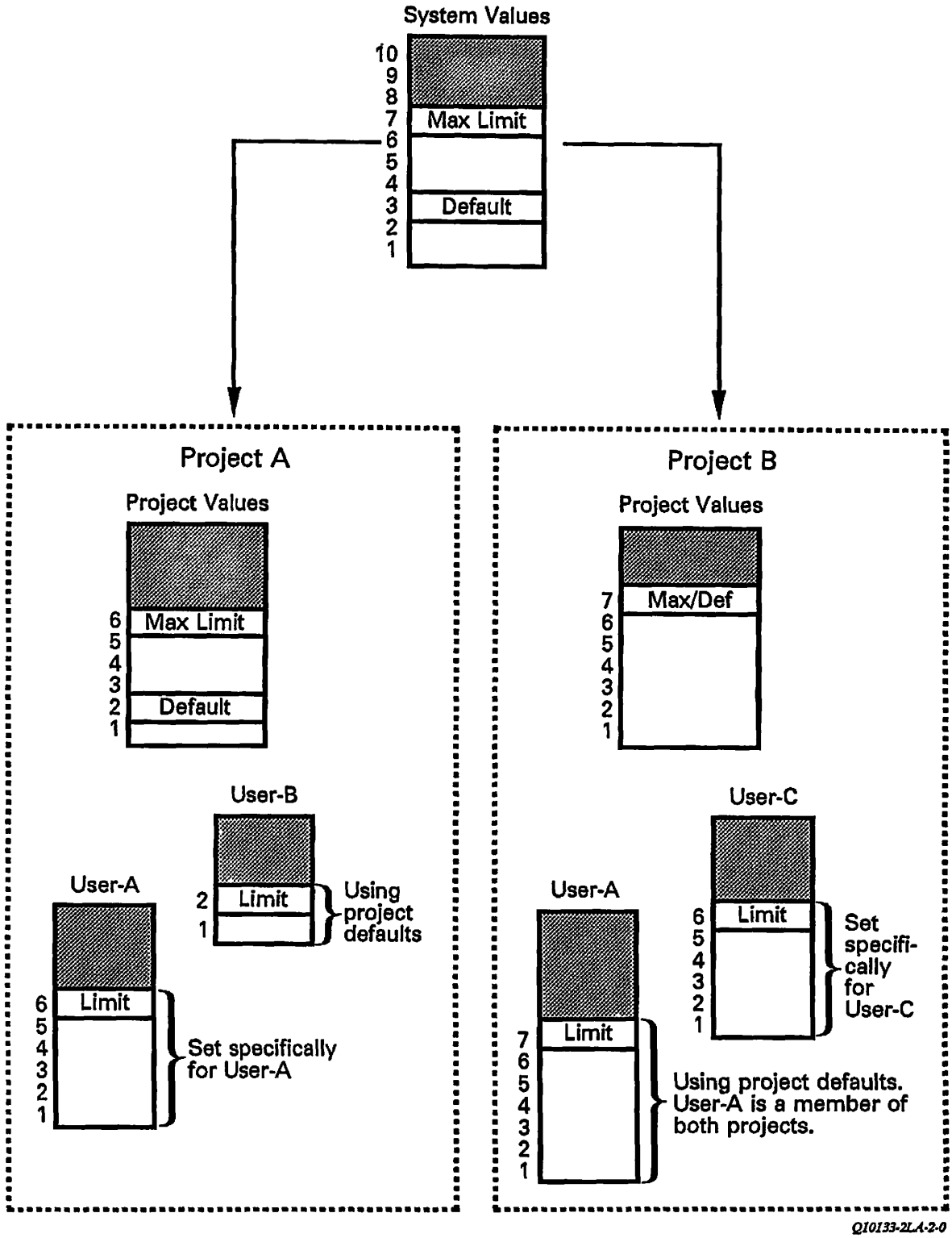

FIGURE 4-1. Command Environment: Setting Command-level Defaults and Limits

Figure 4-2 shows all the command-level settings on a different system. The system has one project and two users. User-A is using project default limits for command level and breadth. User-B has had these limits set. Note again that the project is constrained by the system limits, not by the system default. Likewise note that User-B has individual limits that are different from, but still within, the project limits. User-A, on the other hand, inherits user limits from the project default.

	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	1016	64	80	40	40	60
Private Static Segments	1008	64	80	40	40	50

FIGURE 4-2. Command Attributes: System, Project, and User

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.

A second, and optional, login program in CMDNC0 may submit the user to further validation procedures, but this program normally makes no changes in a user's environment. The user either passes the additional validation procedure or is logged out. (See Appendix A for a sample external LOGIN program.)

A third login program may also exist, and within this program the user may personally define a customized user environment. The program must be located at a user's IAP, and it must have one of these four names: LOGIN.RUN, LOGIN.SAVE, LOGIN.CPL, or LOGIN.COMI. PRIMOS looks in the user's IAP for a program so named, in this order. The user login program can perform tasks such as the following:

- Set terminal characteristics, such as the erase and kill characters

- Change the system OK, and ER! prompts
- Activate EDIT_COMMAND_LINE (ECL), the command line editor
- Activate abbreviation and/or global variable files
- Run other user-defined programs

See Chapter 7, Security, for a detailed explanation of the login procedure.

THE USER PROFILE DATABASE

The **User Profile Database** 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 database 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 database 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 database.
- The system database, 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 and references other user information such as password and password life.
- One or more project databases. Each project that you define has its own separate project database. Project databases are described in the following section.

Figure 4-3 illustrates the User Profile Database from the point of view of the System Administrator.

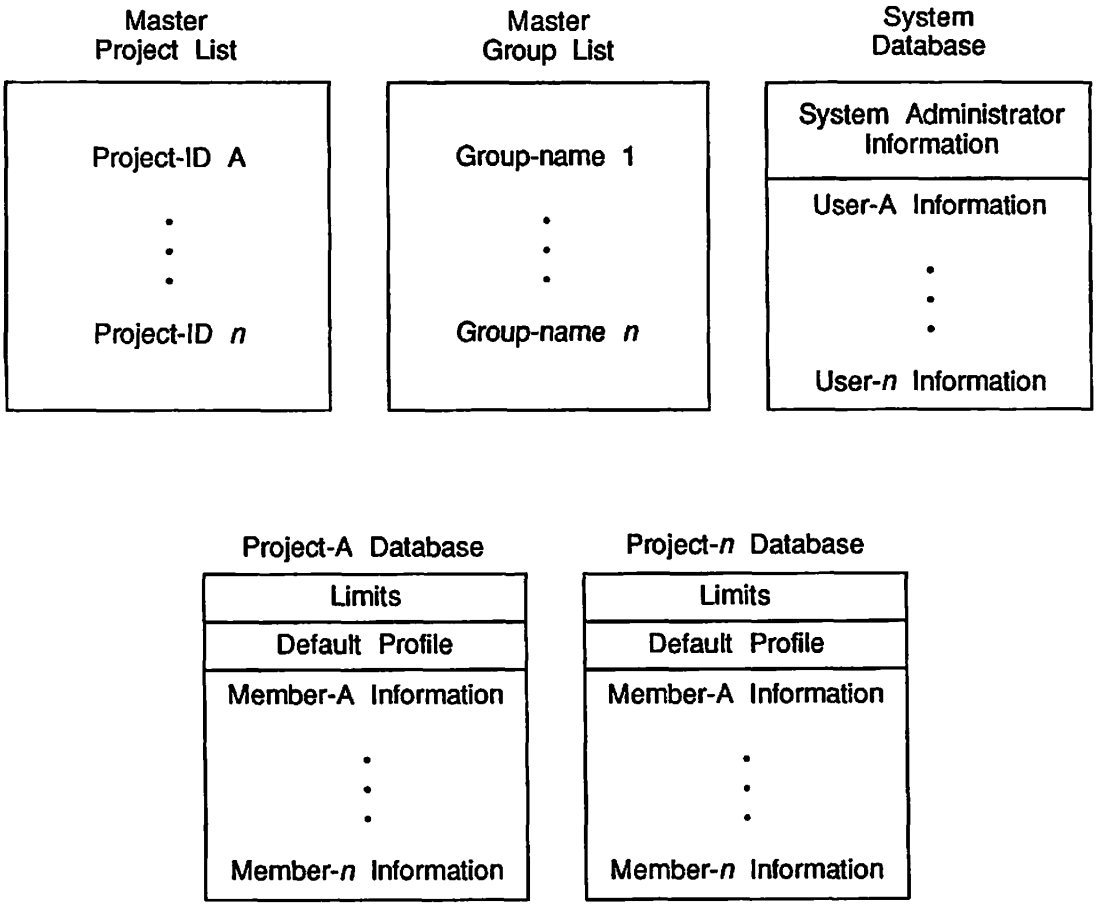

FIGURE 4-3. User Profile Database From System Administrator's Viewpoint

From the user's point of view, the database 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 probably pictures these personal attributes as isolated from those of any other system user. The user's point of view resembles that sketched in Figure 4-4.

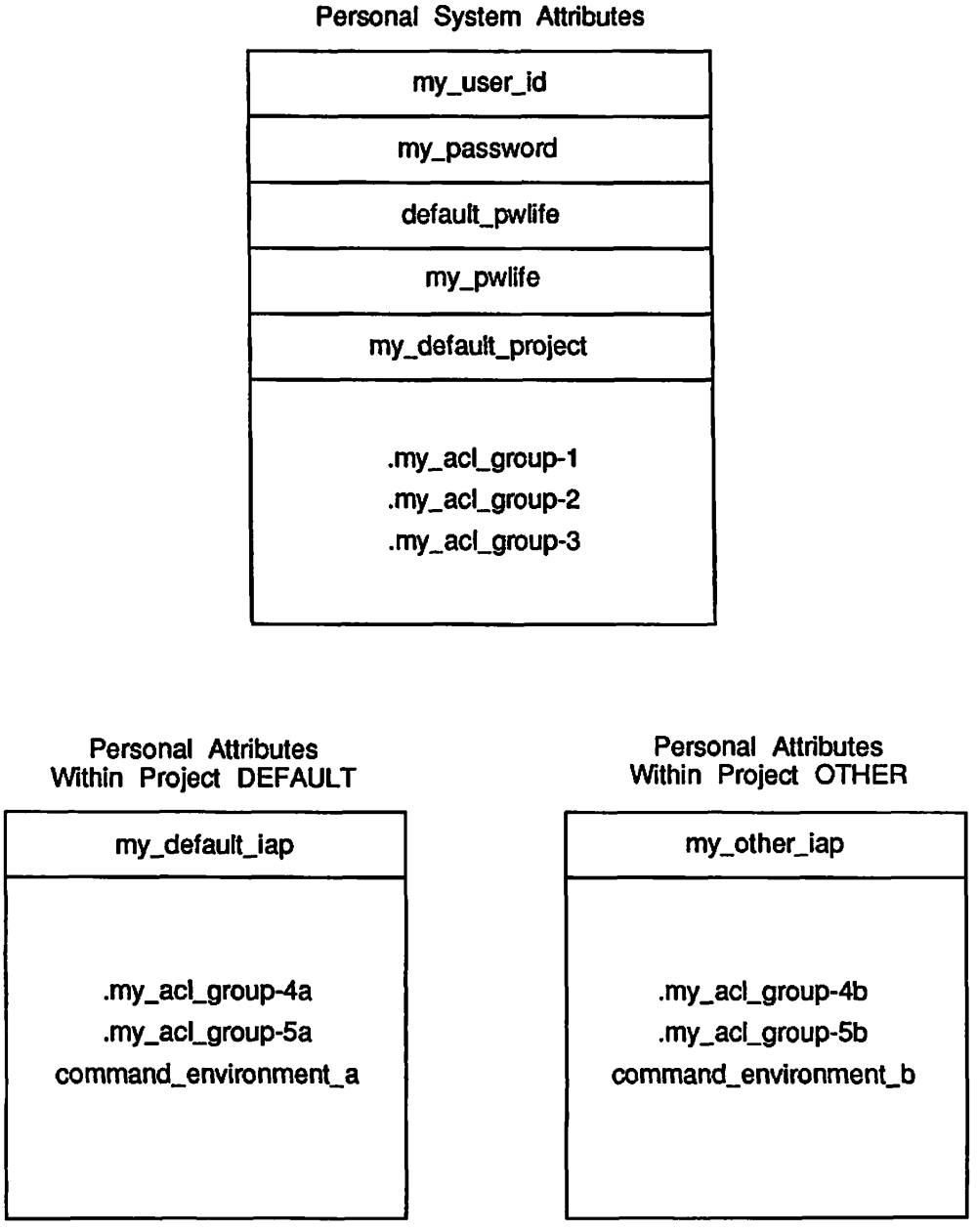

FIGURE 4-4. User Profile Database From User's Viewpoint

Project Databases

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

A project database contains the following types of material:

- The user ID of the Project Administrator. (The Project Administrator does not have to be a member of the project.)
- 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.
- The project profile. The profile consists of the default 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 default attributes are not set for the project, users who are assigned no specific command environment limits use the system defaults instead.

  Notes

  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 is top-level directory CMDNC0.

- An entry for each user who is a member of the project. The entry contains the user's project attributes (IAP, ACL groups, command environment limits).

System attributes (defined in the system database) are 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 Database: As an example of using project defaults for project members, consider the database for Project DENMARK, portrayed in Figure 4-5.

```
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:      10
    Default programs per level:  10
    Default dynamic segments:    64
    Default static segments:     64


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:
```

FIGURE 4-5. Database for Project DENMARK

As Figure 4-5 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 EDIT_PROFILE's CHANGE_USER subcommand, CLAUDIUS edits the project DENMARK database 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 4-6 shows the project database 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.

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 the contents of an ACL, use the LIST_ACCESS command.

```
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:     10
    Default programs per level: 10
    Default dynamic segments:   64
    Default static segments:    64


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:   100
    Static segments:     75

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

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

FIGURE 4-6. Project DENMARK After Changes

Types of Access Rights and Identifiers

The access rights that can be granted by an ACL are shown in Table 4-2.

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 (P) access to the object and List (L) 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 5, Setting Access Rights.

These ACL symbols can be combined to specify a variety of rights. For example, the combination ULAR allows a user to attach to a directory, list and add to its contents, and read any file within it that is not otherwise protected.

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.)

TABLE 4-2. ACL Access Rights

Symbol	*Right*	*Applies To*	*Meaning*
R	Read	Files	File can be read or executed.
W	Write	Files	File can be modified.
X	Execute	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 can attach to directory.
L	List	Directories	Directory contents can be listed.
A	Add	Directories	Directory entry can be added.
D	Delete	Directories	Directory entry can be deleted.
P	Protect	Directories	Access can be changed.
O	Owner	Files and directories	Owner can set all rights, except P and ALL, and can change RWLOCK.
ALL		Files and directories	All of the above rights.
NONE		Files and directories	No access allowed.

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 database. Project-based ACL groups are also registered in a project database.

A system-based ACL group forms part of the user's entry in the system database. 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 directories.

Project-based ACL groups are part of the user's entry in a project database. 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 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 database. Such an entry can occur whenever a new user or project is added to the system database, 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 database can thus be kept up-to-date to reflect the current needs of the system.

Reserved Names for ACL Groups and Servers

The SA must be aware that several utilities and subsystems on the system require that particular names be reserved for their use. These names indicate ACL groups and (phantom) servers. For example, the ACL group named .BATCH_ADMIN$ is reserved for operators and administrators who need privileged rights to administer the Batch subsystem.

If the SA creates ACL groups or adds new user IDs without regard for these reserved names, users may receive certain privileged rights or certain limitations that they neither need nor want.

Table 4-3 gives a summary of reserved names for ACL groups. Table 4-4 gives a summary of names that are recommended for reservation to avoid administrative confusion. Table 4-5 gives a summary of reserved names for servers.

TABLE 4-3. Reserved Names for ACL Groups

Subsystem or Utility	*Reserved Name*
Batch[1]	.BATCH_ADMIN$ .BATCH_USER$ .BATCH$
BRMS	.BACKUP$
DBMS	.DBMS_ADMIN
EDIT_PROFILE	.PROJECT_ADMINISTRATORS$
NTS	.NETWORK_MGT$
PRIME INFORMATION™	.INFO_ADMIN
ROAM	.ROAM_ADMIN
Spooler	.SPOOL$$ .SPOOL_ADMINISTRATOR$
WSI300	.WSI_FTP$

[1]**Batch since Rev. 21.0 requires both .BATCH_ADMIN$ and .BATCH_USER$. Pre-Rev. 21 BATCH uses .BATCH$.**

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 Database
- Inability to use projects (other than project DEFAULT) on your system
- Inability to use certain products that require ACLs (for example, Spooler, Batch, and PRIMIX)
- Decreased security on other subsystems

You can mix ACLs and password directories, but only as follows. Beneath a password MFD, use only password directories; you cannot use ACLs. You can use both ACLs and password directories beneath an MFD with ACLs. If you make a password subdirectory under an MFD with ACLs, you can no longer use ACLs under that subdirectory.

For a comparison of the security provided by ACLs and passwords, see Chapter 7, Security.

DESIGNING YOUR DATABASE

Before you use EDIT_PROFILE to create your User Profile Database, you should sketch out its design and parameters. You should take three general steps:

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.

Each step is discussed below.

TABLE 4-4. Names Recommended for Reservation

Subsystem or Utility	*Reserved Name*
DSM	DSM Groups:[1] .ALIEN_NODES$ .ANY_NODE$ .ANY_USER$ .ANY_FUNCTION$ .GROUP$ Other DSM Names:[2] ALIEN$ DSM$ LOCAL$

[1]These are not reserved ACL groups within the SAD. DSM groups are recorded within the Configuration File (CF) for DSM. You may also use them as ACL groups within EDIT_PROFILE without any confusion for PRIMOS. However, you are urged not to do so to avoid possible administrative confusion. See the DSM User's Guide for details.

[2]DSM uses these functions and roles within its CF. While you can use these names as login user IDs within EDIT_PROFILE, you are urged not to do so to avoid possible administrative confusion.

TABLE 4-5. Reserved Names for Servers

Subsystem or Utility	*Reserved Name*	*Server*
Batch	BATCH_SERVICE[1]	Batch Process Server
C2	AUDITOR	Security Audit Server
DSM	DSMSR	DSM Process Server
	DSMASR	DSM Application Server(s)
	DSM_LOGGER	Special DSMASR to log Users
	SYSTEM_MANAGER	DSM System Manager (redirects system messages to event logging)
FTS	FTP (suggested)	File Transfer Server
	YTSMAN	FTS Manager
LAN300™	LHC_DLL_SERVER	LHC Downline Loader
	LHC_ULD_SERVER	LHC Upline Dumper
	LTS_DLL_SERVER	LTS Downline Loader
NPX	SLAVE$	Slave(s) for Network Process Exchange
NTS™	NM_SERVER	NTS Network Manager
	NTS_SERVER	NTS Server
PRIMENET™	NETMAN	PRIMENET Network Manager
	RT_SERVER	Route-through Server
	ISC_NETWORK_SERVER	Inter Server Communications Server
PRIMOS	LOGIN_SERVER	Login Server
	TIMER_PROCESS	Timing Server
	SYSTEM[1]	Server (for User 1)
	LOGOUT_SERVER	Logout Process Manager
	UBI_SERVER	User Backplane Interconnect Server
PRIME/SNA™	SNA_SERVER	SNA Server
	SNA_3270	SNA Interactive Server
WSI300™	WSI_MANAGER	Manager activated by
	WSI_USER0	START_WSI
	WSIFTP_USER_PHANTOM*nn*	Spawns servers for local
	WSIFTP_SERVER*nn*	users
	WSIFTP_SERVER_PHANTOM*nn*	Server(s) for local users Server(s) for FTP process Server(s) for remote users

[1]**Both BATCH_SERVICE and SYSTEM must be reserved names for those systems that maintain a C2-certified level of security. No user may receive either of these names as a user ID. While an SA for a site not maintaining a C2-certified level may use them for user IDs, it is recommended that they remain reserved names.**

Grouping Users

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

- What groups do your users 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 Database you create. See Chapter 7 for security issues you must be aware of in choosing the degree of security most suitable for your system. 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 in the section below, Examples of Databases.
- 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 Databases.
- 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 Databases.

For more information on security considerations, see Chapter 7, Security.

Drawing Up User and Project Lists

After you know how you want to organize your database, 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 database with EDIT_PROFILE. Table 4-6 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 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 as many as 20,000 users in a project.)

   Figure 4-7 shows a sample form for creating a master system-user list. Figure 4-8 shows a sample form for creating a project list.

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

   a. For each user define a user ID, a temporary password, and (optionally) a password lifetime. (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 *PRIMENET Planning and Configuration 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 Database for your system. Chapter 6, Using EDIT_PROFILE, explains how to use the EDIT_PROFILE utility to build your database.

TABLE 4-6. Rules for Defining User and Project Attributes

User ID	Must be 1 through 32 characters long; must begin with an alphabetic character; can contain only letters, digits, periods (.), underscores (_), and dollar signs ($).
User password	Must be 0 through 16 characters long. A password with no characters is a null password (that is, the user enters only a carriage return). (In EDIT_PROFILE, the command NO_NULL_PASSWORD allows or prohibits null passwords and the command MINIMUM_PASSWORD_LENGTH sets a minimum length.) May contain any characters except PRIMOS reserved characters, which are defined in the *PRIMOS User's Guide.*
User password lifetime	May be set to infinite (-1). May be set to expire in a time period from 1 through 99,000 days. Users assigned a lifetime of 0 inherit the system default for a password lifetime. Users not assigned a password lifetime also inherit the system default.
ACL group name	Must be 2 through 32 characters long; must begin with a period; can contain only letters, digits, periods (.), underscores (_), and dollar signs ($).
Project name	Follows the same rules as user IDs.
Command environment limits	Command levels, 1 through 100; live program invocations per level, 1 through 100; private dynamic segments, 16 through 504; private static segments, 8 through 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 default password lifetime	If not set, defaults to an infinite lifetime. May be set, following the same rules for user password lifetime, except that a setting of 0 is invalid.

System User List

Drawn up by: BYRD

Date: 7/7/88

System Name: SYSX

Command Levels: 10
Programs per Level: 10
Dynamic Segments: 64
Static Segments: 64

ID: FROG

ACL Groups: .AMPHIB

Password: GREEN
Password Lifetime: (DEFAULT)
Default Project:
Other Projects: SWAMP, HOLLYWOOD

ID: PIG

ACL Groups: .VIPS
.PIGS
.BEAUTIES

Password: BEAUTIFUL_STAR
Password Lifetime: 120
Default Project:
Other Projects: HOLLYWOOD

ID: POSSUM

ACL Groups:

Password:
Password Lifetime:
Default Project: SWAMP
Other Projects:

ID: MONSTER

ACL Groups: .TRASH_LOVERS

Password: COOKIES
Password Lifetime: -1
Default Project: DUMPSTER
Other Projects: HOLLYWOOD

FIGURE 4-7. Sample Master System-User List

Project Data List

Drawn up by: ALL
Date: 7/7/88

Project Name HOLLYWOOD

Project Administrator: Name ANN L. ROONEY
User ID ANN

Master Project Limits:

ACL Groups Defined for This Project: .STARS .HEROES .VILLAINS .OTHERS .SUPERHEROES .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: 64
Static Segments: 64

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:

FIGURE 4-8. Sample Project List

EXAMPLES OF DATABASES

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: 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 directories named A, B, and C. He sets a quota of 10,000 records on each directory, which establishes the space for each team.

3. The SA creates the database 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. Directory A is the Initial Attach Point of anyone in Team A. (If directory A had subdirectories, any of those subdirectories could serve as Initial Attach Points for team members.)

   The SA wants to limit the chance that users will accidentally reveal a password. Therefore each user is registered with a password lifetime of 30 days.

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 directories. The ACL for directory A is set as follows:

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

   The ACL gives Project Administrator AMY ALL rights to her project's directory, 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 directory A, use its files, or gain any information about its contents.

ACLs for directories B and C are similar.

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

   system_administrator: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 directories. To do that, the Project Administrators need both List (L) and Use (U) 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.

9. 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 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 large, new 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 applies:

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

Example 2: 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 system possible. 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:

system_administrator:ALL
$REST:DALURW

ACLs on top-level directories 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, and 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:

   system_administrator: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: 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 projects, and whatever guests may visit the system.
2. The SA sets up six additional projects, one for each of the six math courses in which students will use the computer. As research projects are defined, she may set up projects for them as well.
3. 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.
4. 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, and so on. 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 64. 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 used most heavily. For the graduate courses, she defines a few other ACL groups that may be used for joint projects.
5. After the system is established, teams of a transitory nature may arise. These teams may want security for their work, but there is 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.
6. The SA sets up the top-level directories on the system, protecting them with the following ACL:

   system_administrator:ALL
   .FACULTY:DALURW
   $REST:U

7. She sets a quota on each top-level directory, 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.

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

9. 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.

Every student is registered on the system with a password lifetime of 120 days. The students can remain in the system database 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.

5

SETTING ACCESS RIGHTS

The System Administrator is responsible for setting access rights on Master File Directories (MFDs), system directories, and top-level user directories. The System Administrator is also responsible for setting access rights on assignable peripheral devices. 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.) Proper *device access* limits the use of assignable devices.

The sections of this chapter provide the System Administrator with the information necessary for carrying out the following responsibilities:

- To decide whether to use ACLs or password directories for governing system access. Use the command PASSWORD_DIRS to reflect your decision. The default for this command is PASSWORD_DIRS -ON. To run the system at a C2-certified level of security, you *must* change this to PASSWORD_DIRS -OFF.
- To maintain and modify both System Default search rules lists and Administrator search rules lists in the directory SEARCH_RULES*.
- To use priority ACLs on appropriate occasions. You may need to set a priority ACL on a partition when a user needs special access to the entire partition.
- To apply device ACLs
    - You must verify or create the proper subdirectories under DEVICE*, one subdirectory for each device that is to receive device ACLs.
    - You must provide authorized users the proper device ACLs right (U only) on those subdirectories.

  Then you may use the command DEVICE_ACLS -ON. This command defaults to DEVICE_ACLS -OFF to allow you to establish the proper subdirectories and device ACL groups.

The first part of this chapter discusses what protection to set on MFDs, system directories, and top-level user directories. It also describes the PASSWORD_DIRS command.

The second part describes the effect of ACLs on the operation of the ATTACH command.

The third part describes the System Administrator's role regarding search rules and the directory SEARCH_RULES*.

The fourth part describes the System Administrator's role regarding priority ACLs.

The fifth part describes the System Administrator's role regarding device ACLs and the directory DEVICE*.

PROTECTING SYSTEM AND USER DIRECTORIES

Although you can set system access by using passwords on directories, the use of Access Control Lists (ACLs) is recommended because ACLs provide better security and more flexibility. (See Chapter 7, Security, for a comparison of ACLs and passwords.) If your system is to maintain security at a C2-certified level, you are prohibited from using password directories. See the PASSWORD_DIRS command below.

For an explanation of ACLs to new users, see the *PRIMOS User's Guide.* For a review of ACLs, see Chapter 4 of this guide, Planning the User Environment.

The PASSWORD_DIRS Command

The PASSWORD_DIRS command enables the System Administrator to control the use of password directories on the system. It is recommended that you use ACLs, instead of passwords, to control access to directories. To prevent the creation of additional password directories, use the command PASSWORD_DIRS -OFF and embed the command in the PRIMOS.COMI file.

Format

PASSWORD_DIRS {**-ON** | **-OFF**}
PWDIR

Options

The -ON option allows password directories, and the -OFF option prevents the creation of new password directories.

Discussion

If you specify PASSWORD_DIRS -OFF, password directories may not be created either from programs or at command level. An attempt to create a password directory elicits this error message:

```
Use of password directories not allowed on this system (CREATE)
```

An attempt to convert an ACL directory to a password directory elicits an error code. Its error message E$NPDA indicates `no password directories allowed`.

Note that at cold start PASSWORD_DIRS -ON is the default, mainly for users of previous PRIMOS revisions. The System Administrator may initiate a gradual transition to all ACL directories. First the SA must convert any password MFD to an ACL MFD. The SA then issues the command PASSWORD_DIRS -OFF and inserts it in the PRIMOS.COMI file for ensuing system startups. The old password directories may still be accessed, but no new ones may be created. Thereafter, the SA can urge users to change old password directories to ACL directories.

Systems that are to maintain a C2-certified level of security must have the command PASSWORD_DIRS -OFF inserted before the MAXUSR command in the PRIMOS.COMI file. No password directories may exist on a system maintaining security at a C2-certified level. The System Administrator is supplied software utilities to enforce this condition. See detailed directions for running a C2-secure system in Chapter 7, Security.

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 (U) rights to access a partition at all. (See the section below, ACLs and the ATTACH Command.)
- Users need List (L) rights to list the partition contents or to protect top-level directories.
- Users may need Read (R) rights on an open system. At a minimum, you should grant users R 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 (A) rights to the MFD can create top-level directories, and only users who have Protect (P) and List (L) access to the MFD can set protection on the top-level directories. (On many systems, only the System Administrator and operators have P and A rights to the MFD. Thus, if users do not have P rights to the MFD and they accidentally lock themselves out of their top-level directories by changing their ACLs, you have to modify ACLs for them in order to restore their access.)

It is your responsibility to decide what rights to top-level directories to give users to users. 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 an EPF runfile is on a local partition, the X access allows the user to execute the runfile but not to read or copy it with a standard file utility, such as the COPY command. If an EPF runfile is on 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.

O This access right applies to files and directories. It allows the user to set access rights, except for P and ALL. If the object is a file or a segment directory (SEGDIR), the owner is permitted to set the read/write lock.

P	Users can change protection, the ACL rights themselves, on the directory or file. This right is not recommended for the general user, and it is rarely given alone (see ALL below).
ALL	Users can do anything to the directory (or to any directories beneath it in the tree structure), including changing ACLs. The ALL right corresponds to the rights OPDALURWX. However, ALL rights to a directory are limited to those rights given its parent directory.

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 such as 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 5-1 lists the minimum protection required for standard system directories. Table 5-2 lists the minimum access required for special products. (You may have all, some, or none of these products on your system.) Refer to Chapter 4 for tables of reserved names for ACL groups and system servers. An additional table (Table 4-4) indicates certain names that are recommended but not required for reservation.

TABLE 5-1. Access Rights for System Directories

Directory	*Minimum Access Needed*
BATCHQ[1]	(protection set by Batch subsystem)
CMDNC0	*system_administrator*:ALL SYSTEM:ALL $REST:LUR
DEVICE*	SYSTEM:PDALU *system_administrator*:PDALU $REST:U
Subdirectories to DEVICE*	SYSTEM:U *system_administrator*:U $REST:NONE
DSM*	.DSM$:ALL SYSTEM:ALL *system_administrator*:ALL $REST:U
DOWN_LINE_LOAD*	SYSTEM:ALL *system_administrator*:ALL LHC_DLL_SERVER:LUR LTS_DLL_SERVER:LUR .DSM$:LUR $REST:LUR
DOS	SYSTEM:ALL *system_administrator*:ALL $REST:LUR
HELP*	SYSTEM:ALL *system_administrator*:ALL $REST:LUR
INFO	SYSTEM:ALL *system_administrator*:ALL $REST:LUR
LIB	SYSTEM:ALL *system_administrator*:ALL $REST:LUR (DALURW recommended for users who must modify the libraries)

[1]**This must not be a password directory for the Batch subsystem for PRIMOS Rev. 21.0 and beyond. See Chapter 8, Adding Subsystems, for details on the use of other versions of Batch with Rev. 22.0 PRIMOS.**

TABLE 5-1. Access Rights for System Directories (continued)

Directory	*Minimum Access Needed*
LIBRARIES*	SYSTEM:ALL *system_administrator*:ALL $REST:LUR (DALURW recommended for users who must modify the libraries)
LOAD_MAPS*	SYSTEM:ALL *system_administrator*:ALL .DSM$:UR $REST:LUR
LOGREC*[1]	SYSTEM:DALURWX *operators*:ALL (recommended) $REST:LUR
MFD **(on command device)**	$REST:LU
NETWORK_MGT*	DSM_LOGGER:ALL .DSM$:UR SYSTEM:ALL $REST:LUR
PRIRUN	SYSTEM:ALL $REST:NONE
SAD	LOGIN_SERVER:ALL *system_administrator*:ALL $REST:LU (These ACLs, normally maintained by EDIT_PROFILE, should not be modified.)
SEARCH_RULES*	SYSTEM:ALL *system_administrator*:ALL $REST:LUR
SEGRUN*	SYSTEM:ALL *system_administrator*:ALL $REST:LUR
SERVERS*	SYSTEM:ALL *system_administrator*:ALL $REST:LUR
SIT*	SYSTEM:ALL *system_administrator*:ALL $REST:LUR

[1]**The directory LOGREC* holds only system event log files previous to Rev. 21.0 PRIMOS. Refer to DSM*>LOGS for event log files at Rev. 21.0 and beyond.**

TABLE 5-1. Access Rights for System Directories (continued)

Directory	*Minimum Access Needed*
SPOOL* [1]	.SPOOL_ADMINISTRATOR$:ALL $REST:LUR
SPOOL_DATA* [1]	.SPOOL$$:ALL $REST:NONE
SPOOL_QUEUE* [1]	.SPOOL$$:ALL $REST:NONE
SYSCOM	SYSTEM:ALL *system_administrator*:ALL $REST:LUR
SYSOVL	SYSTEM:ALL *system_administrator*:ALL $REST:LUR
SYSTEM	SYSTEM:ALL *system_administrator*:ALL $REST:LUR (for SYSTEM>DISCS)
SYSTEM_DEBUG*	SYSTEM:ALL *system_administrator*:ALL $REST:ALL
TERM* **TERM_U2***	SYSTEM:ALL *system_administrator*:ALL $REST:LUR
UP_LINE_DUMP*>LAN300	SYSTEM:ALL *system_administrator*:ALL LHC_ULD_SERVER:ALL LTS_ULD_SERVER:ALL .DSM$:UR $REST:LUR

[1] **At Rev. 21.0 and beyond, directories for the Spooler subsystem must not use passwords. Use the utility SYSTEM>SPOOL.INSTALL_ACL.CPL at installation to set the correct default ACLs. Refer to Chapter 8, Adding Subsystems, for a summary of directory setups for the Spooler subsystem, revised at Rev. 21.0.**

TABLE 5-2. Access Rights for Special Products

Product	*Directory*	*Minimum Access Needed*
DISCOVER	DISCOVER*	Normally maintained as a password directory.
FED	FED*	$REST:RU
FORMS	FORMS*	$REST:ALL
FTS	FTS	*system_administrator*:ALL
	FTSQ*	SYSTEM, YTSMAN, FTP, RT_FTP, and FTS Servers:ALL $REST:DALURW
NTS	NTS*	SYSTEM:ALL *system_administrator*:ALL $REST:LUR
POWERPLUS	POWER* POWRCM	$REST:ALL
PRIMENET	PRIMENET*[1]	NETMAN:ALL RT_SERVER:AURW *network_administrator*:ALL SYSTEM:ALURWX SLAVE$:LUR $REST:LUR
MFD containing PRIMENET*		NETMAN:U RT_SERVER:U SLAVE$:U
PRIME/SNA	PRIME/SNA*	SYSTEM:ALL SNA_SERVER:ALL SNA_3270:ALL *sna_administrator*:ALL
PRIMIX	PRIMIX*	SYSTEM:ALL *system_administrator*:ALL $REST:LURX

[1]**The directory PRIMENET* holds separate network event log files for pre-Rev. 21.0 PRIMOS. Refer to DSM*>LOGS>NETWORKS for the single file holding network event log files at Rev. 21.0 and beyond.**

TABLE 5-2. Access Rights for Special Products (continued)

Product	*Directory*	*Minimum Access Needed*
RJE	RJSPLQ*	*operator*:ALL *user*:ALL
	RJSPLQ*>CMDHELP RJSPLQ*>ERRHELP	*operator*:LUR *user*:LUR
	RJSPLQ*>BINARY RJSPLQ*>PUNCH RJSPLQ*>Q***nnn*** RJSPLQ*>SAVE RJSPLQ*>SDRF RJSPLQ*>TO_ROUTE	*operator*:DALURW *user*:NONE
	RJSPLQ*>CMDNC0 SYSCOM	*operators*:NONE *user*:NONE
ROAM	ROAM*	.ROAM_ADMIN:ALL SYSTEM:ALL $REST:LUR

ACLS AND THE ATTACH COMMAND

During its operation, the ATTACH command checks access rights on MFDs and directories to determine whether a user may be attached to a directory. If user rights to the directory or MFD are insufficient, the attach is not performed.

PRIMOS performs attaches using the following general order:

1. If a user specifies a partition name or a logical device number in an ATTACH command, only the specified partition is searched.
2. If a user applies a relative pathname in an ATTACH command, only the current directory tree is searched. A relative pathname begins with the *> symbol (for example, *>LETTERS).
3. If the default file ATTACH$.SR (which holds the single rule -ADDED_DISKS) is provided, it mimics the search steps of PRIMOS, as outlined in step 4 below. Note, however, that this order may be changed by altering ATTACH$.SR. Be aware that, if ATTACH$.SR totally excludes the rule -ADDED_DISKS, users may experience the loss of PRIMOS functionality.
4. If the default ATTACH$.SR has not been activated and a user supplies a fully qualified pathname that begins with a top-level directory, PRIMOS performs a search as shown in the flow chart in Figure 5-1. PRIMOS searches only those partitions to which the user has Use (U) rights. The default order for searching partitions is

a. All local partitions first, in logical device order

b. All remote partitions (if any) next, in logical device order

Search Finish

An ATTACH command starts a search that finishes when one of the following occurs:

- A top-level directory of the right name is found.
- All available partitions have been searched.

If the search finds the directory, and the user has U rights to it (and to any subdirectories specified in the pathname), the user is attached.

If the search finds a top-level directory of the right name, but the user does not have U rights to it, the following occurs:

- If the user has List (L) rights to the MFD, the search ends and the user receives the error message `Insufficient access rights.`
- If the user does not have L 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.`

Remote Searches

Remote searches proceed 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).

Table 5-3 shows good and poor orderings of a list. (In Table 5-3, the names of local partitions begin with LOCL; the names of remote partitions begin with SYS.)

Note

If a remote system is not running 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.

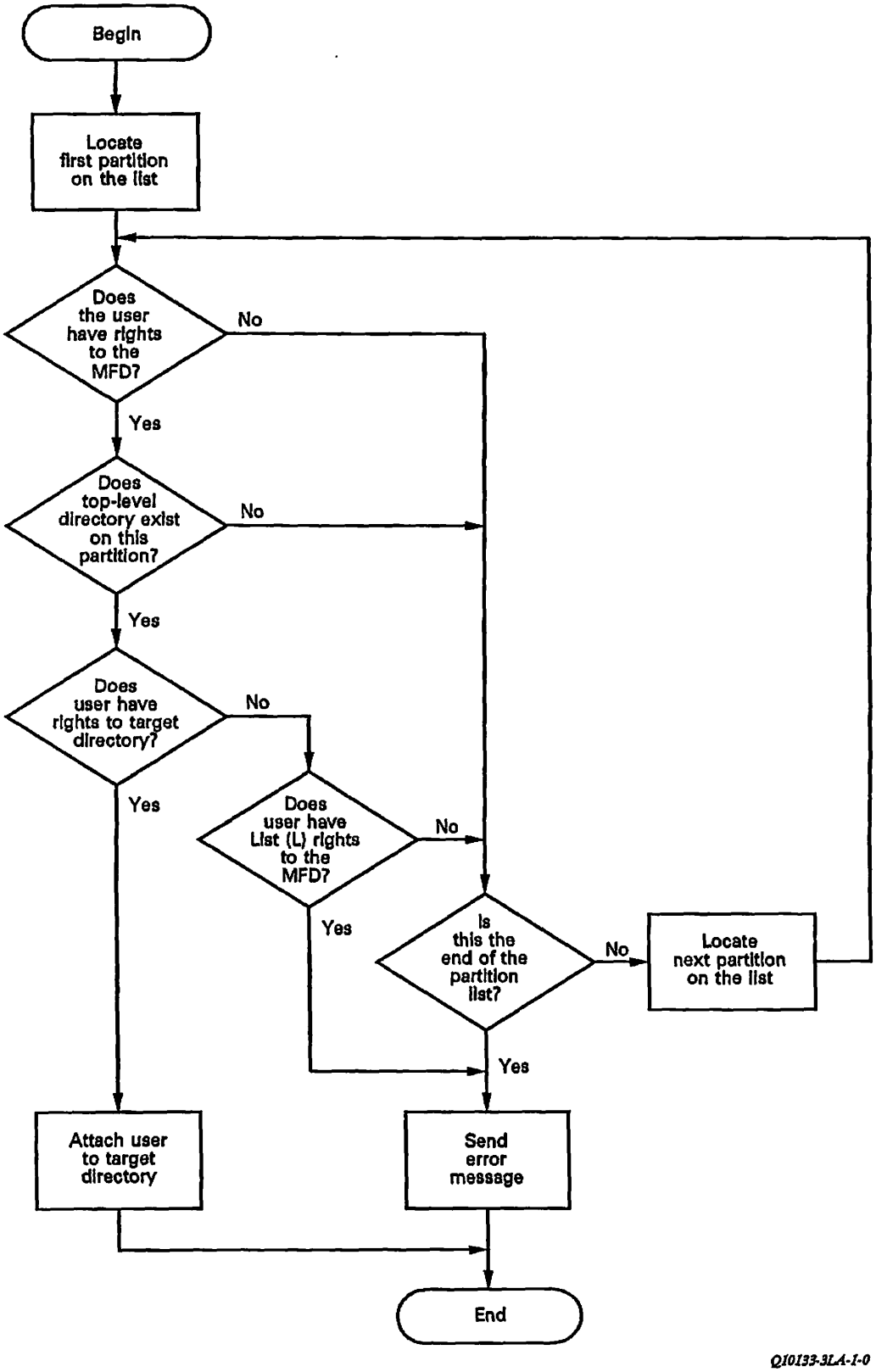

FIGURE 5-1. Search Order for ATTACH

TABLE 5-3. Good and Poor Ordering of Ldev Numbers

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)*	

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, 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 for attaches to remote directories.

Valid attaches to remote directories can be 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 not running. The user receives the message `Remote system down.`
- The remote system is running, 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. A user has at least LU rights to both. The user types ATTACH BLUE in an attempt to attach to <COLOR2>BLUE, but attaches 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.

THE PRIMOS SEARCH RULES FACILITY

PRIMOS uses a basic set of active search rules lists. At the installation of PRIMOS 22.0 the SA should have enlarged and modified the search rules lists. See the *Software Installation Guide, Rev. 22.0* for details. Search rules permit each user to define a sequence of locations to search for file system objects. Each process at initialization reads the presently active set of search rules lists, located in the directory SEARCH_RULES*. The process copies the rules and follows them until process completion, even if the lists in SEARCH_RULES* receive changes in the interim.

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

- System Default search rules. These are the default for a user who does not create a list of personal search rules. If users create a personal search rules list, they have the option of including or excluding System Default search rules.

- Administrator search rules. These are always first, before both system default rules and user-defined search rules. Users cannot modify or exclude Administrator search rules.

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

The Directory SEARCH_RULES*

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

ATTACH$.SR
BINARY$.SR
COMMAND$.SR
ENTRY$.SR
INCLUDE$.SR

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

ADMIN$.ENTRY$.SR

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

System Default Search Rules

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

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

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

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

Finally, if there are no search rules lists, PRIMOS in many cases performs the same location operations as the System Default search rules. (See ACLs and the ATTACH Command earlier in this chapter.) For example, without an ATTACH$ search rules list, PRIMOS searches the disk list when an MFD is requested. Without a COMMAND$ search rules list, PRIMOS searches CMDNC0 only.

System Administrator Rules

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

Administrator search rules lists are defined in SEARCH_RULES*, using the naming convention ADMIN$.*list_name*.SR. While Rev. 21.0 provides a single default Administrator search rules list (ADMIN$.ENTRY$.SR), the SA may add to that list or create additional Administrator search rules lists. The Administrator search rules lists must have the same ACL protection as their parent directory:

system_administrator:ALL
$REST:LUR.

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

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

To maintain security, however, the SA must apply one Administrator search rule to all users. ADMIN$.ENTRY$.SR must contain the entry: -PRIMOS_DIRECT_ENTRIES. Failure to include this Administrator search rule results in a breach of system security. The ADMIN$.ATTACH$.SR search list should not contain the entry -ADDED_DISKS, unless the SA wishes to disable all user-defined ATTACH$ search rules.

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

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

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

Other Administrative Aspects

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

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

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

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

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

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

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

PRIORITY ACLS

System Administrators and operators occasionally need special access to all files and directories on a partition. For example, they need Read (R) access to all files to perform a backup. They may create special access 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 an entire partition, although they may apply to its MFD, and then by default apply to all its subdirectories 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 to $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 those for 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 see if there is a 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.

DEVICE* AND DEVICE ACLS

From Rev. 21.0 onward the System Administrator may control access to assignable peripheral devices by

- Verifying and establishing the proper subdirectories under the directory DEVICE*
- Applying device ACLs to these subdirectories by
    - Using the standard commands for setting ACLs
    - Activating device ACLs

This section explains the System Administrator's role in establishing device ACLs. Topics include

- The DEVICE_ACLS command
- The directory DEVICE*
- The procedures for setting initial device ACLs
- Setting device ACLs on assignable disks
- Other uses for device ACLs

The DEVICE_ACLS Command

Format

```
DEVICE_ACLS {-ON }
DEVACL      {-OFF}
```

Discussion

The default at cold start is DEVICE_ACLS -OFF. This default allows the System Administrator to first configure DEVICE* before turning on device ACLs. Until the SA issues the DEVICE_ACLS -ON command, the system ignores the contents of the directory DEVICE*, and all users are granted access to all assignable devices.

The System Administrator activates device access control features by using the command DEVICE_ACLS -ON. PRIMOS responds to the DEVICE_ACLS -ON command first by searching all local partitions for the directory DEVICE*. If it does not exist, the following message is displayed:

```
Warning: Device ACLs are enabled but DEVICE* could not be found.
```

As part of the installation of DEVICE*, the proper device ACLs are provided for both DEVICE* and its subdirectories, as listed in the next section. However, the System Administrator must provide access to authorized users. In addition to the initially installed

subdirectories to DEVICE*, the SA may need to create optional subdirectories under DEVICE* for particular assignable devices on the system, such as asynchronous lines and assignable disk partitions.

As System Administrator, you may include DEVICE_ACLS -ON in the PRIMOS.COMI file, but you must first provide authorized users the Use (U) access right to the subdirectories under DEVICE*.

Note

Until the System Administrator provides Use (U) rights to other users on the system, the SA is the only user with device access rights. See the section Procedures for Setting Initial Device ACLs later in this chapter.

The Directory DEVICE*

The top-level directory DEVICE* may reside only on the command partition. The System Administrator must install it, since it requires the user ID of the SA for the creation of proper ACLs. To install DEVICE*, the SA types the following:

R SYSTEM>DEVICE_ACLS.INSTALL_ACL.CPL *system_administrator's_name*

The installation assigns the following default ACL settings to DEVICE*:

system_administrator:PDALU
SYSTEM:PDALU
$REST:U

All subdirectories in DEVICE* have the following default ACL settings:

system_administrator:U
SYSTEM:U
$REST:NONE

The System Administrator controls access to peripheral devices by providing a U right to the user(s) for one or more of the subdirectories listed in Table 5-4. For details on how to provide the U right, see the section Procedures for Setting Initial Device ACLs later in this chapter. Only the U right provides device access. The NONE right specifically denies device access. All other access rights are ignored by the device ACLs mechanism.

Device List

Table 5-4 provides the names of valid subdirectories to DEVICE*. These subdirectories function as a list of devices that may be protected by ACLs.

The System Administrator must provide the device access right of U to each authorized user of the subdirectories under DEVICE*. (See the next section.) The SA may also wish to create subdirectories that correspond to assignable disks and asynchronous lines on that particular system.

TABLE 5-4. Device List for Device ACLs

Subdirectory	*Description of Device Name*
CENPR	The serial printer.
CE2PR	The second serial printer.
CARDR	The serial card reader.
PTR	The paper tape reader/punch.
PUNCH	The card punch.
PR*n*	MPC printer *n*, where *n* ranges from 0 through 3.
CR*n*	Parallel card reader *n*, where *n* ranges from 0 through 1.
MT	A directory that is checked when the -ALIAS option is used with the ASSIGN command.
MT*n*	Magnetic tape unit *n*, where *n* ranges from 0 through 7.
SMLC*n*	Synchronous Communications line *n*, where *n* ranges from 00 through 07. Preceding zeros *must* be present.
SPARE*n*	Spare device *n*, where *n* ranges from 1 through 2. These devices may be assigned, but do not now correspond to any configured device.
PLOT	The plotter.
MG*n*	Megatek graphics display terminal *n*, where *n* ranges from 0 through 3.
GS*n*	Vector General graphics display terminal *n*, where *n* ranges from 0 through 3.
AL*n*	Asynchronous line number *n*, where *n* is normally a decimal number ranging from 0 through 512 for local assigned lines, or ranging from 1024 through 1535 for NTS assigned lines. The SA must customize the numbered range to fit within the number of processes supported by the system (presently a maximum of 960). Omit any preceding zeros. Thus, for asynchronous line number 07, the device directory must be named AL7.
DK*n*	Disk partition *n*, where *n* is the octal pdev of the partition. When making a pdev assignable by means of the DISK command, you can also create, and set access on, a corresponding DK*n* device directory within DEVICE*. If you are altering a previous DK*n* directory either by increasing the size of the partition or by unassigning the partition, first be sure to update the device ACLs on the old DK*n*.
DEFAULT	A default directory that is checked when an assignable partition is assigned. If you do not create a specific DK*n* device directory for a given partition, the device ACLs mechanism provides access to it for any users with a U right to DEFAULT.

Procedures for Setting Initial Device ACLs

Before activating device ACLs, the System Administrator could provide the U right for device ACLs to a core group of users on the system. The SA could use EDIT_PROFILE to do the following:

1. Use the LIST_SYSTEM -ALL command to list all system users.
2. Identify a core group of users requiring device access.
3. Assign the generic ACL group .DEV_USERS to each of them (see Chapter 6, Using EDIT_PROFILE).

Then, outside of EDIT_PROFILE, the SA could

1. Attach to the directory DEVICE*.
2. Create an access category as follows:

```
SET_ACCESS DEV_USERS.ACAT .DEV_USERS:U
```

   The system asks if you want to create an access category; respond YES.

3. Apply this access category to all the subdirectories of DEVICE* as follows:

```
SET_ACCESS *>@@ -CATEGORY DEV_USERS.ACAT
```

After all files and subdirectories have received protection from the access category, the following harmless error message is displayed:

```
Not a file or directory.  DEV_USERS.ACAT (set_access)
```

The error occurs after all files and subdirectories in this directory receive the proper ACL rights. Final application of the access category to the access category itself caused the error because the ACAT is neither a file nor a directory.

Note

If other users need U rights later, the SA must continue to include .DEV_USERS:U with any new ACLs change, unless the SA purposely wants to restrict device access. (See the following examples.) The SA most efficiently provides U rights to other users thereafter by entering EDIT_PROFILE and adding the ACL group .DEV_USERS to the profile of each new device user.

The SA may now issue the command DEVICE_ACLS -ON. The SA may also include this command in the PRIMOS.COMI file.

Setting Device ACLs on Assignable Disks

Device ACLs do not apply to partitions that have been added to the system (via the ADDISK command).

Device ACLs may be applied to assignable disks (those placed in the Assignable Disks Table by means of the DISK command). For this purpose DEVICE* holds the subdirectory

DEFAULT and the optional subdirectories DK*n*, where *n* is the pdev for a given partition. The device ACLs set on DEFAULT provide the U right to a list of users who would normally be allowed to assign such assignable disks. The device ACLs set on DK*n* provide the U right to a more specific list of users, for the particular partition with a pdev of *n*. For example, members of the ACL group .DEV_USERS presently have the U right to the subdirectory DEFAULT. The SA wants to create an even more restricted group of users with access to assignable disks. The SA uses EDIT_PROFILE to assign the ACL group .DEFAULT_USERS to each member of this restricted group. The SA then attaches to DEVICE* and changes the ACLs on the subdirectory DEFAULT, as follows:

```
SET_ACCESS DEFAULT .DEFAULT_USERS:U
```

If device ACLs have already been activated, then only members of the ACL group .DEFAULT_USERS may now assign disks to themselves.

To create one further level of restriction, the SA may decide to set device ACLs on a particular partition. For example, the SA may create a subdirectory to DEVICE* for a particular assignable partition with the pdev 10460. The SA names the subdirectory DK10460. The SA may grant exclusive permission to a user by setting the appropriate ACL on DK10460.

Note that such a level of restricted device access is optional. For example, user MEL has been included in the device ACL group .DEFAULT_USERS with U rights to DEFAULT. The partition with the pdev 10460 has been added to the Assignable Disks Table, with the command DISKS 10460. However, the System Administrator has not created the DEVICE* subdirectory DK10460. Now MEL issues the command ASSIGN DISK 10460. The device ACLs facility looks for DK10460 in DEVICE* but cannot find it. It then looks at the device ACLs for DEFAULT, finds MEL assigned to .DEFAULT_USERS, and so grants device access.

Setting Device ACLs on Assignable Asynchronous Lines

Other optional subdirectories to DEVICE* are those for assignable asynchronous lines. If you create any of these, you must name them AL*n*, where *n* ranges from 0 through 512 for local lines. The range from 1024 through 1535 is reserved for assigned lines within a local area network supporting Network Terminal Service (NTS).

You might use such assigned line device ACLs if your system has several printers and you have already engaged all the printer device ACL subdirectories. Or perhaps you wish to set a device ACL on a workstation that is assigned so that it may receive files from the central processor but is not able to ship files back. You might also wish to set device ACLs on printers that are assigned and available on NTS for a LAN300.

For example, an installed LAN300 is presently being used mainly to provide NTS to a series of terminal clusters on two different floors of a building next to the one holding the CPU with its local lines and local printers. The SA has already assigned a printer to each floor

of the building with NTS users. The SA has been requested to set a device ACL on the second floor NTS printer, so that only the payroll department may have access to the printer while checks are being printed. That printer has already been NTS-associated and assigned; its assigned line number is 1026. The SA has already assigned the ACL group .PAYROLL to the users who belong to the payroll department. The SA must first create the subdirectory AL1026 under DEVICE* before setting a device ACL:

```
OK, ATTACH DEVICE*
OK, CREATE AL1026
OK, SET_ACCESS AL1026 .PAYROLL:U
```

Until the device ACL is removed from the second floor printer, only members of the payroll department may use it.

Device ACLs and Magnetic Tape Security: The System Administrator should realize that a device ACL on a magnetic tape drive provides security for information on a magnetic tape only if the proper tape has been mounted. In fact, a user could potentially gain unauthorized access to information on a tape if the operator mounts the wrong one. Such an error might also lead to unauthorized modification of information.

The System Administrator must therefore establish and enforce strict tape handling procedures to eliminate the possibility of an operator error. This is especially true at a site maintaining a strict C2-certified configuration.

Other Uses for Device ACLs

The following examples show some other uses for device ACLs.

Example 1: The System Administrator wants to dedicate the use of a particular plotter to the group working on a presentation that must be ready as soon as possible. The group has been established in EDIT_PROFILE as .PRESENTERS. The subdirectory PLOT under DEVICE* presently grants a U right to members of the ACL group .DEV_USERS. The SA attaches to the directory DEVICE* and issues the command

```
SET_ACCESS PLOT .PRESENTERS:U
```

The system provides the default $REST:NONE.

If device ACLs have not yet been activated on the system, the System Administrator must invoke the command DEVICE_ACLS -ON before device access limits are set for PLOT. Thereafter the plotter may be used only by those users who were assigned the .PRESENTERS ACL group during the EDIT_PROFILE session.

Example 2: A publications group on the system has a serial printer that they wish to be accessible only to those already defined in the ACL group .PUBS. The System Administrator knows that another printer is available to other users, and that the printer in question corresponds to the subdirectory CE2PR under DEVICE*. The SA attaches to the directory DEVICE* and issues the following command to satisfy their request:

```
SET_ACCESS CE2PR .PUBS:U $REST:NONE
```

Only members of .PUBS may now use this printer.

Example 3: A project group working on customer service wants to reserve a partition for testing and analyzing problems received from the field. Their activities involve the use of FIX_DISK, PHYSAV, PHYRST, and other utilities. While they want the partition on an active disk drive available for their purposes, they do not wish the System Administrator to use the ADDISK command to make the partition common to all system users. The SA has therefore added the partition, whose pdev is 1460, to the Assignable Disks Table, using the command DISKS 1460. Now the customer service group wants to maintain private access to this partition for members of their group alone. They want to stop anyone outside their group from using the ASSIGN command on that partition.

The SA agrees to use a device ACL on the partition. The SA first creates the subdirectory DK1460 under DEVICE*. The SA enters EDIT_PROFILE and adds the ACL group .CUSRVC to the profile of each user in the customer service project. The SA then attaches to the directory DEVICE* and issues the following command:

```
SET_ACCESS DK1460 .CUSRVC:U $REST:NONE
```

Only members of .CUSRVC may now assign the partition for individual use.

Example 4: The computer operations group in charge of doing backups in a large machine room has to perform this chore while other users with access to the machine room are also using magnetic tape drives. The Ops Group convinces the SA to reserve one unit, MT0, for their own use. Several other units are available for users. An ACL group, .OPS, is already established for the Ops Group. The SA attaches to the directory DEVICE* and issues the following command:

```
SET_ACCESS MT0 .OPS:U $REST:NONE
```

The tape unit may now be used solely by members of the Ops Group.

Number Conversion For ALn Subdirectories

Those subdirectories beneath DEVICE* that are used for assigned asynchronous lines have names of AL followed by a unique decimal number (AL0 through AL512, AL1024 through AL1535). Prior to Rev. 22, they had octal numbers. If your site has a Rev. 21 version of PRIMOS, the numbers within the names of these subdirectories are automatically converted to decimal as part of the installation of Rev. 22.0 PRIMOS.

If, for some reason, a System Administrator wishes to revert to Rev. 21.0 PRIMOS, the SA must run a program to convert these subdirectories back to octal format.

The SA converts them to octal as follows:

```
RESUME TOOLS>CONVERT_DEVICE*.CPL -TO_REV.21
```

When Rev. 22.0 is reinstalled, the SA converts the subdirectories back to decimal format by running the program again without a command-line option:

```
RESUME TOOLS>CONVERT_DEVICE*.CPL
```

6

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 user. 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-based 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.

Plan your system before you create any profiles using EDIT_PROFILE. The worksheets in Chapter 4 may be helpful for planning. You should also read Chapter 5, Setting Access Rights, before you use EDIT_PROFILE.

INSTALLATION OF REV. 22.0 PRIMOS

System Administrators updating their systems to Rev. 22.0 of PRIMOS should use EDIT_PROFILE before admitting users. This procedure is recommended to initiate internal changes to the SAD.

To perform the actual installation, the System Administrator should refer to the step by step instructions provided in the *Software Installation Guide, Rev. 22.0.*

OVERVIEW OF EDIT_PROFILE

System Administrators use EDIT_PROFILE to do two kinds of tasks:

- To create a new System Administration Directory (SAD). The SAD contains a database that includes information about the users of your system and any groups and projects you create. When you install a Rev. 22.0 system for the first time, you must define each user and project for your system. You must create a SAD before your 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 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 need 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.*

USING PROJECT DEFAULT

Before using EDIT_PROFILE, you must decide whether you will create a system default project (which is always named 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 a maximum of 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 database unless it has at least one project.)

You can create project DEFAULT only while you are in Initialization mode. (Within the Initialization Mode section below, see step 4 in the section 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 Project DEFAULT.

INITIALIZATION MODE

While you are in Initialization mode, EDIT_PROFILE prompts lead you through the series of steps necessary to create the SAD.

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.)

Note

It is strongly recommended that you use ACLs instead of passwords. If your MFD has ACLs, EDIT_PROFILE establishes the proper ACL rights for the System Administrator.

After you issue the EDIT_PROFILE command, EDIT_PROFILE enters Initialization mode. The following text appears:

```
OK, EDIT_PROFILE
[Edit_Profile Rev 22.0 Copyright (c) 1988, Prime Computer, Inc.]
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 the EDIT_PROFILE initialization procedure that allow you to create a SAD and its respective files:

1. These prompts appear only at SAD initialization:

   a. `SAD does not exist.  Create it?`

      Type YES to create a SAD. Type NO to terminate EDIT_PROFILE and return to PRIMOS.

   b. `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. `*** Creating User Validation File. 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 Database. 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 database 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 database.

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 message that it has created the following five files, which are part of the SAD:

- User Validation file (UVF)
- Security Information file (SIF)
- Master Project file (MPF)
- Master Group file (MGF)
- System Default file (SDF)

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 database.

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 following chart for valid values.

Attribute	*Minimum*	*Maximum*	*Recommended*
Command levels	1	100	10
Live invocations per level	1	100	10
Dynamic segments	16	1016	64
Static segments	8	1008	64

Note

If you do not specify any of the above values, PRIMOS supplies default values of 10 command levels, 10 live invocations per level, 64 dynamic segments, and 64 static segments.

The sum of the private dynamic and static segments may not exceed 1024. 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 dialog, 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 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 to 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 you use them, an error message similar to the following is displayed:

```
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 SA forgets to use the -MFD_PASSWORD option, receives an error message, and then uses the option correctly.

```
OK, EDIT_PROFILE
[Edit_Profile Rev 22.0 Copyright (c) 1988, Prime Computer, Inc.]
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)
ERI EDIT_PROFILE -MPW XXXXXX
[Edit_Profile Rev 22.0 Copyright (c) 1988, Prime Computer, Inc.]
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 01 June 88 10:36:52 Wednesday.
*** 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 01 June 88 10:37:44
    268 entries in prime area; file is 13 records long.

Security Information File created 01 June 88 10:37:44
    268 entries in prime area; file is 13 records long.

Master Project File created 01 June 88 10:37:44

Master Group File created 01 June 88 10:37:44

System Default File created 01 June 88 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 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 Database 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 project 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 22.0 Copyright (c) 1988, Prime Computer, Inc.]
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 01 June 88 11:08:56 Wednesday.
*** 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 01 June 88 11:09:16
    92 entries in prime area; file is 5 records long.

Security Information File created 01 June 88 11:09:16
    268 entries in prime area; file is 13 records long.

Master Project File created 01 June 88 11:09:16

System Default File created 01 June 88 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 22.0 Copyright (c) 1988, Prime Computer, Inc.]
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 02 June 88 10:03:56
    124 entries in prime area; file is 6 records long.

Security Information File created 02 June 88 10:03:56
    124 entries in prime area; file is 6 records long.

Master Project File created 02 June 88 10:03:56

Master Group File created 02 June 88 10:03:56

System Default File created 02 June 88 10:03:56

>
```

SYSTEM ADMINISTRATOR MODE

After you have initialized the User Profile Database, 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 6-1 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 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 6-1. EDIT_PROFILE Commands

Command	*Used by*	*Function*
System Commands		
CHANGE_SYSTEM_ADMINISTRATOR **CSA**	SA	Changes ID of the SA
CHANGE_SYSTEM_DEFAULTS **CSD**	SA	Changes system defaults for command environment attributes
COMPUTER_GENERATED_PASSWORD **CGPW**	SA	Enables or disables generation of passwords by the computer
DEFAULT_PASSWORD_LIFETIME **DPWLIFE**	SA	Sets default value for a systemwide password lifetime
FORCE_PASSWORD **FPW**	SA	Prohibits the use of passwords on the login line
HELP	SA/PA	Displays syntax of all commands
LIST_SYSTEM **LS**	SA	Displays system and other attributes
MAXIMUM_PASSWORD_LENGTH **MXPWL**	SA	Sets maximum length for user passwords
MINIMUM_PASSWORD_LENGTH **MPWL**	SA	Sets minimum length for user passwords
NO_NULL_PASSWORD **NNPW**	SA	Prohibits use of null passwords
QUIT **Q**	SA/PA	Ends the EDIT_PROFILE session
REBUILD **RE**	SA/PA	Rebuilds the validation files
SET_DEFAULT_PROTECTION **SDPR**	SA	Restores protection to the SAD
SYSTEM_DEFAULTS **SD**	SA	Overrides the limits set for project-based and user-based command environments with the system-default values
VERIFY_PASSWORD_FORMAT **VPF**	SA	Verifies format of user passwords

TABLE 6-1. EDIT_PROFILE Commands (Continued)

Command	*Used by*	*Function*
Project Commands		
ADD_PROJECT **AP**	**SA**	Creates a new project
ATTACH_PROJECT **ATP**	SA/PA	Specifies the current project
CHANGE_PROJECT **CP**	SA/PA	Changes attributes of a project profile
DELETE_PROJECT **DP**	**SA**	Removes a project from the system
DETACH_PROJECT **DTP**	SA/PA	Detaches current project
LIST_PROJECT **LP**	SA/PA	Lists the attributes of a project profile
User-control Commands		
ADD_USER **AU**	SA/PA	Adds user to the system or to a project
CHANGE_USER **CU**	SA/PA	Changes a user's system or project attributes
DELETE_USER **DU**	SA/PA	Removes a user from the system or from a project
LIST_USER **LU**	SA/PA	Lists a user's system or project attributes
VERIFY_USER **VU**	**SA**	Checks for existence of a user ID on networked systems

SYSTEM COMMANDS

The 12 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 the user who will be the new System Administrator can log in to 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 rebooted 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 to be changed unless it recognizes the previous administrator making the change.

Format

CHANGE_SYSTEM_ADMINISTRATOR [*user-id*] [-ALL]
CSA

Discussion

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 user ID of 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.

Format

CHANGE_SYSTEM_DEFAULTS *option-1* [*...option-4*]
CSD

Options

You must supply at least one option to the command. If an option is not specified, the previous value of the attribute remains unchanged.

-DYNAMIC_SEGMENTS *n*
-DS

Sets the system default for private dynamic segments to *n*, where *n* has a range from 16 through 1016. The Prime-supplied value for *n* is 64. Certain programs may require more. For example, EMACS requires 100 private dynamic segments. The sum of private dynamic and static segments cannot exceed 1024. EPFs use dynamic segments.

-LEVELS *n*
-LEV

Sets the system default for the number of command levels to *n*, where *n* has a range from 1 through 100. 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*, where *n* has a range from 1 through 100. The Prime-supplied value for *n* is 10.

STATIC_SEGMENTS *n*
-SS

Sets the system default for private static segments to *n*, where *n* has a range from 8 through 1008. The Prime-supplied value for *n* is 64. The sum of private static and dynamic segments cannot exceed 1024. Programs loaded with SEG and LOAD use static segments. Certain programs (such as DBG) require at least 32 segments.

The COMPUTER_GENERATED_PASSWORD Command

Use the COMPUTER_GENERATED_PASSWORD command to enable or disable computer-generated passwords.

Format

COMPUTER_GENERATED_PASSWORD [{-ON | -OFF | -STATUS}]
CGPW

Options

The -STATUS option queries the system as to whether CGPW is presently enabled or disabled.

The -OFF option disables computer-generated passwords.

The -ON option activates computer-generated passwords. With a setting of CGPW -ON, the system provides a computer-generated password in three situations:

- When a user wants to change passwords.
- When a user's password has expired and must be changed.
- When all users first login after the activation of computer-generated passwords.

Note

Specifying COMPUTER_GENERATED_PASSWORD without an option or COMPUTER_GENERATED_PASSWORD -ON produces the same result: computer-generated passwords are enabled.

EDIT_PROFILE now allows the System Administrator to establish a period of time after which a user must change a login password. (See the DEFAULT_PASSWORD_LIFETIME command and the -PASSWORD_LIFETIME option for the ADD_USER and CHANGE_USER commands.) Refer to Examples of User Validation at Login in Chapter 7. The examples show various password change interactions, depending upon the status of computer-generated passwords after the expiration of a password lifetime.

The DEFAULT_PASSWORD_LIFETIME Command

Use the DEFAULT_PASSWORD_LIFETIME command to set a default value for a system-wide password lifetime. The SA may override this default by assigning a specific password lifetime to a user. (See the ADD_USER and CHANGE_USER commands later in this chapter.) If you have never used the DEFAULT_PASSWORD_LIFETIME command on your SAD (for example, when upgrading from Rev. 21), the default setting is an infinite password lifetime.

Format

```
DEFAULT_PASSWORD_LIFETIME [{-1           }]
DPWLIFE                   [{positive-num }]
```

Options

The -1 option specifies an infinite password lifetime. The *positive-num* option indicates a value ranging from 1 through 99,000, inclusive. The value indicates the number of days after which a password must be changed. The value zero is invalid.

The SA uses DPWLIFE several times in the following example:

```
> DEFAULT_PASSWORD_LIFETIME
Current default password lifetime is infinite.
    Password lifetime in days (-1 = infinite):

> DPWLIFE 0
Current default password lifetime is infinite.
*** Error - lifetime out of range.  Use -1 (infinite) or 1 to 99000.

> DPWLIFE 3
Current default password lifetime is infinite.
New default password lifetime is 3 days.
```

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.
```

Format

```
FORCE_PASSWORD {-ON  }
FPW            {-OFF }
```

Options

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.

Format

HELP [*command_name*]

Discussion

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 22.0 Copyright (c) 1988, Prime Computer, Inc.]
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
                                  -PassWord_LIFEtime, -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--

Change_User         user        -PROJect -LIST
                                -SYStem -PassWord -PassWord_LIFEtime
Computer_Generated_PassWord     -ON, -OFF, -STATUS
Delete_Project      project     none
Delete_User         user        -PROJect
DeTach_Project      project     none
Default_PassWord_LIFEtime
                    days        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
MaXimum_PassWord_Length
                    length      none
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_Password_Format
                    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 parentheses 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 *m*. (MINIMUM_PASSWORD_LENGTH was used with a length of *m* characters)

- Maximum password length is *n*. (MAXIMUM_PASSWORD_LENGTH was used with a length of *n* characters)
- Passwords must have a specific format. (VERIFY_PASSWORD_FORMAT was used)
- Null passwords not allowed. (NO_NULL_PASSWORD was used)

Format

LIST_SYSTEM [*options*]
LS

Options

-ALL
Lists all the information provided by the combination of the -USERS, -GROUPS, and -PROJECTS options.

-APPEND *n*
-APP
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
-DET
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
-GR
Lists all groups on the system.

-OUTPUT *pathname*
-OUT
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
-PROJ
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
-US
Lists systemwide attributes of all system users.

Example

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 22.0 Copyright (c) 1988, Prime Computer, Inc.]
In system administrator mode.
> LIST_SYSTEM

*************************************************************************
<SYS01>SAD                      Administrator: SEGOVIA
    Version 2 validation file.
    One entry in use out of 124.
    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 64.
    4. Maximum number of private static segments is 64.
*************************************************************************
>
```

The MAXIMUM_PASSWORD_LENGTH Command

Use the MAXIMUM_PASSWORD_LENGTH command to set a maximum length for user passwords. You can also use this command when computer-generated passwords are enabled. In this case, the computer-generated passwords will not exceed the specified maximum length.

Format

MAXIMUM_PASSWORD_LENGTH [*length*]
MXPWL

Discussion

length is a decimal integer ranging from 0 through 16, inclusive. Specifying a value for *length* outside of this range produces the error message

```
Maximum password length must be between zero and 16.
```

A length of zero means that no checking is done for a maximum length. In this case, the maximum allowable length is 16. A length of zero is the default. The maximum length should also be equal to or greater than the minimum length set by the MINIMUM_PASSWORD_LENGTH command.

After you set a maximum password length, users cannot issue the PRIMOS command CHANGE_PASSWORD with new passwords greater than *length*. Existing passwords are not affected.

If a maximum password length is in effect, the LIST_SYSTEM command displays the maximum length.

The MINIMUM_PASSWORD_LENGTH Command

Use the MINIMUM_PASSWORD_LENGTH command to set a minimum length for user passwords. You can also use this command when computer-generated passwords are enabled. In this case, the computer-generated passwords will be no shorter than the specified minimum length.

Format

MINIMUM_PASSWORD_LENGTH [*length*]
MPWL

Discussion

length is a decimal integer ranging from 0 through 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 length should also be equal to or less than the maximum length set by the MAXIMUM_PASSWORD_LENGTH command.

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 setting a minimum password length, you cannot assign login passwords shorter than *length*, and users cannot issue the PRIMOS command CHANGE_PASSWORD with new passwords 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.

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.

Format

NO_NULL_PASSWORD [{-ON | -OFF}]
NNPW

Options

The -ON option 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.

Notes

The NO_NULL_PASSWORD -ON command is required for a system to maintain a C2-certified level of security.

Specifying NO_NULL_PASSWORD without an option or NO_NULL PASSWORD -ON produces the same result: null passwords are prohibited.

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 Database, at either system level or at individual project level. You may want to rebuild the database 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 Database 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.

Format

REBUILD [**-PROJECT** [*project-id*] [**-SIZE** *entry-count*]]
RE

Options

-PROJECT [*project-id*]
-PROJ

Specifies that you want 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 Database 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.

-SIZE

Specifies the number of users for whom you need space, either in the system or the project-related database. EDIT_PROFILE always allows space for at least 20 users, both for the system and for each project, and can accommodate a maximum of 21,000 user profiles per system, and a maximum of 20,000 user profiles 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 primary area and the number of entries in use in the overflow area. EDIT_PROFILE then establishes the new size of the validation file by evaluating the size currently in use, increasing that size by 60%, and then rounding it off to the next higher prime number. This prime number is chosen to make searching the database as efficient as possible.

Example

In the following example, a System Administrator rebuilds the entire User Profile Database. 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 22.0 Copyright (c) 1988, Prime Computer, Inc.]
In system administrator mode.
> REBUILD
*** UVF backed up into file "UVF.OLD" 05 June 88 11:02:28.
*** MPF backed up into file "MPF.OLD" 05 June 88 11:02:28.
*** MGF backed up into file "MGF.OLD" 05 June 88 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" 05 June 88 11:02:32
*** MPP for project "DEALERS" backed up into
    "DEALERS>MPP.OLD" 05 June 88 11:02:36
*** MPP for project "PARTS" backed up into
    "PARTS>MPP.OLD" 05 June 88 11:02:36
*** MPP for project "FRITZ" backed up into
    "FRITZ>MPP.OLD" 05 June 88 11:02:36

*** Rebuild complete 05 June 88 11:02:36| ***
    Version 2 validation file.
    4 entries in use out of 20.
Delete old files? YES
*** SIF backed up into file "SIF.OLD" 05 June 88 11:02:36.
*** Rebuild complete 05 June 88 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.

Format

```
SET_DEFAULT_PROTECTION  [-CONVERT]
SDPR
```

Option

The -CONVERT option (abbreviated as -CNV) 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.

Format

```
SYSTEM_DEFAULTS  {-ON  }
SD               {-OFF }
```

Options

If you specify -ON, all users are logged in with the system default number of levels and segments. If you specify -OFF, 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.

THE VERIFY_PASSWORD_FORMAT COMMAND

Use the VERIFY_PASSWORD_FORMAT command to enable or disable verification of certain format restrictions on user passwords. If password-format verification is enabled, a user password must

- Begin and end with a letter
- Contain at least one digit

You can also use the VERIFY_PASSWORD_FORMAT command when computer-generated passwords are enabled. In this case, the computer-generated passwords will conform to the format restrictions.

After you enable password-format verification, users cannot issue the PRIMOS command CHANGE_PASSWORD with new passwords that do not conform to the format restrictions. Existing passwords are not affected.

If password-format verification is enabled, the LIST_SYSTEM command acknowledges the fact.

Format

VERIFY_PASSWORD_FORMAT [{-ON / -OFF}]
VPF

Options

The -ON option enables password-format verification.

The -OFF option disables password-format verification. -OFF is the default if the command has never been used before.

Note

Specifying VERIFY_PASSWORD_FORMAT without an option or VERIFY_PASSWORD_FORMAT -ON produces the same result: password-format verification is enabled.

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.

Format

ADD_PROJECT [*project-id* [*options*]]
AP

Options

If you specify any options, you must also specify *project-id*, which is the name of the project to be created.

-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 database can handle efficiently. This warning enables you to rebuild the database, 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

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. DLR_MAN is the very first Project Administrator added to the system. Consequently, when the SA completes the registration of DLR_MAN, the system announces that a new ACL group, .PROJECT_ADMINISTRATORS$, has been added to the system.

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 22.0 Copyright (c) 1988, Prime Computer, Inc.]
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".
    Password lifetime in days (-1 = infinite, 0 = default): 120

    New password lifetime value:  120 days
```

```
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: 64
    Maximum number of private, static segments: 64
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: 64
    Maximum number of private, static segments: 64
-----------------------------------------------------
    Number of command levels: 10
    Number of live program invocations per command level: 10
    Number of private, dynamic segments: 64
    Number of private, static segments: 64
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: 64
    Maximum number of private, static segments: 64
-----------------------------------------------------
```

```
    Number of command levels: 5
    Number of live program invocations per command level: 5
    Number of private, dynamic segments: 64
    Number of private, static segments: 64
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: 64
    Maximum number of private, static segments: 64
------------------------------------------------------

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: 64
    Number of private, static segments: 64
************************************************************************
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.

Format

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.

Format

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.

Options

-LIMITS
-LIM
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
-PROF
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 Database 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.

Discussion

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*]
-DL

Example

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 22.0 Copyright (c) 1988, Prime Computer, Inc.]
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: 64
    Maximum number of private, static segments: 64
-----------------------------------------------------
```

```
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: 75
    Maximum number of private, static segments: 75
Project "DEALERS" updated 03 June 88 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.

Format

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. (For the explanation of a current project, see the ATTACH_PROJECT command earlier in this chapter.)

Example

In the following example, the System Administrator deletes the project DUMMY.

```
OK, EDIT_PROFILE
[Edit_Profile Rev 22.0 Copyright (c) 1988, Prime Computer, Inc.]
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 10 June 88 11:12:44.
    (5 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.

Format

DETACH_PROJECT [*project-id*]
DTP

You need not specify *project-id* to detach 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.

Format

LIST_PROJECT [*project-id* [*options*]]
LP

project-id is the project to be listed. You must specify *project-id* to use an option.

Options

-ALL
Lists the profiles of all project members.

-APPEND
-APP
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*
-OUT
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
-PROF
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

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 22.0 Copyright (c) 1988, Prime Computer, Inc.]
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: 64
    Maximum number of private, static segments: 64
-------------------------------------------------------

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: 64
    Number of private, static segments: 64
*************************************************************************
>
```

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.

Format

ADD_USER [*user-id* [*options*]]
AU

user-id is the user to be added. You must specify *user-id* to use an option.

Options

-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*]
-PW

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.)

-PASSWORD_LIFETIME [*value*]
-PWLIFE

Specifies the lifetime of a user's password. Specify any positive number ranging from 1 through 99,000 for *value* to set the password duration in days. Specify -1 for *value* to provide an infinite password lifetime. Specify 0 for *value*, or omit *value* entirely, to use the system default password lifetime. (See the DEFAULT_PASSWORD_LIFETIME command earlier in this chapter.)

-PROFILE
-PROF

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*]
-PROJ

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
-SYS

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 node names of the systems where identical user IDs were found. The -VERIFY_NS option helps prevent duplication of user IDs across the network.

Discussion

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***]
-DL

Example

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 SA creates the user profile.

The System Administrator makes ALFRED a member of the group .KINGS. The SA presses Return to accept the system default for ALFRED's password lifetime. After checking the entry, the SA decides that ALFRED should also belong to the group .EARLS, and therefore types YES at the `Change Entry?` prompt. After the SA adds the group .EARLS to ALFRED's groups, ALFRED belongs to .KINGS and .EARLS.

```
OK, EDIT_PROFILE
[Edit_Profile Rev 22.0 Copyright (c) 1988, Prime Computer, Inc.]
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
    Password lifetime in days (-1 = infinite, 0 = default):   Return

User "ALFRED" added to system.
Check entry? YES

*************************************************************************
System-wide attributes for user "ALFRED":
    Groups: .KINGS
    Default login project: SAXON
    Current password lifetime:  default
*************************************************************************
Change entry? YES

System-wide attributes for user "ALFRED":
    Groups: .KINGS
    Default login project: SAXON
    Current password lifetime:  default

Set system-wide attributes for user "ALFRED":
    Groups: -ADD .EARLS
*** New group added to system: ".EARLS".
    Default login project:
User "ALFRED" updated 10 June 88 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.

Format

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.

Options

-LIST
Lists the user's attributes after the changes have been made.

-PASSWORD [*password*]
-PW
Specifies a new login password for the user. If you omit *password*, you are prompted for it.

-PASSWORD_LIFETIME [*value*]
-PWLIFE
Specifies the lifetime of a user's password. Specify any positive number ranging from 1 through 99,000 for *value* to set the password duration in days. Specify -1 for *value* to provide an infinite duration to the user's password. Specify 0 for *value*, or omit *value* entirely, to use the system default password lifetime. (See the DEFAULT_PASSWORD_LIFETIME command earlier in this chapter.)

-PROJECT [*project-id*]
-PROJ
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
-SYS
Specifies that you are changing the user's systemwide groups, default login project, or command environment attributes.

Discussion

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]***
-DL

Example

The first part of the following example shows how the System Administrator uses the CHANGE_USER command to change the attributes of user CHRIS in project PARTS.

The system first responds to the command line with a display of the present attributes for user CHRIS within the project PARTS. The prompt to set new attributes for this user pauses for input at `Groups:`. Because the System Administrator presses Return after this and the next prompt, the values for `Groups` and `Initial Attach Point` do not change. In order to change the command environment limits of CHRIS, the SA replies YES to the `Create/change user attributes?` prompt.

The command environment limits for the project are displayed, to remind the SA that the user cannot be assigned limits that exceed those of the project. The SA then enters the new command environment limits for user CHRIS.

The second part of the example shows how the SA gives a thirty day lifetime to this user's password. The SA issues a second CHANGE_USER command with the option -PASSWORD_LIFETIME followed by the value 30.

```
OK, EDIT_PROFILE
[Edit_Profile Rev 22.0 Copyright (c) 1988, Prime Computer, Inc.]
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:  Return
    Initial attach point:  Return
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 09 June 88 10:41:52.
> CHANGE_USER CHRIS -PASSWORD_LIFETIME 30
    Current password lifetime: default
    New password lifetime value: 30 days
User "CHRIS" updated 09 June 88 10:42:37
>
```

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.

Format

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.

Options

If you do not specify the -PROJECT option (abbreviated as -PROJ) 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

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" 23 June 88 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" 23 June 88 11:05:56.
>
```

In the third example, the System Administrator removes user JIMMY from the system.

```
> DELETE_USER JIMMY
*** User "JIMMY" deleted from system 23 June 88 11:07:00.
*** User "JIMMY" deleted from project "EDUCATION" 23 June 88 11:07:00.
*** User "JIMMY" deleted from project "BAD_BOYS" 23 June 88 11:07:04.
    (Project "BAD_BOYS" is now empty.)
*** User "JIMMY" deleted from project "DEFAULT" 23 June 88 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 projects

Format

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 of the options listed below unless you specify *user-id* on the command line.

Options

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, only the systemwide attributes of the user are listed.

-ALL
Lists the user's attributes systemwide and in all the user's projects.

-PROJECT [*project-id*]
-PROJ
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

In the following example, the LIST_USER command lists the attributes of user CAROL in all her projects.

```
OK, EDIT_PROFILE
[Edit_Profile Rev 22.0 Copyright (c) 1988, Prime Computer, Inc.]
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 node names 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.

Format

VERIFY_USER {*user-id* | **-ALL**}
VU

If you specify *user-id*, the SADs of the other systems are searched only for that ID and, if that ID is 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 must 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 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 22.0 Copyright (c) 1988, Prime Computer, Inc.]
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.

Format

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.

Options

The four Project Administrator options are -LIKE, -NO_QUERY, -PROFILE, and -PROJECT. For details on these options, see the ADD_USER command in the section User-Control Commands earlier in this chapter.

The CHANGE_PROJECT Command in Project Administrator Mode

Use the CHANGE_PROJECT command to change the profile of your project.

Format

CHANGE_PROJECT [*project-id* [*options*]]
CP

Options

The three Project Administrator options are -LIST, -PROFILE, 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.

Format

CHANGE_USER [*user-id* [**-PROJECT** [*project-id*]]]
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.

Options

The -PROJECT option (abbreviated as -PROJ) 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.

Format

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.

Option

The -PROJECT option (abbreviated as -PROJ) 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 attributes of either one or all users in the project. The list always includes the project limits imposed by your System Administrator.

Format

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.

Options

For details on the options, see the LIST_PROJECT command described in the section Project Commands earlier in this chapter.

The LIST_USER Command in Project Administrator Mode

Use the LIST_USER command to display the attributes of an individual member of your project.

Format

LIST_USER [*user-id* [*options*]]
LU

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.

Options

The two options for the LIST_USER command are listed below. You cannot specify both options at the same time.

-ALL
Displays the user's attributes in each project to which the user belongs, provided that you administer that project.

-PROJECT [*project-id*]
-PROJ
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.

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.

Format

REBUILD [**-PROJECT** [*project-id*]] [**-SIZE** *entry-count*]
RE

Options

Use the -PROJECT (abbreviated as - PROJ) 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.

7

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 2, 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

To ensure that the equipment (which includes terminals, printers, and modems) is secure, you must keep track of its use outside the computer room. 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 offsite. 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 ACL system.

Security Control by Passwords

Before Rev. 19.0, both login security and data security were maintained by a system of passwords. 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. 19.0, the User Profile system controls login security (system access) and Access Control Lists (ACLs) control 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 to 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.
- The login password may be given a limited lifetime to avoid its inadvertent disclosure through use over an extended time period.
- The login password may be computer-generated to further ensure against inadvertent disclosure.
- Passwords are recorded in the machine in an encrypted (scrambled) form, so that they cannot be read by humans or be easily decoded.
- If required, passwords can still be set on directories.

LOGIN SECURITY

Starting at Rev. 19.0, the User Profile system provides login security. The User Profile system consists of a database that you build with the EDIT_PROFILE utility. (EDIT_PROFILE is explained in Chapter 6, Using EDIT_PROFILE.)

The User Profile Database 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 database and determines if the user should be allowed access to the system. If the user is allowed access, the database also provides 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. 19.0 systems, you should convert them to take advantage of the more powerful security features available after Rev. 20.0. 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 Database. See Chapter 4, Planning the User Environment, for the information associated with a user ID.

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 disadavantage 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 Packet Switched Data Network (PSDN). 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. You must decide two questions about how login passwords are to be used on your system:

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

These two issues are discussed in the next two sections. See Chapter 4, Planning the User Environment, for strategies on the assignment of 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. 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. The -ON option is the default.

Note

NO_NULL_PASSWORD -ON is required for systems to maintain a C2-certified level of security.

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 (that 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 thus is 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.

You are urged to enable the non-echoing of passwords by using the FORCE_PASSWORD command of EDIT_PROFILE. If your system is to maintain a C2-certified level of security, you must enable non-echoing. 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 are. The only remedy is to use EDIT_PROFILE to assign new passwords.

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

Requiring That Users Change Passwords: The System Administrator may also use certain EDIT_PROFILE subcommands to require users to change passwords.

The SA uses the -PASSWORD_LIFETIME option with either the ADD_USER or CHANGE_USER command to establish a password lifetime for an individual user. The lifetime may be set for a number of days (1 through 99,000) or for infinity (-1). If the lifetime is left unset, the user's password lifetime is defined by the system default password lifetime.

The SA uses the DEFAULT_PASSWORD_LIFETIME command to establish the system default password lifetime. The lifetime may be set for a number of days (1 through 99,000) or for infinity (-1). If the lifetime is left unset, the system default password lifetime is infinity. (See the section entitled Examples of User Validation at Login later in this chapter.)

Computer-generated Passwords: The System Administrator may issue the EDIT_PROFILE subcommand COMPUTER_GENERATED_PASSWORDS -ON to generate and issue a password when an old password is being changed. As a default, computer-generated passwords are inactive (-OFF).

Project IDs

Every user must be registered in the User Profile Database 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 Database.

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 6, 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 user 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 that user's 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.

Notification of Failed Logins

At Rev. 22.0, PRIMOS provides an automatic security feature to each user. Immediately after logging in, a user is notified about any failed attempts to log in under this user ID. The count of failed logins is then immediately reset to zero. The SA should inform users of this failed login notification feature, assuring them that the notification does no harm, but is meant to keep them aware of attempts to illicitly enter the system. Users must be urged to inform the SA if they cannot account for the number of failed logins attempted under their user IDs.

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, a password lifetime, computer-generated passwords, and a project ID provides the most system-supplied security. It is possible, however, to have too much system-supplied security.

The features of password lifetimes and computer-generated passwords are meant to provide greater system security, but overuse of them may actually cause an unnecessary nuisance or even a possible security breach.

For example, the computer-generated password feature is especially designed to protect systems from outside infiltration via modem connections. The use of computer-generated passwords on a localized system may lead a user to unknowingly reveal a password, because the user may be forced to write out a more-difficult password and keep it near the user's terminal. For another example, should you establish too short a password lifetime, the user may grow tired of selecting a new password each time. The user then may begin to merely alternate the same two passwords. The System Administrator should periodically remind users to select new passwords carefully. The SA should also maintain a reasonable password lifetime (60 to 90 days under normal conditions).

Network Security

For information on setting up security over a network, see the *PRIMENET Planning and Configuration Guide.*

DSM and System Security

Distributed Systems Management (DSM) provides a security management umbrella over distributed systems. The multiple systems (or a single one) are unified under a single management configuration. Therein a user with IDs on several systems may receive different functional rights on each of those systems. These functional rights, or roles, supplement the ACL rights granted users on a particular system. Note that DSM role rights do not override system ACL exclusions. However, note that an ACL permission on your system may be cancelled by a DSM role exclusion.

Since a System Administrator might not be the DSM Administrator, the SA must know how DSM security affects the local system. Since controlling local security is the SA's duty, local control of DSM always remains active at the supervisor terminal with the use of the commands START_DSM and STOP_DSM. Furthermore, various levels of control for an active DSM may be defined for each node within the DSM Configuration File (CF). The SA should add a hard copy of the CF to the collection of other system logs.

The CF first lists all registered DSM function names. The SA will normally find each supervisor terminal (as SYSTEM) included in the DSM_ADMINISTRATOR$ role, with access to every DSM function. If the CF has been configured differently, the SA must be sure that, at the very least, SYSTEM has access to PRIVATE_LOGGER. The DSM Administrator supports the SA's local administration of security; no conflict need exist between local system administration and DSM administration.

Configuration Directives and Security

CONFIG directives in the system configuration file have an important impact on the security of the system, especially those directives in the following list. Most of these are concerned with service to remote users, connected either by network or by modem. For a full description of these directives, see the *System Administrator's Guide, Volume I: System Configuration.*

CONFIG Directive	*Description*
AMLTIM	Sets time intervals for modem disconnect operations.
DISLOG	Enables or disables logout when a line is disconnected.
DTRDRP	Controls the dropping of the DTR (Data Terminal Ready) signal associated with an asynchronous line.
LOUTQM	Specifies the interval of inactivity before a user is automatically logged out.
LOTLIM	Specifies the time interval within which a user must complete the login procedure.
LOGBAD	Specifies whether unsuccessful login attempts should be reported at the supervisor terminal.
NRUSR	Specifies the number of remote users.
NSLUSR	Specifies the number of slave users.

Note

Systems that must maintain a level of C2-certified security do not support remote users. The System Administrator achieves this by eliminating both the NRUSR and the NSLUSR directives from the configuration file.

The Login Server

The Login server handles all terminal login attempts for local and remote users. The Login server starts when the system is booted. If, for some reason, the Login server either 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 *Operator's Guide to System Commands* for a description of this command.)

The Login server has no interaction with users who are logged in and who have had their lines associated with a process or another 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	*Meaning*
DATE	Displays the current calendar date and clock time.
DELAY	Defines a time function that delays printing a character after a carriage 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 supervisor terminal to function as a user terminal (only on systems where the VCP command MO USER is unavailable).

For more information on these commands, see the *PRIMOS Commands Reference Guide.*

As the System Administrator, you need to know the following facts about the Login server.

- You need not configure (with the NPUSR configuration directive) a phantom for the Login server or for other system servers (with the exception of NETMAN, ROUTE-THROUGH, and BATCH_SERVICE).
- 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 5, Setting Access Rights.

- 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. Do *not* create a separate entry for the Login server in the SAD.
- Use the supervisor terminal command START_LSR to start the Login server. Use the supervisor terminal command STOP_LSR to stop the Login server. (See 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 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. 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 not running 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 exist.
  - 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 issue the SET_SEARCH_RULES (SSR) command. If an error occurs, error messages are displayed to assist you. If necessary, fix the search rules, reissue the SSR command, and restart the Login server with the START_LSR command.
  - If problems persist, check SERVERS*>LOGIN_SERVER.ENTRY$.SR for typographical errors.
  - If the Login server does not begin at cold start, verify that SERVERS* is on the boot device.

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 database to verify that the person attempting to log in is an authorized user of the system. It also notes the values recorded for password lifetime — both the system default lifetime and the individual password lifetime.

   If the user fails validation, the login procedure terminates and steps 2 through 5, described below, are not performed. Each failed login for a particular user ID is recorded. When that user ID next logs in successfully, a warning message indicates the number of failed logins since the last successful login. The count is then reset to zero.

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

2. After a user ID passes login validation, PRIMOS establishes the user's membership in ACL groups that are active during this session.

   PRIMOS checks to determine whether computer-generated passwords are active and whether a password lifetime has expired. The status of computer-generated passwords is checked first. The individual's password lifetime is then verified (before the system default password lifetime). If no change has occurred in the status of computer-generated passwords, and if no password has expired, PRIMOS proceeds to step 3.

   If computer-generated passwords have been activated since the user's last login, the user is prompted for a password change, even if no password lifetime has expired. The prompt is similar to that shown in Example 1, save for the absence of the first two lines.

   If the individual's password lifetime has expired, the user is prompted to change the login password (see Example 3). If no individual password lifetime is active, then the system default password lifetime is checked. If that lifetime has expired, then the user is prompted to change the login password (see Example 1). The notification prompt changes format, depending on the status of computer-generated passwords. Example 1 illustrates the prompt when CGPW is set to -ON within EDIT_PROFILE. Example 3 illustrates the prompt when computer-generated passwords are inactive.

3. PRIMOS searches for LOGIN, a site-supplied login program located in CMDNC0. If LOGIN exists, PRIMOS executes it. If the program does not exist, PRIMOS proceeds to step 4.

   This optional external login program (that is, outside the SAD) must be a static-mode program named LOGIN and must reside in CMDNC0. No suffix is allowed in the name of this program. 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.

4. PRIMOS attaches the user to the user's Initial Attach Point (IAP), which is the user's origin directory for all ensuing activities.

5. PRIMOS searches the user's IAP 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 an abbreviation file, setting terminal characteristics, and executing other programs or commands. For more information on user login programs, see the *PRIMOS 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) depend on EDIT_PROFILE settings and the information supplied by the user on the command line.

The next four 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 database illustrated in Figure 7-1 and assume that null passwords and the entry of passwords on the command line are allowed. Example 1 assumes that computer-generated passwords are active, while Examples 2 and 3 assume a setting of CGPW -OFF.

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 database.
2. Verifies that GREEN is the password associated with FROG.
3. Notes that at this time computer-generated passwords are activated. Notes that the user password lifetime defaults to the system default value, and verifies that this password lifetime has just expired. (See final step for dialog that results from these notes.)
4. Checks the project database for project SWAMP and finds FROG listed as a member.
5. Checks project SWAMP's database for FROG's IAP. The directory is <SWAMP>LILYPAD. The system attaches FROG to that directory.
6. Checks the system database for systemwide ACL groups for FROG. It finds one (.AMPHIB) and marks FROG as a member of that group.
7. Checks project SWAMP's database 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.
8. Finally issues a notification to FROG that the password has expired. Because computer-generated passwords are presently activated, the notification and prompts take the following format:

```
Your password has expired.

Computer generated passwords are in effect.
Please ensure that you can view your new password in privacy.
Type RETURN to continue: [Return]

Your new password is ARIBIT
Reenter new password for confirmation: <new password not echoed>
Your new password has been confirmed.
```

System Database

Status of computer-generated passwords Value for system default password lifetime
ID: FROG Passwd: GREEN PWLIFE: (default) Def. Proj: ACL Groups: .AMPHIB
ID: POSSUM Passwd: PWLIFE: Def. Proj: SWAMP ACL Groups:
ID: PIG Passwd: BEAUTIFUL_STAR PWLIFE: 120 Def. Proj: ACL Groups: .VIPS .PIGS .BEAUTIES

Project SWAMP Database

Default IAP: Default ACL groups:
ID: FROG IAP: <SWAMP>LILYPAD ACL Groups: .FLYCATCHERS .MUSICIANS
ID: POSSUM IAP: <SWAMP>TREE ACL Groups: .POSSUMS

Project HOLLYWOOD Database

Default IAP: <MOVIES>HOLLYWOOD Default ACL groups: .STARS
ID: FROG IAP: ACL Groups:
ID: PIG IAP: ACL Groups: .STARS .SUPERSTARS

FIGURE 7-1. Sample Portion of User Profile Database

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 database.
2. Checks for POSSUM's password, and finds that the password is null. Because the password is null, the system does not prompt for the password.
3. Since null passwords nevertheless have a password lifetime, the system continues to check for password lifetime and the status of computer-generated passwords. Notes that at this time computer-generated passwords are inactive. Notes that neither individual nor system default password lifetimes have expired.
4. 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.
5. Checks project SWAMP's database and finds POSSUM listed as a member with <SWAMP>TREE as a private IAP.
6. Attaches POSSUM to <SWAMP>TREE.
7. Finds no systemwide ACL groups for POSSUM in the system database.
8. Checks project SWAMP's database and finds that POSSUM is a member of the ACL group .POSSUMS. For this session .POSSUMS becomes POSSUM's only ACL group.

Example 3: Only the user ID and login password are supplied on the command line. The command is

```
LOGIN PIG BEAUTIFUL_STAR
```

The system does the following:

1. Finds the user ID PIG in the system database.
2. Verifies that BEAUTIFUL_STAR is the password associated with PIG.
3. Notes that at this time computer-generated passwords are inactive. Notes that the user password lifetime for PIG has just expired. (See final step for dialog that results from these notes.)
4. Since the command line indicates no project, checks the system database to see if PIG has a default project. The default project for PIG is null, so the system prompts for a Project ID, to which the user responds as follows:

   ```
   Project ID: HOLLYWOOD
   ```

5. Checks project HOLLYWOOD's database and verifies that PIG is a member. Checks for PIG's individual IAP.
6. The individual IAP for the user ID PIG is null, so the system identifies the Default IAP for this project: <MOVIES>HOLLYWOOD. The system attaches PIG to that directory.

7. Checks the system database for systemwide ACL groups for PIG. It finds three (.VIPS, .PIGS, and .BEAUTIES) and marks PIG as a member of each group.
8. Checks project HOLLYWOOD's database for project-specific ACL groups for PIG. It finds two (.STARS and .SUPERSTARS) and adds them to the list of other ACL groups for PIG during this session.
9. Finally issues a notification to PIG that the password has expired. Because computer-generated passwords are presently inactive, the notification and prompts take the following format:

```
Your password has expired; please change it.
New password: <new password not echoed>
Reenter new password for confirmation: <new password not echoed>
Your new password has been confirmed.
```

Example 4: Only the LOGIN command is supplied on the command line. 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 Database and does not find ORK.
3. Prompts for a password. The user responds: FABLE.
4. The system terminates login and issues 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 more secure and easier to use.

Access Control Lists

Access Control Lists (ACLs) are the cornerstone of the file system access control mechanism. Users with Protect (P) 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. You can override this default protection by setting a specific ACL on a lower level file or directory with the SET_ACCESS command.

An ACL can provide access control both for named individual users and user groups. 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 Database, 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) 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 the *PRIMOS User's Guide* or Chapter 4 of this manual, Planning the User Environment. For guidelines to setting system-level ACLs see Chapter 5, Setting Access Rights.

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 5, Setting Access Rights.

Password Directories

PRIMOS allows users to create directories as ACL directories or password directories. Access to password directories is controlled by owner and nonowner 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 *PRIMOS User's Guide.*

COORDINATING LOGIN SECURITY AND DATA SECURITY

Because login security and data security can both be handled through the User Profile Database, 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.

There are three general systems of security control:

- 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 and device ACLs set on appropriate equipment. 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. Different printers and magnetic tape units are accessible to different subgroups within applications development.
- 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. 19.0 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 7-2a and 7-2b diagram a loosely controlled system.

Figure 7-2a shows that within the SAD all system users are listed as belonging to project DEFAULT. No other projects are in use. No Project Administrators are required.

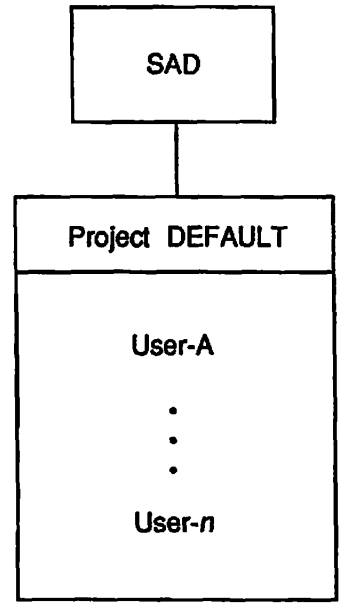

FIGURE 7-2a. Loosely Controlled System

Figure 7-2b shows that ACLs can govern the rights of individuals or groups at any level of the file hierarchy, from the MFD to an individual file.

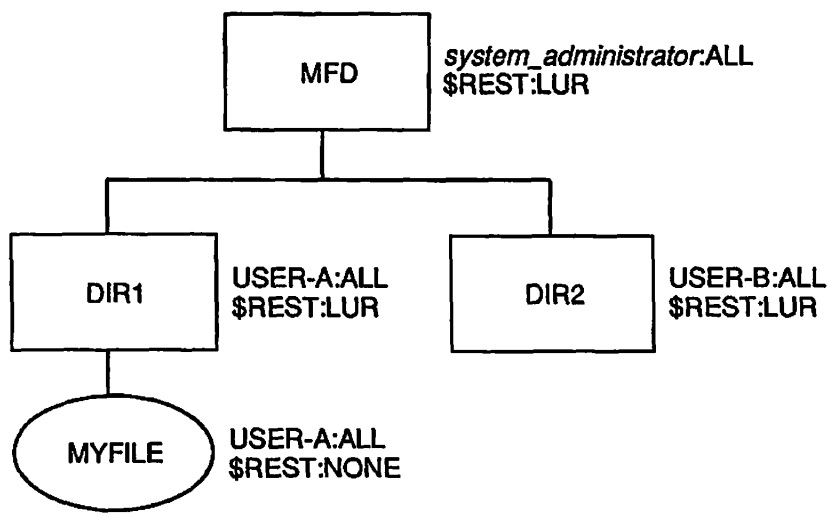

FIGURE 7-2b. ACLs on a Loosely Controlled System

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. Furthermore, the system uses device ACLs to limit the availability of a device to a defined set of users.

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 database 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 7-3a, 7-3b, and 7-3c diagram a tightly controlled or project-based system. In this example, project DEFAULT is not in use.

Figure 7-3a shows that every user must log in to a specific project. Within the SAD a user has membership in only one project.

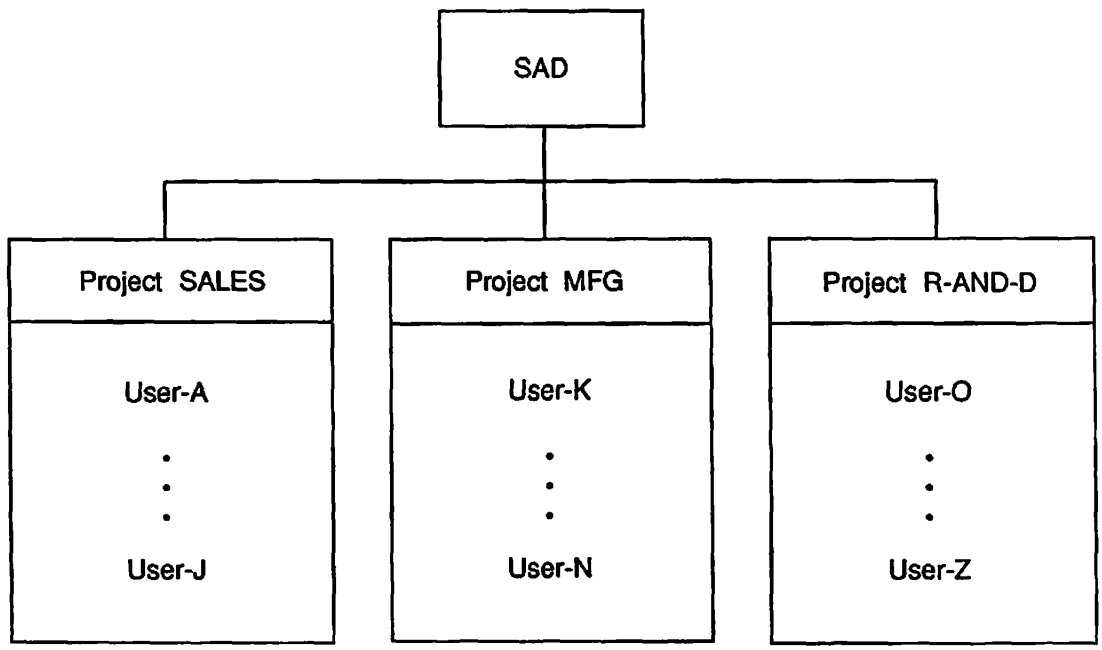

FIGURE 7-3a. Tightly Controlled or Project-based System

Figure 7-3b shows that project-based ACL groups can be used to deny non-project members access to project files and directories. The two first level directories have names to suggest project organizations, SALES and MFG. System users within each project have personal Initial Attach Points as subdirectories under these project names. Users may belong to several projects, but their IAP and their access rights during any session depend on the project ID they specified at login time. Project Administrators may perform limited security functions.

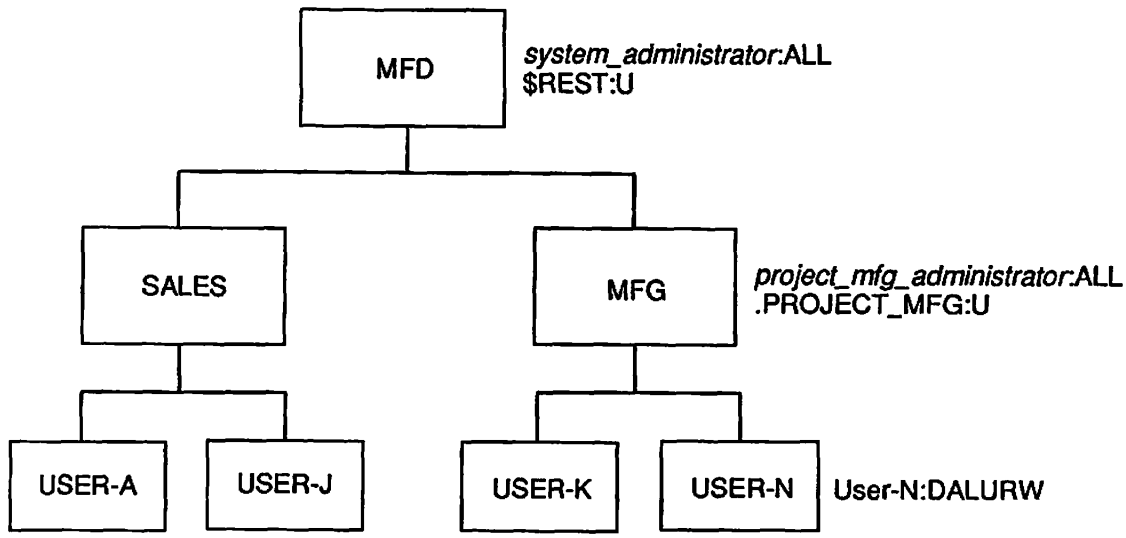

FIGURE 7-3b. ACLs on a Tightly Controlled System

Figure 7-3c shows device ACLs set to limit user access to certain devices. Since members of the ACL group .PROJECT_SALES have exclusive access to the printer and magnetic tape units, PR0 and MT0, these devices are exclusively assigned to the users in project SALES. Likewise, members of the ACL group .PROJECT_MFG have exclusive access to devices PR1 and MT1. The devices are reserved for users under the project MFG.

While the Project Administrator may add members to or delete members from the project's ACL group, only the System Administrator may set the ACL on the subdirectory corresponding to each device.

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:

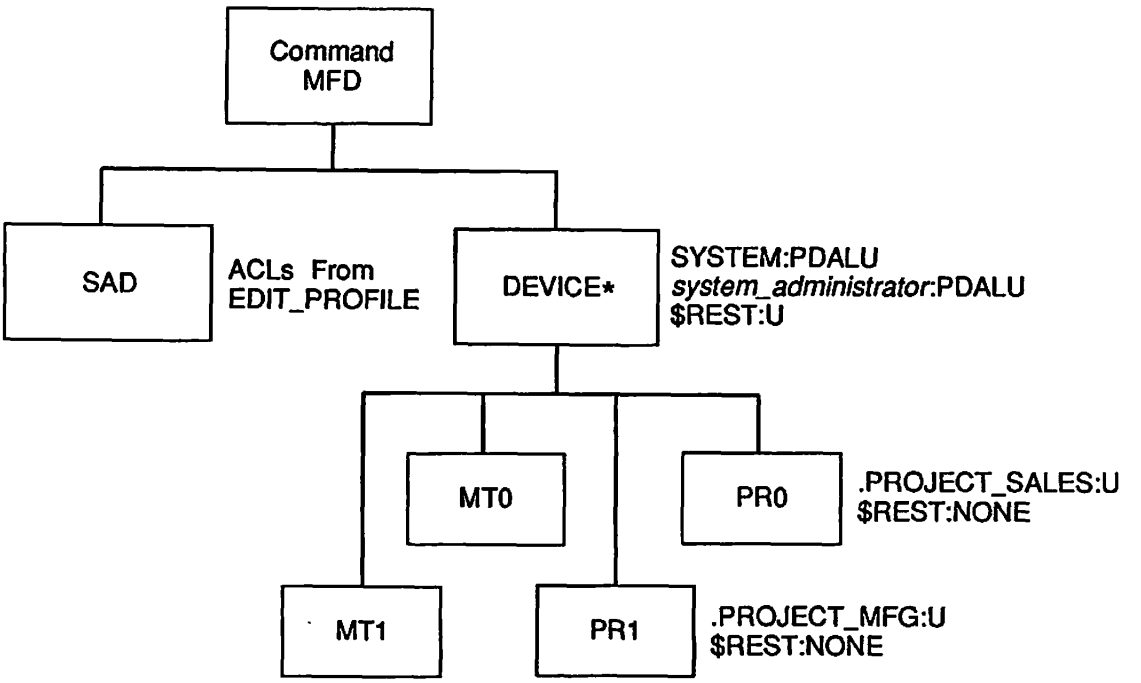

FIGURE 7-3c. Device ACLs on a Tightly Controlled System

- Some users belong to project DEFAULT and are given broad 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 broad 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 7-4 diagrams a mixed system. 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.

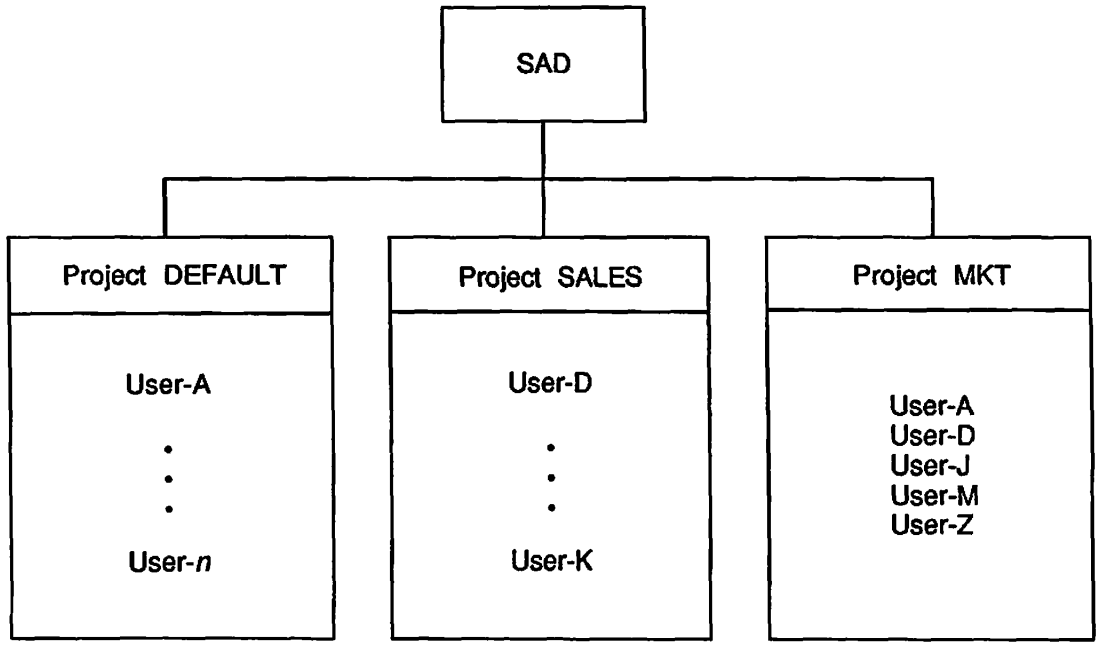

FIGURE 7-4. Mixed System

SECURITY FOR C2-CERTIFIED SITES

Customers who have purchased the Security Audit facility have also received the CONVERT_TO_ACLS utility. Use of this utility fulfills the C2 requirement to eliminate all password directories from partitions on the system.

The CONVERT_TO_ACLS Utility

The CONVERT_TO_ACLS utility may be used as a command by the System Administrator or at the supervisor terminal, if search rules to the directory TOOLS have been properly set. (See the section Security Measures During Installation later in this chapter.) Before using the utility, you must convert all password MFDs on the system.

- Attach to an MFD and give an LD command. MFDs governed by ACLs will indicate the ACL rights as part of the first display line.
- If no ACLs are listed, you can convert the MFD from password format by entering SET_ACCESS (abbreviated as SAC), followed by whatever access lists are relevant to that MFD.

The MFD is then ready to be submitted to CONVERT_TO_ACLS, which will convert all password subdirectories to ACLs.

Format

CONVERT_TO_ACLS [*options*]

Options

-DEFAULT
Applies the ACLs on the MFD to those password subdirectories being stripped of their passwords. If you supply no option, this is indeed the default.

-NONE
Elicits a prompt for the name that will be granted ALL rights to the password subdirectories that are being coverted to ACL directories. The System Administrator is normally the one to receive ALL rights.

Discussion

This utility requires the user to have rights to set a priority ACL. Therefore you as SA (or an operator with priority ACL rights) must run the utility, using your own user ID. If you run the utility at the supervisor terminal, you may use a default priority ACL for SYSTEM.

The utility automatically removes the priority ACLs when the conversion is complete.

The example below illustrates the use of the CONVERT_TO_ACLS utility.

```
OK, CONVERT_TO_ACLS -DEFAULT

Enter a list of local partitions to be converted,
separated by a blank.
EXAMPLE: OSGRP1 DMDATA TECHPB

Partition names: OP_SYS TELLER BANKIT

Enter user id for priority ACL (carriage return = SYSTEM).

User id:
*** From PRIMOS: Priority ACL set on partition "OP_SYS"
    by user "SYSTEM" (#1) at 01 June 88 07:05:03

ACL protecting "<OP_SYS>MFD>MFD"
        IRENE:    ALL
        SYSTEM:   PDALURWX
        .ADMIN:   DALURWX
        $REST:    LUR

Priority ACL in effect for "<OP_SYS>MFD>MFD":
        SYSTEM:   ALL
```

```
"<OP_SYS>MFD>CMDNCO" protected by default ACL (from "<OP_SYS>MFD"):
        IRENE:    ALL
        SYSTEM:   PDALURWX
        .ADMIN:   DALURWX
        $REST:    LUR
Priority ACL in effect for "<OP_SYS>MFD>CMDNCO":
        SYSTEM:   ALL
          .
          .
          .
```

Note

Since the utility must scan every directory in the tree structure to find the password directories that need conversion, the task is time-consuming. If the conversion process is aborted for any reason, verify that all files are closed and that the priority ACLs have been removed.

Security of System Hardware

If you are an administrator of a system that maintains a C2-certified level of security, you are responsible for carrying out the following security measures for system hardware.

- Be sure that your system includes at least one magnetic tape drive unit. You need a tape drive to process tape dumps with the CRASH_AUDIT utility.
- Keep the supervisor terminal and a magnetic tape unit (for tape dumps) in a protected room.
- Never leave a terminal unattended while logged in as System Administrator.
- Store under lock all the disk and tape media used on the system. Never leave the storage area unlocked and unattended.
- Establish and enforce strict tape and disk security handling procedures for personnel in charge of media storage and distribution.
- Do not mount your disk packs on any other system, even under secured conditions, since the packs may receive password directories on the other system. If such a procedure is unavoidable, run the CONVERT_TO_ACLS utility on the returned disk pack before allowing user access.
- Submit all scratch disk and tape media to a bulk erasure before making them available for reuse.
- Bulk erase media before disposal. Media that held very sensitive data should be erased and incinerated.

Security of System Software

If you are an administrator of a system that maintains a C2-certified level of security, you must provide the following security measures for system software.

Installation of C2–certified Software: The following Prime products, some of which are separately priced, have been approved for use on systems maintaining strict C2 security: EMACS, DBG, all compilers and language interpreters, PRIMAN, and PRIMEWAY™. All utilities and commands, for example, LD and BRMS, on the Master Tape may be used.

If PRIMEWAY is used, the system must not have terminals generally available to users on which PRIMOS login is possible. In other words, all terminal lines not assigned by PRIMEWAY must be disabled or must be within the boundary of physical security in the computer room.

Other Prime database products were not included in the evaluation at Rev. 21.0.1, and so must not yet be used on a system maintaining strict C2 security. These products are DBMS, MIDASPLUS™, PRISAM, Prime INFORMATION™, and ORACLE™.

Security Measures During Installation: The following summarizes suggested security measures during installation.

- Create private search rules to TOOLS for both the supervisor terminal and the System Administrator, as follows:
    - At the supervisor terminal
        1. Issue the ORIGIN command to ensure that User 1 is in CMDNC0 (normally the IAP for the supervisor terminal).
        2. Using ED (or EMACS at a video display terminal), create a file named TOOLS.COMMAND$.SR, and put in the file the single line

           ```
           TOOLS
           ```

        3. Issue the following command line to add the TOOLS directory to the COMMAND$ search rules:

           ```
           SSR TOOLS.COMMAND$.SR
           ```

        4. To verify that the TOOLS directory has been added to other COMMAND$ search rules for this terminal, issue the command line

           ```
           LSR COMMAND$
           ```

        5. Within the PRIMOS.COMI file add this line before the MAXUSR command:

           ```
           SSR TOOLS.COMMAND$.SR
           ```

           Thereafter you no longer need to set this search rule manually.

- For the System Administrator

  1. While logged in at a user terminal, issue the ORIGIN command to attach to your IAP.
  2. Follow steps 2 through 4 above.
  3. Within your LOGIN.CPL file or LOGIN.COMI file add the line

     ```
     SSR TOOLS.COMMAND$.SR
     ```

     Thereafter you no longer need to set this search rule manually.

- If you have used a previous revision of PRIMOS, convert all password-protected directories to ACL-protected directories. To do this, follow the directions in the section Adding Partitions later in this chapter, and apply the cold start directions to each partition on your system.
- Set device ACLs on each device name that exists as a subdirectory to DEVICE*. Use standard ACL commands to provide only U rights to authorized device users. Create under DEVICE* those additional subdirectories needed on your system for assignable disks and asynchronous lines, and set the proper ACLs on them.
- Ensure that the directory TOOLS contains CREATE_NUMSEM*.CPL.
- To guarantee complete security audits, include the following commands in the PRIMOS.COMI file *before* the MAXUSR command:

  ```
  CO SYSTEM>C2.INIT.COMI  7
  START_DSM
  SECURITY_MONITOR -START [options]
  ```

  The C2.INIT.COMI file turns on device ACLs (with DEVACL -ON) and disallows the creation of new password directories (with PWDIR -OFF). The command input file also ensures that the system has a directory NUMSEM* and that ACLs on numbered semaphores have been activated. (Refer to Appendix D for details.) The START_DSM command activates Distributed Systems Management (DSM). After the SECURITY_MONITOR -START command, you may append those options most suitable to your own system. (See Chapter 11, Security Audits.) If you choose not to run security audits after cold start, issue the command SECURITY_MONITOR -STOP at the supervisor terminal after the system has completely booted. For an alternative to stopping the facility, see Maintaining the Security Audit Facility later in this chapter.
- Since systems that maintain a strict level of C2-certified security do not use a network, you may exclude the two configuration directives NRUSR and NSLUSR. Also exclude START_NET from your PRIMOS.COMI file. Create a directory for security audits that allows access to the Security Administrator, the System Administrator, and the security audit phantom AUDITOR.

- Within EDIT_PROFILE
  - Maintain (the default) NO_NULL_PASSWORD -ON.
  - Maintain (the default) FORCE_PASSWORD -ON.
  - As a security safeguard, specify at least five characters for the MINIMUM_PASSWORD_LENGTH command.

- Set ACLs on the following runfiles as indicated below:

```
SAC CMDNCO>LIST_UNITS.RUN SYSTEM:ALL system_administrator:ALL $REST:NONE
SAC CMDNCO>LIST_PROCESS.RUN SYSTEM:ALL system_administrator:ALL $REST:NONE
```

Security Measures During Operations: The following summarizes suggested security measures during operations.

- Before adding a partition to the system, remove all password directories on that partition. See Adding Partitions later in this chapter for details.
- To ensure complete security audits
  - Maintain a RING0.MAP file on the system. At Rev. 22.0 this is normally located in the new directory LOAD_MAPS*. (Maps for previous revisions of PRIMOS may already exist in the directories PRIRUN or MAPS.)
  - Take a tape dump after every unplanned system halt before the next cold start of the system. See your CPU handbook for tape dump directions.
  - After you cold start the system, submit the tape dump to the CRASH_AUDIT utility.
  - This procedure ensures that buffers left open on security audits can be retrieved and written out to another audit file. For directions on using CRASH_AUDIT, see Chapter 11, Security Audits.
- Withhold the password for SYSTEM from operators to discourage their use of it as a user ID for administrative purposes.
- Disallow BATCH_SERVICE as an active user ID.
- Never use the RESUS command to enable a remote supervisor terminal.
- Do not modify the ACLs set on the SAD. These ACLs, established by EDIT_PROFILE when the SAD was created, should remain

  LOGIN_SERVER:ALL
  system_administrator:ALL
  $REST:LU

- Never use the command NUMSEMACL -OFF while users are on the system or allowed to log in.

- Maintain all security audit files in directories that prohibit R (Read) and W (Write) access to any user other than the System Administrator and the supervisor terminal. This rule especially applies when the tape or partition holding an audit file becomes full and requires the SA to switch to another audit file while the security monitor is still running.
- Do not shut down a partition that contains an open security audit file until you have switched to another audit file. Similarly, if the active security audit file is on tape, do not take the tape offline without first switching the audit file.

Adding Partitions: Defer adding partitions until a cold start. At that time, do the following *before* you issue the MAXUSR ALL command:

1. Add the partitions.
2. Manually remove the password on each MFD (if necessary).
3. Run CONVERT_TO_ACLS on the partitions. (See The CONVERT_TO_ACLS Utility earlier in this chapter.) Be prepared for a long report of all the resulting ACLs set by this utility. If the conversion process is aborted for any reason, ensure that all files are closed and that the priority ACLs have been removed.

If you must add the partition immediately, follow the same three steps above, but first warn users of the conversion process. All directories will be converted successfully, but a user in the process of entering a password directory at the moment of its conversion to an ACL directory may experience some inconvenience and confusion.

Maintaining the Security Audit Facility

Every C2-certified system must be able to run the Security Audit facility. You ensure this by inserting the command SECURITY_MONITOR -START into your PRIMOS.COMI file. Thereafter the System Administrator either may tune the facility to audits of specific objects or may decide to turn it off. See Chapter 11, Security Audits, for details on how to tailor the audit facility to your own needs and for descriptions of all the commands and utilities used to operate and maintain the Security Audit facility.

In order to record failed logins, you must audit the following:

- Privileged operations
- The Login server

You activate these audits by initializing the Security Audit facility in your PRIMOS.COMI file.

If you choose to turn off the Security Audit facility, do not do so at the same time each day.

Idling Security Audits: You can deactivate the Security Audit facility without revealing this by starting the facility but disabling all the events. When the facility is in this state, users may see the phantom AUDITOR but be unaware that it is inactive. To set security audits at an idle, use the SECURITY_MONITOR command in the following form:

```
SECURITY_MONITOR -EVENTS ALL -DISABLE
```

Switching Audit Files From Filled Media: When the audit file is on a disk or tape that fills up, a warning message is displayed at the supervisor terminal. Eventually, the system buffers used for auditing fill up. Any user being audited cannot proceed any further, because the records cannot be written to the disk or tape. If all users are being audited, then only the supervisor terminal remains active, since it has a special buffer reserved for such media overflow situations.

When a disk or tape fills up, you must switch the audit file to a disk or tape that has available space. At that time, the Audit Collection facility records the contents of the system buffers and thus frees them for new audits. The users then become active again. No events are lost.

Be careful to generate as little activity as possible before switching the audit file.

Caution

If you use extra commands before switching the audit file, you risk overflowing the supervisor terminal's special buffer. Use as few commands as possible in deciding the location of the new audit file.

You are not pressed for time to complete the audit file switch. The critical factor is the number of commands issued at the supervisor terminal. The available number of commands may vary, because different commands generate varying amounts of activity to fill the special buffer.

WARNING

The system crashes if the special buffer on the supervisor terminal becomes filled. This shutdown guarantees that no audit data is lost. The System Administrator then performs the system recovery, following procedures for the Crash Audit Recovery facility (see Chapter 11).

To avoid a system crash resulting from filled media, you as System Administrator should establish in advance a procedure for switching audit files from filled media, based on the availability of tape drives or disks at your site. If you cannot guarantee online disk space when media fills, train the operator to mount a tape on Magnetic Tape Drive 0 and issue the following commands at the supervisor terminal:

```
OK, ATTACH DEVICE*
OK, SAC MTO SYSTEM:U $REST:NONE
OK, SECURITY_MONITOR -MTO
```

After the system buffers are written to tape, users become active again. Ensuing audits also are written to tape. The SA meanwhile may establish additional disk space, create the proper ACLs for security audits, and then switch audits back to a disk file. (See Chapter 11 for details on the SECURITY_MONITOR command.)

Typical Security Audits: You can gather an adequate general audit by issuing the SECURITY_MONITOR command twice, as follows:

```
OK, SECURITY_MONITOR -EV FS SYS -EVTYPE FAIL NOACC -ENABLE
    .
    .
    .

OK, SECURITY_MONITOR -EV PRIV -ENABLE
```

You thereby enable auditing of actions that result in failure and no-access for both file system and system events. You also enable the auditing of all privileged operations. (See Chapter 11 for a description of the SECURITY_MONITOR command.)

Here are some other guidelines for your security audits:

- Vary the objects of your audits.
- Do not audit all events, unless necessary.
- Unless you suspect a breach in ACLs security, do not audit attaches.
- If you suspect a security problem
    - Audit those events and user IDs most likely to provide clues.
    - If your suspicions are not specific, audit all events for irregular time intervals.

Contents of Security Audits

While only the System Administrator needs to know fully how to use the Security Audit facility, all users should know that every action on the system may be subjected to a security audit.

Each security audit trail provides the user ID being audited, a time stamp, the event group and event type involved with the particular audit, and a code to indicate the status of the audited action. See Appendix C for the order of record fields within a security audit.

8

ADDING SUBSYSTEMS

Each revision of PRIMOS includes a Spooler and a Batch subsystem at no extra charge. You may wish to add other subsystems to your system. Prime Computer offers separately priced subsystems to provide you with a complete system solution.

The Spooler subsystem and the Batch subsystem were revised at Rev. 21.0. Further Spooler modifications are new at Rev. 22.0. This chapter summarizes significant changes for each subsystem. The detailed set-up guidelines are in the following documents:

Spooler	*Operator's Guide to the Spooler Subsystem*
Batch	*Operator's Guide to the Batch Subsystem*

Some of the other Prime subsystems, and the documents that describe them, include

DBMS	*DBMS Administrator's Guide* (DOC6292-2LA) and *ROAM Administrator's Guide* (DOC7345-3LA and UPD7345-31A)
DPTX	*Distributed Processing Terminal Executive Guide* (DOC4035-4LA)
FTS	*PRIMENET Planning and Configuration Guide* and *Operator's Guide to Prime Networks*
Prime INFORMATION	*Prime INFORMATION Administrator's Guide* (DOC10065-2LA)
OAS	*OAS Administrator's Guide* (DOC10017-4LA)
PRIMENET	*PRIMENET Planning and Configuration Guide*
PRIME/SNA	*PRIME/SNA Administrator's Guide* (DOC8908-3LA)
PRIMIX	*Using PRIMIX on the Prime 50 Series* (DOC9709-2LA)
RJE	*Remote Job Entry Phase II Guide* (DOC6053-4LA and UPD6053-41A)

THE SPOOLER SUBSYSTEM

The Spooler subsystem changed substantially at Rev. 21.0 and has incurred some additional changes at Rev. 22.0. Changes for both revisions are summarized below. See the *Operator's Guide to the Spooler Subsystem* for

- A description of the structure of the Spooler subsystem
- Directions for setting up the spooler
- Directions for daily operator maintenance of the spooler
- Directions for using the spooler

Since changes made at Rev. 22.0 are built on the changes made at Rev. 21.0, both sets of changes are summarized below.

Changes to Spooler at Revision 21.0

Directories: Beginning with Rev. 21.0, the Spooler subsystem uses three directories: SPOOL*, SPOOL_QUEUE*, and SPOOL_DATA*. (SPOOLQ still exists, for the sake of Prime OAS, which directly references the directory.)

- SPOOL_DATA* holds the files to be printed. It is a directory separate from the spool queue itself. There can be several SPOOL_DATA* directories.
- SPOOL_QUEUE* contains the list of print requests awaiting printing. It may also contain two optional files:
  - FULL_LIST_USERS
  - DATA_PARTITIONS
- SPOOL* contains all other files and subdirectories.

Printer Environments: Beginning with Rev. 21.0, printer environments are defined by creating ASCII files with a standard text editor. The PROP command is no longer used to create and modify environments. A new utility, CONVERT_ENV, produces environment files from pre-Rev. 21.0 environment files.

Printer Environment Attributes: The SPOOL command uses a new -ATTRIBUTE option to specify form type and destination site, plus other print requirements. The -FORM and -AT options are thereby made part of the -ATTRIBUTE option, although they both still function at Rev. 21.0. The System Administrator controls access to printers (by individuals or user groups) by means of optional attribute lists.

Access, Security, and Privileged Users: Beginning at Rev. 21.0, all access is controlled by setting ACLs on the relevant files and directories, and through two special ACL groups: .SPOOL$$ and .SPOOL_ADMINISTRATOR$. The new Spooler subsystem maintains the requirements for C2 level of security.

Spool Queues: Only one spool queue can exist on each node. There is no limit to the number of entries in the queue. The only size limit is the partition size. Spool queues cannot be accessed by despooler phantoms of pre-Rev.21.0 versions of the spooler. However, the Rev. 22.0 despooler phantom can access spool queues of earlier revisions of the Spooler subsystem.

Output Format: A new, easier-to-use EVFU file format is provided. The pre-Rev. 21.0 type of EVFU file continues to function to allow use of existing EVFU files.

An additional line of information can be included at the top right of the header page.

A rotating banner allows you to cycle the character used to build large letters and make lines on the header pages. The rotating banner equalizes wear on printer characters. It also provides a perforation aid for bursting pages.

Accounting Routine: A new interface allows users to supply their own accounting routines.

Changes to PROP: The PROP command no longer functions to create or modify environments. Environments are now created by editing ASCII files with a standard text editor. A new option PROP -VERIFY allows environment files to be checked before they are put into service.

A new -RESET option at Rev. 21.0 allows one environment to be stopped, and another to be started in its place, in a single operation.

The IDLE, FINISH, and NOW arguments are changed to -IDLE, -FINISH, and -NOW for consistency across the range of Prime products.

The command PROP -STATUS produces a substantially changed display.

Changes to SPOOL: At Rev. 21.0, a SPOOL -LIST command produces a new display format. SPOOL -LIST -ALL is used to view all the spool queues rather than SPOOL -LIST * as in pre-Rev. 21.0 revisions.

A new -SFI option suppresses file information from the header and the trailer of a listing. The information that is suppressed includes the file pathname and the date and time the file is modified. Those users with rights to this information may use SPOOL -LIST -DETAIL to obtain it while the file is still in the despooler queue.

Request numbers are no longer prefaced by PRT.

A new -ATTRIBUTE option allows you to specify more printing requirements. -ATTRIBUTE supersedes -FORM and -AT, although -FORM and -AT are still supported.

A new -NOCOPY option prevents a copy of the file from being created in the SPOOL_DATA* directory. The option is used with exceptionally long files or for security reasons.

Performance: At Rev. 21.0, a new queue polling algorithm, and several detail changes, result in significantly improved performance.

Changes to Spooler at Revision 22.0

Changes at this revision affect the PROP command and the SPOOL command.

Changes to PROP: At Rev. 22.0 the PROP command has two new options plus an alteration to an old option.

The two new options, PROP -SUSPEND and PROP -RELEASE, enable a user to suspend a print job that is currently being printed (to allow for adjustments), and then to reactivate the print job.

The altered PROP -BACK *n* option enables backing up a number of pages (to reprint material lost because of printer problems). You are no longer limited on the number of pages you may back up.

Changes to SPOOL: At Rev. 22.0 the SPOOL command has eight new options:

- The -SET_FONT option to SPOOL allows a choice of font typeface.
- The -SET_LANDSCAPE and -SET_PORTRAIT options allow the selection of paper orientation.
- The -SET_PAPER_BIN option allows for a choice of paper source.
- The -SPOOL_WHILE_OPEN option enables the printing of a file while it is still open for writing to disk.
- The -XLATE option enables a user to define and select a personal character set map as an alternative to the printer's default.
- The -FROM and -TO options may be used either separately or together to enable the printing of parts of a document.

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.

Since Rev. 21.0, BATCHQ must be an ACL directory. The ACL group .BATCH_ADMIN$ is reserved for administrators and operators of the subsystem. The System Administrator must use EDIT_PROFILE to assign members to this group *before* initializing the subsystem.

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.
- Use EDIT_PROFILE to
  - Assign the .BATCH_ADMIN$ ACL group to administrators and operators.
  - Assign at least two command levels to each Batch user.
- 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.
- Set up enough phantoms for Batch to run (by specifying a sufficient number for the NPUSR directive in the system configuration file).
- Initialize the Batch subsystem (with the INIT program in the BATCHQ directory) after Rev. 21.0 software is installed.
- Verify that BATCHQ is on a suitable partition and has ACL directories (done at installation).
- Add the BATCH -START command to the PRIMOS.COMI file to bring up the Batch subsystem at cold start.
- 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 database.

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 and Batch Versions

Rev. 22.0 PRIMOS will run three previous versions of Batch (Rev. 20.0, Rev. 20.2, Rev. 21.0) until Rev. 22.0 Batch is installed. However, you cannot revert to a pre-Rev. 21.0 version of Batch after you have installed either its Rev. 21.0 or Rev. 22.0 version. These two versions of Batch will not run on a pre-Rev. 21.0 version of PRIMOS.

If you are converting from a pre-Rev. 21.0 Batch, you must initialize the Batch database using the INIT program in the BATCHQ directory.

Rev. 20 versions of Batch do not work correctly with a pre-Rev. 20 PRIMOS BATDEF file (which contains queue definition information) and queue files.

Note

When a Rev. 20.0 or greater version of Batch 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. 22.0.

Table 8-1 summarizes the above information.

TABLE 8-1. Converting to Rev. 22.0 Batch

From Rev	*Run INIT?*	*Need New BATDEF?*	*Possible to Revert?*
19.4	Y	Y	N
20.0	Y	N	N
20.2	Y	N	N
21.0	N	N	Y

Phantoms: The Batch subsystem requires one phantom (which runs under the name BATCH_SERVICE) to control the Batch monitor exclusively. Another phantom is required for each executing Batch queue.

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 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
- Timeslice

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.

Note

The Batch Administrator must either make the 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 number of queues to be defined
- The number of phantoms to be allocated to run Batch
- The timeslice 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 queue ordering 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 postponed with the -HOLD option to 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 to 16 queues to allow 10 to 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.

Timeslices and Scheduler Priorities: Every process on the system, including Batch phantoms, has a timeslice and a scheduler priority. By default, user phantoms run at the same priority level and with the same timeslice as the user.

You can set up a Batch subsystem with different queues running Batch jobs at different priority levels and with different timeslices. 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 timeslice to control the length of time Batch jobs are allowed to run before being rescheduled.

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

- Queues for short jobs should have a limited CPU time, a relatively high priority, and a short or normal timeslice. 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 maximum CPU time without a default. A job that is not submitted with the -CPTIME option cannot use such a queue.
- Queues for average jobs should have default timeslices and priorities.

- Queues for large, slow jobs should have no CPU time limit, no elapsed time limit, a large timeslice, 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.)

Note

Terminal users have priority level 1.

To take full advantage of the scheduler, distribute processes on your system evenly across each of the priority levels you intend to use, but do not penalize terminal users. 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 timeslice levels of processes other than the Batch monitor and Batch jobs.

Search Order of Batch Queues: The Batch monitor handles tasks submitted to it through the JOB command. 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. If the JOB command does not include a specific queue request (the -QUEUE option), then the Batch monitor searches the queues in the order in which they were added to the system. The monitor uses the first queue that can handle the estimated CPU and elapsed times (-CPTIME and -ETIME) of this job. If the job submittal did not specify these times, then the Batch monitor uses the first queue that does not specify time requirements.

To find out the search order of your Batch subsystem, use the BATGEN -STATUS 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 for a pre-Rev. 21.0 version of Batch. The INIT program uses this information to set up the access to the Batch database. Batch Administrators receive ALL access to the directory BATCHQ and to all its subdirectories 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 Rev. 21.0 version of Batch uses the ACL group .BATCH_ADMIN$ for those you want to designate as Batch Administrators, and the INIT program no longer prompts you to designate them.

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 database. 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 four ongoing tasks:

- Ensuring that the proper users are assigned to .BATCH_ADMIN$ with EDIT_PROFILE
- Ensuring that BATCH_SERVICE does not have a login type user ID within EDIT_PROFILE
- 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 timeslice or the scheduler priority of Batch jobs is changed)

Controlling the Batch Subsystem

After the Batch subsystem is set up and started, the Batch Administrator (plus those others included in .BATCH_ADMIN$) 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 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.*

9

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 service users are

- 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, the SA must assign the user a set of system attributes and one or more sets of project attributes.

The system attributes must include a user ID, a password (possibly null), and the user's password lifetime (possibly defaulting to the system's default password lifetime). 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 database) 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 6, to define a user's system attributes and project attributes.

Origin Directories

The SA must twice define a user's origin directory (also called an Initial Attach Point or IAP). It is not enough to define the user's IAP within 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

The most common user problems involve unsuccesful 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.`

- Verify that the user correctly typed the user ID and password.
- 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, the user 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 database 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 that cannot be accessed now.
- The user's Initial Attach Point is unreachable due to disk problems.

Check the user's project database in the SAD with EDIT_PROFILE to identify 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 because 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 not running 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. If the search rules for the Login server are not installed on the command device, the server will not start at cold start. Verify that the search rules are installed in SERVERS*>LOGIN_SERVER.ENTRY$.SR and that the file contains no typographical errors. From the supervisor terminal, fix the search rules and start the Login server with the START_LSR command. If problems persist, check SERVERS*>LOGIN_SERVER.ENTRY$.SR for typographical errors.

If the Login server does not begin at cold start, verify that SERVERS* is on the boot device.

Access Problems

Users may come to you because their programs are failing due to access problems. Access problems are signaled 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 handling different from 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 5, Setting Access Rights.

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 solve the problem indicated by the error message.

`Bad password.` *dir-name df_unit_*
The user attempted to attach to a password-protected directory, named *dir-name*, with an incorrect or missing password. To solve this problem, you have three choices:

- Give the user the password.
- Remove the password, let the user complete the task, and then replace the password.
- Remove the password and place ACLs on 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 insufficient rights. (Some commands do not print the object's name.) *cmd-name* is the PRIMOS command (such as ATTACH) or module (such as OPEN$A or std$cp) that returned the error. Give the user appropriate access rights to accomplish the task.

`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 that returned the error (for example, LIST_ACCESS or LIST_QUOTA). Give the user appropriate access rights to accomplish the task.

`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 that returned the error (for example, ATTACH or LD). Some of the causes and solutions to this error are

- 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 *Operator's Guide to Prime Networks* or the *PRIMENET Planning and Configuration 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 to the next 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 Problems With Crowded Disks later in this chapter.

Quota Problems

On a system where directory quotas are in use, users with quota problems may require your intervention.

The most common case is that 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 exceeded was not for the user's own directory, but for a top-level directory. Figures 9-1 and 9-2 illustrate how the violation may occur one or more levels away from the established quota level.

In Figure 9-1, no individual subdirectory has exceeded its quota, but no subdirectory can add records because the sum of records within the top-level directory and its subdirectories equals the quota set on the top-level directory.

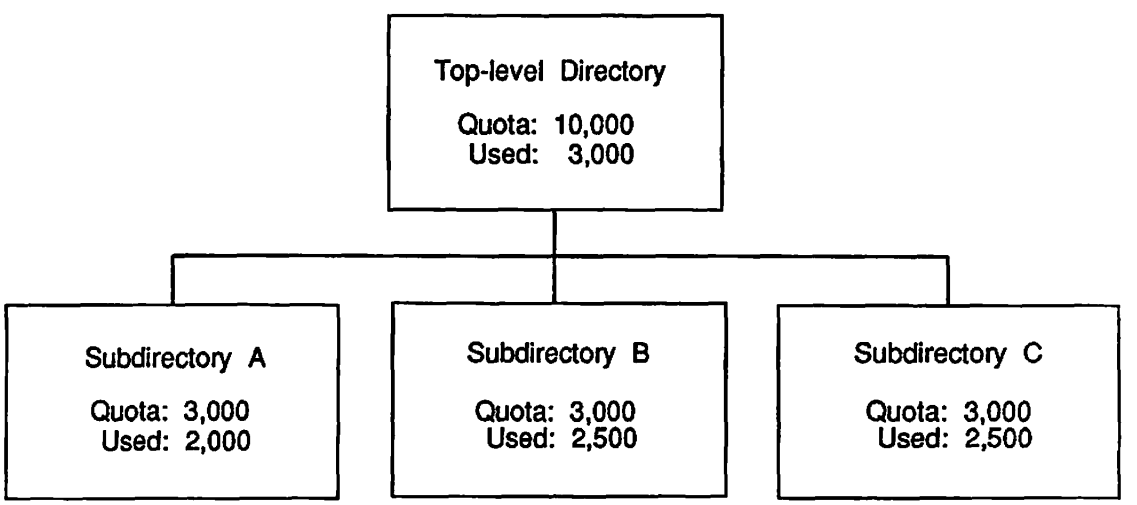

FIGURE 9-1. Quota Error: Sum of Two Levels Fills Quota

In Figure 9-2, the blockage for Subdirectory B occurs even higher in the tree. Users of Subdirectory B must search up two levels to find it.

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.

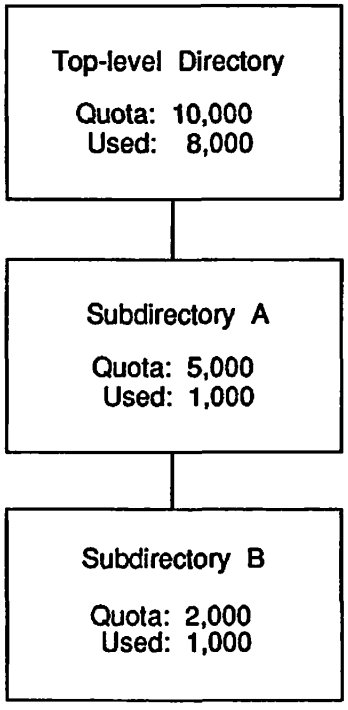

FIGURE 9-2. Quota Error: Sum of Three Levels Fills Quota

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 top-level 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 later section, 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.

EPF-level Problems

Users with EPF-level problems should release any unnecessary 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 based 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 have 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 directory 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 10, 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 recover enough space using the last five 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, increase the quota limit for the directories that most need extra space. (See the discussion of quota strategies in the chapter on Disks and Tape Drives in the *System Administrator's Guide, Volume I: System Configuration.*)
- 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 unnecessary 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.)

10

SYSTEM MONITORING

The System Administrator must always be aware of whether the system is running normally or malfunctioning. This chapter discusses four methods to keep track of system events:

- The system logbook
- Event loggers for Rev. 22.0 and earlier
- System-monitoring commands for Rev. 22.0 and earlier
- System Information and Metering (SIM) 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 review 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. There is no set definition for the format of the logbook or for what type of information should go into it. Rather, it is up to you, as the System Administrator, to determine the makeup of the logbook. (See the following sections for 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 more efficiently.

Format of the Logbook

To help you determine the format of the system logbook, here are some suggested standards and procedures that have been used successfully by operators of Prime systems.

- 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: Relevant hardware information includes

- 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: Relevant environmental information includes

- 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: Relevant software information includes

- 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 search rules file (SEARCH_RULES*>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 version of PRIMOS used by the system.
- A listing of the network configuration as produced by CONFIG_NET.
- A listing of the DSM configuration file (CF) as produced by CONFIG_DSM.
- 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: Relevant operations information includes

- 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 names of the partitions created and the disk drive 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 CMDNC0 or libraries in LIB or LIBRARIES*, with the date, time, and reason for the action.
- All changes to the default entrypoint search rules file, SEARCH_RULES*>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 are added to or deleted from the system.
- All passwords that are changed or revealed to users.
- All telephone requests for passwords or for telephone numbers.

Information on Halts: Relevant information on halts includes

- 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 and DSWPARITY2).
- 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 2, 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 half an 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

Rev. 22.0 PRIMOS has 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 is recorded in two event log files, one for system events and one for network events. These files can be useful in tracking problems, especially those problems that develop or worsen over time.

Event Logging at Rev. 22.0

Since Rev. 21.0, a new PRIMOS utility, Distributed Systems Management (DSM), handles event logging plus many other system management services. Among other things, the command START_DSM activates the logging of both system and network events. The command is valid only from the supervisor terminal. The command is in the PRIMOS.COMI.TEMPLATE file that you customize to make your system's PRIMOS.COMI file.

Do not stop DSM on a system that is to remain active. (The DSM Network Administrator, however, may need to stop DSM in order to activate a new DSM configuration file or to run FIX_DISK on the directory DSM*.) Event logging automatically proceeds using certain initial settings, without any adjustments to DSM or to event log files. There is a single log file for system events and a single log file for network events. As one initial setting, DSM appends messages indefinitely to these files. The files are located in subdirectories to DSM*>LOGS. Access rights to DSM*>LOGS should be

```
SYSTEM:UR
SYSTEM_MANAGER:UR
DSM_LOGGER:ALL
```

The one event logging command that you as System Administrator will use with some regularity is the DISPLAY_LOG command. Use the DISPLAY_LOG command to display log file information at the terminal or to write this information to a disk file for printing.

Note

You may run DISPLAY_LOG when DSM is inactive.

If changes to the initially set event logging attributes are deemed necessary, use a second logging command, the ADMIN_LOG command, to change the way the logging mechanism handles log file information.

You may find a third command useful for modifying general event logging. The CONFIG_UM command enables the logging of other products in the event log files. You may also use it to redirect the logging of products or to create private logs of certain events of particular interest.

The use of the above three logging commands is described in the next three paragraphs. Refer to the *DSM User's Guide* for complete details on the use of these commands. The

guide provides a full description of the private, system, and network event logging features of DSM, as well as the many other features of DSM.

Using DISPLAY_LOG: You can use this command to read the event log information that DSM has automatically recorded in two files.

The system event log file is DSM*>LOGS>PRIMOS>PRIMOS.LOG. The network event log file is DSM*>LOGS>NETWORKS>NETWORK.LOG. As system logs, they must have a pathname beginning with DSM*>LOGS. The DISPLAY_LOG command defaults to system log files, so you need not provide the -SLOG option with this command. For example, display the contents of the system event log file by issuing this command at the supervisor terminal:

```
OK, DISPLAY_LOG DSM*>LOGS>PRIMOS>PRIMOS.LOG
[DISPLAY_LOG Rev. 21.0.0 Copyright (c) 1987, Prime Computer, Inc.]
*** Message from product LOG_COLD, generated by SYSTEM on SYSNAM
    (Severity Information, occurred at 07 Jun 88 08:20:16 Tuesday)
COLD START PRIMOS REV 22.0.0 CPU TYPE = P9655
MICROCODE REV = 7
PROCESSOR ID = xxxx

*** Message from product LOG_MISC, generated by SYSTEM on SYSNAM
    (Severity Information, occurred at 07 Jun 88 08:20:16 Tuesday)
DISK MOUNT: LOGTST  ON 003060  (OCT)

*** Message from product LOG_MISC, generated by SYSTEM on SYSNAM
    (Severity Information, occurred at 07 Jun 88 08:20:16 Tuesday)
DISK MOUNT: LOGUSR ON 061060  (OCT)

*** Message from product LOG_MISC, generated by SYSTEM on SYSNAM
    (Severity Information, occurred at 07 Jun 88 08:20:16 Tuesday)
DISK MOUNT: PAGER  ON 100461  (OCT)

*** Message from product LOG_MISC, generated by SYSTEM on SYSNAM
    (Severity Information, occurred at 15 Jun 88 13:59:20 Wednesday)
PRIORITY ACL set on disk LOGTST  by user 1.
     .
     .
     .
32 messages retrieved from log
OK,
```

You may designate a file to hold this output by providing the desired pathname immediately after the pathname of the event log file.

Using ADMIN_LOG: You use the ADMIN_LOG command to create, purge, or delete a log. The command is also used to list and modify log attributes.

Under normal operations, you need not use this command on the event log files because DSM automatically creates these files and automatically overwrites their oldest material. The files are cyclic by default. Their default size is 10 records (20 kilobytes). Once the file is full, the oldest event log is overwritten by the newest. Unless a system has a serious problem, its event log files will hold more than a month's accumulation of event logs. Nevertheless, you may change any of the defaults except the cyclical/linear format, which is fixed at the creation of the file.

For example, suppose you are a very cautious System Administrator who wants to change the system event log's defaults for maximum size, warning level, and retention time. You want to limit the retention of events to 40 days and are willing to increase the size of the log to 20 records in order to ensure the file is large enough for this time period. You also want to be warned in case the file threatens to have its oldest events overwritten before the end of the forty days. (The default warning level for a log file is undefined.) You want the system to generate a warning when the system event log is filled to 90% capacity. This setting may become a nuisance when that threshold is reached because every new entry thereafter will generate a warning. However, you expect to reach that level only when problems exist.

At the supervisor terminal, issue the ADMIN_LOG command first with the -LIST option and then with the -MODIFY option.

Note

You may issue these DSM commands at another terminal by adjusting the Configuration File for DSM. To do this, use the CONFIG_DSM command. See the *DSM User's Guide* for details.

As a quick check of default attributes, first list the attributes of PRIMOS.LOG. Then modify the maximum size, the warning level, and the retention time attributes. Finally, list the attributes again to verify the new settings:

```
OK, ADMIN_LOG DSM*>LOGS>PRIMOS>PRIMOS.LOG -LIST
[ADMIN_LOG Rev. 21.0.0 Copyright (c) 1987, Prime Computer, Inc.]
Node: SYSNAM
       Listing of DSM*>LOGS>PRIMOS>PRIMOS.LOG
       Cyclic: Yes
       Maximum size: 10 Records
       Minimum size: 1 Record
       Warning level: Undefined
       Current size: 4 Records
       Retention time: Infinite
       Age of oldest message: 10 Days
       Purge time: 01:00
OK, ADMIN_LOG DSM*>LOGS>PRIMOS>PRIMOS.LOG -MODIFY -MXSZ 20 -WL 90 -RET 40
[ADMIN_LOG Rev. 21.0.0 Copyright (c) 1987, Prime Computer, Inc.]
Node: SYSNAM
Log has been modified.
OK, ADMIN_LOG DSM*>LOGS>PRIMOS>PRIMOS.LOG -LIST
[ADMIN_LOG Rev. 21.0.0 Copyright (c) 1987, Prime Computer, Inc.]
Node: SYSNAM
       Listing of DSM*>LOGS>PRIMOS>PRIMOS.LOG
       Cyclic: Yes
       Maximum size: 20 Records
       Minimum size: 1 Record
       Warning level: 90 %
       Current size: 4 Records
       Retention time: 40 Days
       Age of oldest message: 10 Days
       Purge time: 01:00
OK,
```

Using CONFIG_UM: You can use this command to record other events in the system and network event logs, to adjust the severity of events being logged, or to redirect the log to a different or additional destination. See the *DSM User's Guide* for complete details on the use of this command and on the products available for customized logging.

For example, suppose you have a system with a LAN300 network installed. You are aware that LAN300 has its own set of private event logs located in the directory NETWORK_MGT*. However, you decide to make a second record of any event with the severity of FAILURE affecting the performance of the LAN300. You want to record these events in the network event log located under DSM*>LOGS.

Use the CONFIG_UM command to define a customized log selection called LAN_EVS:

```
OK, CONFIG_UM LAN_EVS -SELECT
[CONFIG_UM Rev. 21.0.0 Copyright (c) 1987, Prime Computer, Inc.]
Product Name: NMSR
Product Name: CONTROLLER_DLL
Product Name: <CR>
Severity: FAILURE
Severity: <CR>
Destination: LOGGER DSM*>LOGS>NETWORKS>NETWORK.LOG
Destination: <CR>
Do you wish to edit this selection ? YES
Selection Name: LAN_EVS
Product Name: NMSR
Product Name: CONTROLLER_DLL
Product Name:
Severity: FAILURE
Severity:
Destination: LOGGER DSM*>LOGS>NETWORKS>NETWORK.LOG
Destination:
Do you wish to edit this selection ? NO
Configuring LAN_EVS on SYSNAM.
Completed OK
OK,
```

Transition From Pre-Rev. 21.0 Event Logging

The installation of Rev. 22.0 PRIMOS automatically activates the subsystem for event logging. If you are upgrading your system from a revision of PRIMOS earlier than Rev. 21.0, first remove the directives LOGREC and NETREC from your system configuration file before running Rev. 22.0. If PRIMOS encounters them in the file, it ignores the directives and issues error messages.

At Rev. 21.0 the PRIMOS command EVENT_LOG became obsolete. PRIMOS issues an error message if the command is used for either system or network logging.

Access to Pre-Rev. 21.0 Event Logs

The event logs previously stored in LOGREC* and PRIMENET* are still accessible, but no new event logs will be stored there. If you have not deleted the pre-Rev. 21.0 commands PRINT_SYSLOG and PRINT_NETLOG, you can use them to recover useful information from these directories. If you decide the remaining information is no longer useful, you may wish to delete it. Be careful to delete only outdated logs from LOGREC* and PRIMENET*.

SYSTEM-MONITORING COMMANDS

Use the following PRIMOS commands to monitor your system:

- STATUS monitors high-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.
- PRIMAN provides a separately priced resource usage monitoring and analysis package.
- MONITOR_NET monitors the events on the network.
- FIND_RING_BREAK locates breaks in the ring.

System Information and Metering Commands

At Rev. 21.0, PRIMOS provides 15 additional System Information and Metering (SIM) commands through the DSM facility. While these commands may address the local system alone, the commands are especially tailored to address a multiple-node system configured through Distributed Systems Management. All SIM commands accept options, such as -FREQ and -TIMES, that enable the collection of periodic system information samples for metering tasks.

The 15 SIM commands have the following functionality.

Command	*Meaning*
LIST_ASSIGNED_DEVICES	Lists assigned devices
LIST_ASYNC	Lists state and configuration of asynchronous terminals
LIST_COMM_CONTROLLERS	Lists communications controller configuration
LIST_CONFIG	Lists PRIMOS configuration directives
LIST_DISKS	Lists logical disks (partitions)
LIST_LAN_NODES	Lists nodes on LAN300 local networks
LIST_MEMORY	Lists physical memory usage
LIST_PRIMENET_LINKS	Lists PRIMENET status
LIST_PRIMENET_NODES	Lists nodes configured under PRIMENET
LIST_PRIMENET_PORTS	Lists assigned PRIMENET ports
LIST_PROCESS	Lists system processes
LIST_SEMAPHORES	Lists semaphores in use
LIST_SYNC	Lists synchronous line configuration
LIST_UNITS	Lists any user's open file units
LIST_VCS	Lists active virtual circuits

Refer to the *Operator's Guide to System Monitoring* for a brief description of each of the SIM commands. See the *DSM User's Guide* for a detailed description of each command.

The other system-monitoring commands are discussed briefly in the sections below.

You can also use the SIZE and LD commands for monitoring. 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.

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, print it by using the SPOOL command.

The STATUS Command

The STATUS command monitors high-level system events. When you invoke STATUS without an argument at the supervisor terminal, the following information is displayed:

- The version of PRIMOS your system is running
- The size of main memory
- Your user name (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, as well as their mirrored partition status
- Semaphore values
- All configured network nodes and their status (UP or DOWN)
- The physical device numbers of the command device (COMDEV), and a maximum of eight paging devices.

Note

At Rev. 21.0, the configuration directives PAGDEV and ALTDEV were replaced by the single configuration directive PAGING accompanied by a maximum of eight pdev numbers. Also, the configuration directive PRATIO was replaced by the PRIMOS command PRATIO. Use the new PRATIO command to establish the proportion of use for these paging partitions.

- All logged-in users, including their user IDs, user numbers, terminal line numbers, in-use partitions, and assigned devices

Use the STATUS COMM format of the command to display information on communication 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 partitions are currently started
- 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 internal events related to the system hardware. 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 (L) access to the target and parent directories, and Use (U) access to any higher level directories. However, you can override this restriction by using a priority ACL. Priority ACLs are discussed in Chapter 5, Setting Access Rights.

Format

LIST_QUOTA [*pathname*] [**-BRIEF**]
LQ

pathname is the directory for which you want quota information. If you do not specify a pathname, information on the current directory is listed.

Example

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 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 as -BR) displays the quota data in tabular form. No message is displayed if the directory is a nonquota directory; instead, the maximum number of records is listed as zero.

The LIST_QUOTA command is also explained in the *PRIMOS Commands Reference Guide* and in the *PRIMOS 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

Format

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.

Discussion

The default output gives the number in 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 (R) rights to the DSKRAT file on each disk, and List (L) and Use (U) 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 (L) and Read (R) 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, information on space usage is also displayed. 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% 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.

The PRIMAN Utility

PRIMAN is a separately priced utility that monitors and analyzes resource usage. The PRIMON command is used to monitor and gather the data. The PRIMAN command is used to analyse the information and prepare reports. For more information, see the *PRIMAN User's Guide.*

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 Planning and Configuration Guide.*

11

SECURITY AUDITS

Prime Computer offers a separately priced Security Audit facility that enables the System Administrator to create audit trails and verify the security of the system. These audit trails provide a record of activity on the system.

The PRIMOS operating system itself provides the levels of security by means of priority ACLs, ACLs, and device ACLs. These access controls allow an easy means of providing or protecting files without any breach in access security. See Chapter 5, Setting Access Rights, for details.

The Security Audit facility provides extra reporting features to the System Administrator. The System Administrator may create an audit trail for

- Certain users
- Certain attempted activities performed by anyone on the system
- The result of attempted activities
- All of the above at once

For example, the System Administrator can audit every occurrence of a certain action (event) on the system, such as an attach. The System Administrator can also tune the facility to audit only a particular result (event type) of that action, such as each failure to perform an attach. The System Administrator might also choose to audit all the actions of a particular group of users. The SA can create audits for many combinations of events, event types, and users.

Each audit trail indicates a user ID, the pathname to the file or program being audited, a time stamp, the type of operation, its result (success or error code), and so on. See Appendix C, Record Fields for Security Audits, for all the record fields in an audit trail.

The Security Audit facility consists of

- An Audit Collection facility using
  - The SECURITY_MONITOR command
- An Audit Reporting facility using
  - The SECURITY_STATUS command
  - The PRINT_SECURITY_LOG command
- An Audit File Backup facility using
  - The TRANSFER_LOG utility
- A Crash Audit Recovery facility using
  - A RING0.MAP file, maintained in LOAD_MAPS*
  - A tape dump procedure after each system halt
  - The CRASH_AUDIT utility to process the tape dump after the ensuing cold start

AUDIT COLLECTION FACILITY

The SECURITY_MONITOR command runs the Audit Collection facility. Different command options allow the System Administrator to

- Start and stop the facility (see -START and -STOP options)
- Turn audits of certain users on and off (see -ON and -OFF options)
- Enable and disable audits of event groups and event types (see -ENABLE and -DISABLE options)
- Tune the Audit Collection facility to record only those events that conclude a given way, such as only attaches that fail (see -EVENTS and -EVENT_TYPES options)
- Manage the audit file (see -OUTFILE and -MT options)

Actions and Results

The Audit Collection facility audits certain actions and the results of those actions. Although an audit identifies the user who initiated the actions, the main objects of the audit are the actions and the results, which are recorded by the -EVENTS and -EVENT_TYPES options of the SECURITY_MONITOR command.

Actions are recorded by the -EVENTS option. All audited actions may be categorized as one of four possible events: FILE_SYSTEM, SYSTEM, PRIV_OPS, ATTACHES. Table 11-1 lists all the actions that may be audited for each event.

FILE_SYSTEM events are actions involving the creation of a file, the deletion of a file, the opening of a file, and the creation or modification of ACLs on a file.

SYSTEM events are actions involving normal logins and logouts, phantom logins and logouts, assignment of devices, allocation of segments, the return of segments to the system, and program mapping and unmapping.

PRIV_OPS events are actions that require special privileges, such as the use of commands restricted to the supervisor terminal or to privileged users, especially the System Administrator.

ATTACHES events are the different types of attach actions that may be performed.

Results are recorded by the -EVENT_TYPES option. The result of events is either success or failure, so one might expect to find only two event types. Indeed, the -EVENT_TYPES option has arguments for SUCCESS and FAILURE. However, -EVENT_TYPES also reports a special type of failure: NO_ACCESS. Furthermore, if you want the audited event to be recorded no matter what the result, you may either specify ALL as an -EVENT_TYPE argument or suppress use of this argument.

The System Administrator may create a large variety of security audits by changing the options mix for -EVENTS, -EVENT_TYPES, and -USERS when using the SECURITY_MONITOR command.

The next section gives a detailed description of the SECURITY_MONITOR command. That section is followed by examples showing the use of the SECURITY_MONITOR command for security audits.

TABLE 11-1. Actions Audited for -EVENTS

-EVENTS FILE_SYSTEM	*-EVENTS SYSTEM*
Allocate a record for an ldev.	Allocate segment.
Calculate access rights to a file.	Assign, unassign device.
Change a name.	Assign, unassign asynchronous line.
Change open mode of file.	Calculate device access rights.
Check existence of a file.	Change a password.
Check for quota overflow.	Change or delete a batch job.
Close a file by pathname.	Copy one segment to another.
Close a file by unit number.	Initialize a VMFA segment or map an EPF.
Close a file by entryname.	Initialize a user.
Create a directory.	Log out.
Delete a ROAM file.	Normal login.
Delete a file.	Phantom login.
Delete a segment directory.	Restore static mode program.
Delete access category.	Return access rights to segment.
Force a disk write of a file.	Return segment, terminate EPF.
Manipulate a segment directory.	Set segment access.
Open a file.	Submit, read, change, or delete a spooler job.
Open a segment directory.	
Read directory password.	
Read directory quota data.	
Revert an ACL.	
Set a directory password.	
Set access category on object.	
Set an ACL on an object.	
Set default protection on object.	
Set file attributes.	
Set quota on a directory.	

TABLE 11-1. Actions Audited for -EVENTS (continued)

-EVENTS PRIV_OPS	*-EVENTS ATTACHES*
Change System Administrator.	Attach to a directory.
Check if a user is privileged.	Attach scan.
Check the privilege of a caller and call SEC_PROB.	Attach to the Initial Attach Point.
Configure an asynchronous line.	Attach to MFD specified by ldev.
Enable and disable audits of particular users.	Perform a relative attach.
Enable or disable audits of -EVENTS and -EVENT_TYPES for the SECURITY_MONITOR command.	
Forcibly log out users.	
Get information about open units.	
Perform SECURITY_STATUS command.	
Process EDIT_PROFILE commands.	
Process privileged commands: ADDISK, CHAP, DISKS, LOOK, MAXUSR, OPRPRI, REPLY, SETIME, SETMOD, SHUTDN, STARTUP, USRASR.	
Set or delete a priority ACL.	
Set scheduler variables.	
Share a segment.	
Specify the audit file for SECURITY_MONITOR.	
Specify the number of buffers for SECURITY_MONITOR.	
Start/stop SECURITY_MONITOR.	
Start/stop the Login server.	

The SECURITY_MONITOR Command

The SECURITY_MONITOR command is a privileged command. The System Administrator may use it at any terminal; other authorized operators may run it only from the supervisor terminal.

Format

SECURITY_MONITOR [*options*]
SECMON

Options

-START
Activates the Audit Collection facility and initializes its data structures. If you enter this option by itself, *all* events, event types, and users are audited, and the system uses default values for its other assignments. The -START option is not valid when the monitor is already running.

Note

The options -ON and -OFF are reserved for auditing selected users only. See below.

-STOP
Shuts down the Audit Collection facility and terminates all audits. You are prompted to verify that -STOP is desired. Type YES as a full word to close the output file. A message at the supervisor terminal warns that security audits are now inactive. The -STOP option is not valid when the monitor is not running.

-ON
Activates auditing of users listed with the -USERS option. The default is -ON when the -USERS option is specified. You must include the -USERS option when specifying -ON.

-OFF
Deactivates auditing of users listed with the -USERS option. You must include the -USERS option when specifying -OFF.

-USERS *list*
-US
Specifies a list of users to be selected. If you omit this option when you specify -START, the system defaults to all users. Thereafter you must enter all changes explicitly. The *list* consists of user IDs separated by blank spaces. An omitted list causes an error message: `Must give user list with -USER option.`

-BUFFERS *n*
-BUFF
Sets the number of 4-kilobyte buffers to be used by the audit mechanism. The number *n* must be an integer ranging from 2 through 12. Omission of this option generates a default value of four buffers. Audits during heavy system usage may slow system processing, unless more buffers are allocated. You may change this value at any time by invoking SECMON with a new -BUFFERS *n* option.

-EVENTS [*arguments*]
-EV
Selects the set of event categories to be audited. If you omit this option when you specify -START, all events are affected. The option -EVENTS presumes the additional option -ENABLE as a default. You must specify -DISABLE to turn off specific events.

After startup, for any ensuing SECMON commands, the same events remain monitored until you explicitly specify -EVENTS followed by one or more of the event category arguments. These arguments are

FILE_SYSTEM
FS
States that all file system events are to be affected.

SYSTEM
SYS
States that all system events are to be affected.

PRIV_OPS
PRIV
States that events involved with privileged operations are to be affected.

ATTACHES
ATCH
States that all attach operations are to be affected. To reduce system overhead, the System Administrator may turn off either all attaches or successful attaches. See the examples in the next section.

ALL
Specifies explicitly that all events are to be affected.

-EVENT_TYPES [*arguments*]
-EVTYPE
Specifies the event types to be affected. If you omit this option with -START, all event types are selected. -EVENT_TYPES presumes the additional option -ENABLE as a default. You must specify -DISABLE to turn off specific event types.

After -EVENT_TYPES, you may specify one or more of the event type arguments. These arguments are

SUCCESS
SUCC
States that only successful events are to be affected. The audit selects successful operations for the specified events.

NO_ACCESS
NOACC
States that only access failure events are to be affected. The audit records an operation failure caused by a user's insufficient access rights to the target object.

FAILURE
FAIL
States that only failed events are to be affected. The audit records an operation failure for reasons other than insufficient access rights.

ALL
States explicitly that all event types are to be selected.

-ENABLE
-DISABLE
Specifies that events and event types are activated or deactivated.

This option allows implied combinations of -EVENTS and -EVENT_TYPES options. For example, to audit only problem occurrences for all event groups and all users presently audited, you can disable audits of non-problems:

```
SECURITY_MONITOR -EVENT_TYPES SUCC -DISABLE
```

You might then choose to eliminate the audit overhead for attaches:

```
SECURITY_MONITOR -EVENTS ATCH -DISABLE
```

You still maintain audits on all operations that have problems accessing a file or executing a program.

-OUTFILE ***pathname***
-OUTF
Writes audit data to a file indicated by *pathname*. If you do not specify the -OUTFILE option or you omit a pathname, the facility generates the default file SECURITY_LOG.*yymmdd.hhmmss*. (The suffix *yymmdd.hhmmss* is a time stamp.) The file is opened in the current directory. If you specify a pathname that does not exist, the file is created. If you specify a pathname that already exists, an error is returned.

If you invoke SECMON and specify a new -OUTFILE option while another file is still active, the old file is completed and closed before the new file is activated. No events are lost if you switch files.

-MT ***n***
Writes the security audit log to tape on unit *n*, with a value for *n* ranging from 0 through 7.

-HELP
-H
Displays a summary of the above options.

Examples of Using the SECURITY_MONITOR Command

The following situations show the use of the SECURITY_MONITOR command as part of a Security Audit facility at a hypothetical bank.

Initializing Audit Trails: The System Administrator at a bank starts her system at about 7:00 a.m. each weekday. Few other employees log in before 9:00 a.m. On July 5, 1988, she starts the system. She has the following line in her PRIMOS.COMI file:

```
SECURITY_MONITOR -START
```

The System Administrator expects only a few audit records for -EVENTS, -EVENT_TYPES, and -USERS to be generated before 9 a.m., so she allows the default to a full audit until then. She also knows these records will be filed by default in the directory CMDNC0 under the name SECURITY_LOG.880705.07*xxxx*, since the security monitor was started from the PRIMOS.COMI file that executes in that directory.

Adjusting to Heavy Use: The SA knows that the system will be very busy between 8:50 and 9:10 a.m. as the other employees begin work. To allow them rapid system response at login time, she uses the SECURITY_MONITOR command a second time at 8:45. She issues the single-line command, as follows:

```
SECURITY_MONITOR  -EVENT_TYPES SUCCESS -DISABLE -BUFFERS 10
                  -OUTFILE <BANKIT>AUDITS>SECLOG1.[DATE]
```

Instead of watching the clock to give the command at the correct time, she has embedded this command in a phantom CPL program, which she starts from her own terminal as soon as she logs in to the system. The phantom has her user ID, with all the rights of the System Administrator. The program waits until 8:45 and then issues the above command, which directs the Security Audit facility to ignore all valid operations, such as successful attaches and successful file accesses. While the facility continues to audit the failures and no-accesses of all system users, it also allocates extra buffers to audit heavy use. It opens another disk file to hold the workday audits. This audit file has the pathname <BANKIT>AUDITS>SECLOG1.[DATE].

Before opening the new audit file, the Audit Collection facility closes the default audit file in CMDNC0. It then opens the file SECLOG1.880705.0845*xx* in the directory AUDITS. Now all users are audited without burdening the system with audit collections on their initial attaches or other valid operations.

Security of the Security Audit Facility: The SA is careful that the supervisor terminal, a magnetic tape unit (which she uses for backups and tape dumps), and her own terminal are always protected from both accidental and intentional tampering. Only a priviliged user may therefore start or stop the Security Audit facility itself. Illicit tampering with the facility cannot go unnoticed, since the facility issues messages to the supervisor terminal, reporting changes in status.

For example, when the facility was first started that day with SECURITY_MONITOR -START, the following message was displayed at the supervisor terminal:

```
[SECURITY_MONITOR Rev. 22.0 Copyright (c) Prime Computer, Inc. 1988]

07:02:06 : Switching log file to: SECURITY_LOG.880705.070206
07:02:06 : The number of buffers is: 4
07:02:06 : The following users have been enabled: ALL
07:02:06 : The following have been enabled:
        Events: ALL
        Event types: ALL
OK,
Message from AUDITOR (User 42): The security auditor is running.
```

The security monitor also sends a message to the supervisor terminal after every change in the status of the Audit Collection facility. For example, after the phantom CPL program disabled the monitoring of successful operations, the following message was displayed at the supervisor terminal:

```
[SECURITY_MONITOR Rev. 22.0 Copyright (c) Prime Computer, Inc. 1988]

08:45:14 : The following have been disabled:
        Events: ALL
        Event types: SUCC
```

Likewise, after SECURITY_MONITOR -STOP, the monitor process issues the following message to the supervisor terminal, requiring a response, before it logs out:

```
[SECURITY_MONITOR Rev. 22.0 Copyright (c) Prime Computer, Inc. 1988]

Do you really wish to stop the SECURITY_MONITOR (yes/no)?: YES
OK,
Message from Auditor (user 42): The security monitor has been
                                shut down.
```

Note that the above verification prompt accepts only a full-word YES or NO. Less than a full word elicits the following message:

```
Only yes or no are acceptable responses.
Do you really wish to stop the SECURITY_MONITOR (yes/no)?:
```

Finally, if the Security Audit facility is still active when you attempt a shutdown of partition(s) or of the system, the following message is displayed at the supervisor terminal:

```
Auditing must be stopped before shutting down.
```

Note

You cannot stop the security monitor by logging out the phantom Auditor, either by name or by user number. If you attempt this at the supervisor terminal, PRIMOS rejects the attempt but does not report an error to indicate that your attempt was unsuccessful.

Auditing Particular Users: Each day at about 9:30 a.m. the System Administrator issues another SECMON command to suit the needs of that day. Today Barry Toan, a new teller, has started work, and the SA wants to make sure his starting problems do not overload the audit facility.

Since an audit of events may not be isolated to a single user, but must apply to all users, the SA chooses a set of events and event types to ignore Barry's problems and still provide a good security audit. Already the system is auditing only the problem operations of all users. If Barry is learning the system, most of his problems will involve simple attaches. The SA decides to audit only the problems that deal with file system, system, and privileged operations:

```
OK, SECURITY_MONITOR -EVENTS ATTACH -DISABLE
```

The facility displays on the supervisor terminal the following notification of status change:

```
[SECURITY_MONITOR Rev. 22.0 Copyright (c) Prime Computer, Inc. 1988]

09:30:24 : The following have been disabled
        Events: ATCH
        Event types: ALL
```

Periodically during the day the SA calls for a report on the status of the facility. See the next section for details on audit reports.

After-hours Security: The SA knows that system activity later in the day is rather low, but system security at this time is most crucial. Just before she leaves for the day, she issues the command

```
OK, SECURITY_MONITOR -ENABLE -EVENTS ALL -EVENT_TYPES ALL
```

Now all users are audited for all events and event types, just as at system startup.

Audits and System Shutdown: The System Administrator knows that the SECMON -STOP command must be issued before the system can be shut down. To make sure that no user remains active after the security audit has been stopped, she has instructed the after-hours operator to shut down the system in the following manner:

1. To prevent further logins, issue STOP_LSR.
2. Issue a message to all users warning them that the system is shutting down and they have five minutes to log out.
3. After five minutes, issue STAT USERS and if necessary give another warning to log out.
4. If necessary, log out any users still on the system.
5. Issue the SECMON -STOP command.
6. Shut down the system.

The SA maintains a supervisor terminal that prints hard copy. She therefore can look at the hard copy in the morning to tell whether there was a significant amount of time between the termination of the Audit Collection facility and the final system message of *** PRIMOS NOT IN OPERATION ***.

AUDIT REPORTING FACILITY

The Audit Reporting facility within the Security Audit facility uses two commands. The first command, SECURITY_STATUS, provides a status report on the Collection facility itself, and indirectly reports on users. The second command, PRINT_SECURITY_LOG, provides an information report on an audit file that is normally closed. (A closed audit file is one that is not actively receiving collected audit trails from the system buffers.)

The SECURITY_STATUS Command

The SECURITY_STATUS command provides information on the status of audit collection. SECURITY_STATUS is a valid command only for the System Administrator or from the supervisor terminal. The command, without options, elicits a summary display of all users being audited and of the events and event types being audited. The command, with options, displays the specific information requested.

After the following command description, some examples show uses of the command to provide collection status.

Format

SECURITY_STATUS [*options*]
SECST

Options

-LIST_USERS
-LU
Displays a list of all users being audited.

-LIST_EVENTS
-LEV
Displays a list of event types open for audit.

-GETF
Retrieves the name of the log file open for audit collection.

-HELP
-H
Prints help information.

Examples

The following situations show the use of the SECURITY_STATUS command to generate brief online status reports.

SECURITY_STATUS Command With No Options: The System Administrator of the system at the hypothetical bank regularly keeps two separate audit logs for each day: one for that period from cold start until the bank opens and another for the rest of the workday.

The employees have arrived and logged in to the system, aided by the second SECURITY_MONITOR command that increased buffers and turned off audits of attaches. Now, before issuing the third SECURITY_MONITOR command suited for that day, the SA checks to see if security audits are in good order. She issues the command SECURITY_STATUS without any options. This command generates on her terminal a brief report summarizing the status of the security monitor. The status report for July 5 shows the following:

```
OK, SECURITY_STATUS

  [SECURITY_STATUS Rev. 22.0 Copyright (c) Prime Computer, Inc. 1988]

  STATUS OF THE SECURITY MONITOR:

      Auditor process:       AUDITOR (user 42)
      Users monitored:       BONWIT, IRENE, JASON,
                             ROGER, SCROOGE, SYSTEM,
                             TOAN, USER12
      Events monitored:      File System: NO ACCESS, FAILURE
                             System:      NO ACCESS, FAILURE
                             Priv Ops:    NO ACCESS, FAILURE
      # buffers configured: 10
      # buffers written:    24
      Audit trail file:      <BANKIT>AUDITS>SECLOG1.880705.084514
```

SECURITY_STATUS Command With Options: The CPL program has switched to the audit file for the workday, and the SA has already issued the SECMON command tailored for that day. Now she issues a SECURITY_STATUS command with the -GETF option, to check at what time the CPL program opened the audit file for that workday:

```
OK, SECURITY_STATUS -GETF

  [SECURITY_STATUS Rev. 22.0 Copyright (c) Prime Computer, Inc. 1988]

     Audit trail file:      <BANKIT>IRENE>SECLOG1.880705.084514
```

The SA can now see from the final suffix on SECLOG1 that the file was opened at 8:45, after 14 seconds.

The PRINT_SECURITY_LOG Command

While the SECURITY_STATUS command gives the status of the Audit Collection facility, it does not provide actual audit trail information. It verifies that audit trails are being generated according to directives given with the SECURITY_MONITOR command.

PRINT_SECURITY_LOG is the command to generate a report of the audit trail information that has been gathered. The PRINT_SECURITY_LOG command normally must address a closed file on disk.

Note

You may submit an open file to PRINT_SECURITY_LOG, if you have first used the RWLOCK -UPDT command on that file. To do so, you must have the Protect (P) access right to the directory.

If the file is already saved to tape, you must first recover it. (See the TRANSFER_LOG command in the section Audit File Backup Facility later in this chapter.) The PRINT_SECURITY_LOG command elicits a terminal display. To get a printable copy, open a COMO file before issuing the PRINT_SECURITY_LOG command.

A report for a large audit file can tie up your terminal for a few minutes. To avoid this delay, create a phantom CPL program to open the COMO file and issue the PRINT_SECURITY_LOG command. In this situation, be sure to include the -NO_WAIT option so that output does not halt at the first —More— prompt. A COMO file with the -NO_WAIT option also suppresses the repetition of the header after the —More— prompt.

An experienced user of PRINT_SECURITY_LOG may use the -NO_HEADER option to suppress headers on screen output, thereby providing each screen an extra audit trail.

Format

PRINT_SECURITY_LOG **-LOGFILE** ***pathname*** [***options***]
PSLOG **-LOG**

pathname is the pathname of an audit file whose contents are to be displayed. The file normally must be a closed file on disk. If already backed up to tape, the file first must be restored to disk.

Options

-USERS [***userid-list***]
-US

Reports only those records generated by the users in ***userid-list***. The ***userid-list*** may contain a maximum of sixteen user IDs separated by blanks. If no ***userid-list*** follows the -USERS option, then all users' records are reported.

-NUMBER_OBJECT [*num-obj-list*]
-NUMOBJ
Reports on only those audited number objects that are specified in *num-obj-list*. The *num-obj-list* may contain a maximum of sixteen number objects separated by blanks. If no *num-obj-list* follows -NUMBER_OBJECT, all number objects are reported. Should there be none to report, the message `No Events Selected` is returned.

A number object may be a positive number, such as a file unit number or a segment number. It may be a negative number, such as an imaginary address used with an EPF. See the second example of PRINT_SECURITY_LOG output later in this chapter.

-TEXT_OBJECT [*text-obj-list*]
-TEXTOBJ
Reports on only those audited text objects that are specified in *text-obj-list*. The *text-obj-list* may contain a maximum of sixteen text objects separated by blanks. If no *text-obj-list* follows -TEXT_OBJECT, all text objects are reported.

A text object may be a device name, by designating a subdirectory to DEVICE*. The text object may be a pathname — either a full or a partial pathname. Thus the SA can generate a report for those actions affecting a particular directory or subdirectory.

-EVENTS [*arguments*]
-EV
Selects the set of event categories to be reported. If you omit this option, a report for all events specified below is displayed. After the option, you specify one or more of the event category arguments.

These arguments are

FILE_SYSTEM
FS
States that file system events are to be reported.

SYSTEM
SYS
States that system events are to be reported.

PRIV_OPS
PRIV
States that events involved with privileged operations are to be reported.

ATTACHES
ATCH
States that attach operations are to be reported.

-EVENT_TYPES [*arguments*]
-EVTYPE
Specifies the event types to be reported. If you omit this option, all event types are reported.

After -EVENT_TYPES, you specify one or more of the event type arguments. These arguments are

SUCCESS
SUCC
States that successful events are to be reported.

NO_ACCESS
NOACC
States that access failure events are to be reported.

FAILURE
FAIL
States that failed events are to be reported.

-NO_WAIT
-NW
Specifies that screen displays are not to stop at —More— prompts on the screen, but to continue to the end of the report. Phantoms used to create COMO files must use this option.

-NO_HEADER
-NHE
Suppresses the table header (normally for screen displays).

-HELP
-H
Displays a summary of the options for this command.

Interpretation of an Audit Trail

The output from the PRINT_SECURITY_LOG command has the general format shown below.

A title header identifies the data that is supplied in each audit trail. Each audit trail requires three lines. The first column in the title header identifies the type of data that begins each of the lines. A blank line separates each audit trail. A typical header and audit trail is

```
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
Event Group  Date/Time          Type    User#   Code     User
Description
Object(s)
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

ATCH         88-12-25 00:06:29   SUCC       10      0     SANTA
Attach to directory
<PRIME>NICE>LIST
```

The sample audit trail shows a successful attach. It contains the following:

- The Event Group column (on the first line of the audit trail) shows ATCH to indicate that the action was an attach.
- The Date/Time column indicates that the action occurred on December 25, 1988, at six minutes and 29 seconds after midnight.
- The (Event) Type column shows SUCC to indicate that the action was successful.
- The User# column indicates the source of the action as user number 10.
- The Code column shows 0 to indicate a successful attach. Codes for an unsuccessful action may be found in the standard system error codes from the ERRD.INS.*language_name* files in SYSCOM.
- The User column indicates that user number 10 has the user ID of SANTA. It is important to point out that, for all ensuing audit trails, the user SANTA with a user number 10 may be distinguished from any other user SANTA with another user number.
- The Description column (the second line of the audit trail) indicates that the action performed was an attach to a directory.
- The Object(s) column (beginning on the third line) shows the text object of the attach to be <PRIME>NICE>LIST. If a number object were associated with this file, it would also be shown here.

If the System Administrator IRENE is audited as she logs in, several audit trails are generated for this one action. An audit trail would show an attach to the SAD, but the User column would show the notation `** userid not set **`. This audit trail would show the Login server's first action to admit a new user.

Ensuing audit trails for this same user number show how the Login server accesses various files in the SAD to verify the password for IRENE, to establish her command environment limits, and to identify her Initial Attach Point. The Login server uses the same user number for all these actions. When user IRENE is successfully logged in to the system, she inherits the same user number thus far used by the Login server.

Note

Certain system services, such as Batch and DSM, generate many audit trails. To eliminate such audits the SA may use SECURITY_MONITOR to disable audits on the DSM servers, BATCH_SERVICE, and so on.

The following example illustrates some of the many audit trails involved in the login procedure for IRENE.

```
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
Event Group  Date/Time          Type    User#   Code     User
Description
Object(s)
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

ATCH         88-07-05 07:04:23   SUCC       17      0     **userid not set**
Attach to directory
<OP_SYS>SAD

FS           88-07-05 07:04:23   SUCC       17      0     **userid not set**
Open file
<OP_SYS>SAD>UVF Unit requested: 32 Unit assigned: 32

FS           88-07-05 07:04:23   SUCC       17      0     **userid not set**
Open file
<OP_SYS>SAD>MPF Unit requested: 31 Unit assigned: 31

FS           88-07-05 07:04:23   SUCC       17      0     **userid not set**
Check existence of file
<OP_SYS>SAD>SDF

FS           88-07-05 07:04:23   SUCC       17      0     **userid not set**
Open file
<OP_SYS>SAD>MGF Unit requested: 30 Unit assigned: 30

ATCH         88-07-05 07:04:23   SUCC       17      0     **userid not set**
Relative attach
<OP_SYS>SAD>DEFAULT

FS           88-07-05 07:04:23   SUCC       17      0     **userid not set**
Open file
<OP_SYS>SAD>DEFAULT>PVF Unit requested: 29 Unit assigned: 29

FS           88-07-05 07:04:23   SUCC       17      0     IRENE
Open file
<OP_SYS>SAD>DEFAULT>PDF Unit requested: 28 Unit assigned: 28

FS           88-07-05 07:04:23   SUCC       17      0     IRENE
Open file
<OP_SYS>SAD>DEFAULT>PPPF Unit requested: 27 Unit assigned: 27

FS           88-07-05 07:04:23   SUCC       17      0     IRENE
Open file
<OP_SYS>SAD>DEFAULT>MPP Unit requested: 26 Unit assigned: 26

ATCH         88-07-05 07:04:23   SUCC       17      0     IRENE
Attach to directory
<BANKIT>IRENE

FS           88-07-05 07:04:23   SUCC      17       0     IRENE
Close file by unit
28
```

```
FS          88-07-05 07:04:23   SUCC      17       0      IRENE
Close file by unit
27

FS          88-07-05 07:04:23   SUCC      17       0      IRENE
Close file by unit
30

FS          88-07-05 07:04:23   SUCC      17       0      IRENE
Close file by unit
26

FS          88-07-05 07:04:23   SUCC      17       0      IRENE
Close file by unit
31

FS          88-07-05 07:04:23   SUCC      17       0      IRENE
Close file by unit
32

FS          88-07-05 07:04:23   SUCC      17       0      IRENE
Close file by unit
29

ATCH        88-07-05 07:04:23   SUCC      17       0      IRENE
Attach to directory
<OP_SYS>CMDNCO

SYS         88-07-05 07:04:23   SUCC      17       0      IRENE
Return segment/delete EPF
-4
     .
     .
     .
```

Note

The final negative segment number in the example above is returned to delete an EPF. This number was the signed imaginary address that BIND had presented to PRIMOS. The minus sign indicates that it was data, and PRIMOS had earlier resolved its imaginary address, searching out and initializing the user information in project DEFAULT. Now PRIMOS returns the imaginary address. For more information on how the system handles EPFs, see the *Advanced Programmer's Guide, Volume I.*

You might also use PRINT_SECURITY_LOG with the -NUMBER_OBJECT option to tailor the report to assist in summarizing audit trails of interest. The option specifies a report output that has sifted out all instances of activity for number objects: the opening and closing of file units, the allocation and release of segments, and the generation and resolution of EPFs.

For example, the System Administrator, looking at a very large PRINT_SECURITY_LOG report, notices that certain units have been opened for a particular pathname. The SA wants to know what other actions occurred for these units, but wants to avoid searching several pages of the report to find this information. The SA can follow the -NUMBER_OBJECT option with a maximum of 16 unit numbers to elicit a summary report of what happened to these units.

The next example of PRINT_SECURITY_LOG output is a reduced version of the previous example, which showed the full audits on the login procedure for user IRENE. The result of using the -NUM_OBJECT option with PRINT_SECURITY_LOG on this audit of startup is as follows:

```
OK, PSLOG -LOG SECURITY_LOG.880705@@ -NUM_OBJECT 26 27 28 29
[PRINT_SECURITY_LOG Rev. 22.0 Copyright (c) 1988, Prime Computer, Inc.]

SECURITY_MONITOR started at 88-07-05 07:04:23

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
Event Group  Date/Time          Type   User#    Code     User
Description
Object(s)
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

FS           88-07-05 07:04:23   SUCC      17       0     **userid not set**
Open file
<OP_SYS>SAD>DEFAULT>PVF Unit requested: 29 Unit assigned: 29

FS           88-07-05 07:04:23   SUCC      17       0     IRENE
Open file
<OP_SYS>SAD>DEFAULT>PDF Unit requested: 28 Unit assigned: 28

FS           88-07-05 07:04:23   SUCC      17       0     IRENE
Open file
<OP_SYS>SAD>DEFAULT>PPPF Unit requested: 27 Unit assigned: 27

FS           88-07-05 07:04:23   SUCC      17       0     IRENE
Open file
<OP_SYS>SAD>DEFAULT>MPP Unit requested: 26 Unit assigned: 26

FS           88-07-05 07:04:23   SUCC      17       0     IRENE
Close file by unit
28

FS           88-07-05 07:04:23   SUCC      17       0     IRENE
Close file by unit
27

FS           88-07-05 07:04:23   SUCC      17       0     IRENE
Close file by unit
26

FS           88-07-05 07:04:23   SUCC      17       0     IRENE
Close file by unit
29
    .
    .
    .
```

More Examples of Using PRINT_SECURITY_LOG

The following situations show the use of PRINT_SECURITY_LOG to generate reports of offline audit files.

Audit File Screen Reports: The System Administrator of the system at the hypothetical bank wants to generate an audit file report for the previous Friday's time interval from cold start to opening time. She wants to generate another audit file report of audits for the remainder of that workday.

She normally keeps these two types of daily logs in separate directories on disk.

The startup audit files are those files with the name SECURITY_LOG.[DATE] stored in the current directory, which is usually the directory CMDNC0 for the supervisor terminal. Each day she moves that day's startup audit file to her own subdirectory STARTS.

The audit files for the rest of each workday are located in the directory where they were opened: <BANKIT>AUDITS. Each file is opened there as SECLOG1.[DATE] by the CPL program that executes at 8:45 a.m. The SA may open a second audit file for a given day as SECLOG2.[DATE]. She thereby closes the first audit file and makes it available for an immediate report.

Once a month the SA backs up both sets of audit files to magnetic tape, so as to preserve disk space. (See the Audit File Backup Facility section later in this chapter for details.) The SA now wants a report of all activities from the previous Friday's startup audit file. First the SA opens a COMO file called FRIDAY1.COMO, in order to print out the report after it is generated. She issues the command to generate the report at her terminal, realizing the report may tie up her terminal for a few minutes.

```
OK, COMO FRIDAY1.COMO
OK, PSLOG  -LOGFILE *>STARTS>SECURITY_LOG.880701@@ -NO_WAIT
```

The report displayed at the System Administrator's terminal may be interrupted using QUIT at any —More— prompt, or Ctrl P to interrupt a report with the -NO_WAIT option. After such an interrupt, PSLOG reports the unread portion of the file, using the following format:

```
PSLOG has read n records out of x from logfile logfile-name.
Type START to continue processing
```

Audit File Printed Reports: The SA obtains a printed copy by closing the COMO file and printing it. The following example shows some of the printer output from this PRINT_SECURITY_LOG command. Because the -NO_WAIT option was specified, only the first page has a header.

```
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
Event Group  Date/Time              Type   User#    Code     User
Description
Object(s)
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

ATCH         88-07-01 07:00:36      SUCC      41       0     DSMASR
Attach to directory
<OP_SYS>DSM*

ATCH         88-07-01 07:00:36      SUCC      41       0     DSMASR
Relative attach
<OP_SYS>DSM*>LOGS

ATCH         88-07-01 07:00:36      SUCC      41       0     DSMASR
Relative attach
<OP_SYS>DSM*>LOGS>UMH

FS           88-07-01 07:00:36      SUCC      41       0     DSMASR
Check existence of file
<OP_SYS>DSM*>LOGS>UMH>DEFAULT.LOG
        .
        .
        .

SYS          88-07-01 07:00:44      SUCC       4       0     DSMSR
Return segment/delete EPF
4402

PRIV         88-07-01 07:01:04      SUCC       1       0     SYSTEM
Check for privileged process
TMR$STI

FS           88-07-01 07:01:08      SUCC       1       0     SYSTEM
Close file by unit
7

PRIV         88-07-01 07:01:08      SUCC       1       0     SYSTEM
MAXUSR command
Maximum number of users: 255

ATCH         88-07-01 07:01:08      SUCC       1       0     SYSTEM
Attach to directory
<OP_SYS>SEARCH_RULES*

FS           88-07-01 07:01:08      FAIL       1       5     SYSTEM
Open file
<OP_SYS>SEARCH_RULES*>ADMIN$.COMMAND$.SR Unit requested: -10000
Unit assigned: -10000
        .
        .
        .

FS           88-07-01 07:01:08      SUCC       1       0     SYSTEM
Open file
<OP_SYS>SEARCH_RULES*>COMMAND$.SR Unit requested:  32 Unit assigned:  32

ATCH         88-07-01 07:01:08      SUCC       1       0     SYSTEM
Attach home
```

```
FS          88-07-01 07:01:08      SUCC       1       0     SYSTEM
Close file by unit
32

FS          88-07-01 07:01:08      SUCC       1       0     SYSTEM
Open file
<OP_SYS>CMDNCO>TOOLS.COMMAND$.SR Unit requested:  32 Unit assigned:  32

FS          88-07-01 07:01:08      SUCC       1       0     SYSTEM
Close file by unit
32
        .
        .
        .
ATCH        88-07-01 07:04:23      SUCC      17       0     **userid not set**
Attach to directory
<OP_SYS>SAD
        .
        .
        .
```

The SA expects Friday's audit file for the entire workday to be quite large. She doesn't want to tie up her terminal while the report is generated, so she runs a phantom CPL program to open another COMO file and then generate the security log for the day. She adds the -NO_WAIT option to the PRINT_SECURITY_LOG command to ensure that the entire report is generated inside the COMO file. The CPL program holds the following lines:

```
COMO IRENE>WRKDAY.880701.COMO
PSLOG -LOG <BANKIT>AUDITS>SECLOG1.880701@@ -NO_WAIT
COMO -E
```

After printing the above COMO file later in the day, the SA also creates a smaller report for the PAYROLL group. They request a daily report of any activity carried out in the subdirectory CHECKS. The SA generates such a PSLOG report by using the -TEXT_OBJECT option followed by a partial pathname:

```
PSLOG -LOG <BANKIT>AUDITS>SECLOG1.880701@@ -TEXTOBJ <BANKIT>PAYROLL>CHECKS
```

The SA first generates a terminal display for the above. If any such audit trails are reported, the SA then creates a hard-copy report.

Excess Audit Trails for TOOLS Directory: If an audit file shows that every privileged PRIMOS command (issued at the supervisor terminal or by the SA) first causes a failed attempt to open its command runfile in the top-level directory TOOLS, then the order of COMMAND$ search rules may need alteration to eliminate the extra audit trails. The SA may verify the sequence of the COMMAND$ search rules as follows.

```
OK, LIST_SEARCH_RULES COMMAND$
      .
      .
      .
    CMDNC0
    TOOLS
OK,
```

The above sequence of rules is most efficient. If TOOLS precedes CMDNC0, the SA may reverse their order by creating the private COMMAND$ search rule to TOOLS according to the directions given in Chapter 7. (Avoid using either a subroutine or the -NO_SYSTEM and -SYSTEM options.)

AUDIT FILE BACKUP FACILITY

The Audit File Backup facility allows the privileged operator or System Administrator to perform two tasks:

- Audit file backups (the transfer of audit files from disk to backup tape)
- Audit file recoveries (the transfer of audit files from backup tape to disk)

Use the TRANSFER_LOG utility to perform both of these tasks. This utility is supplied in the top-level directory TOOLS to customers who have purchased the Security Audit facility. This utility normally processes closed audit files, that is, it does not make a backup copy of an audit file that is still gathering audit trails.

After collecting a number of separate audit files on disk, use this utility to back up the files to tape. After using TRANSFER_LOG to create backup copies, you can delete the original disk files, making disk space for additional audit files.

If it is necessary to recover audit files from a given tape, use this utility to retrieve copies for later formatting by means of the PRINT_SECURITY_LOG command.

Note

Submit to bulk erasure those tapes that hold backup files no longer needed. Always bulk erase used tapes before making them available as scratch tapes.

The TRANSFER_LOG utility is described below, followed by examples of its use.

The TRANSFER_LOG Utility

The TRANSFER_LOG utility does not accept arguments on the command line. While the utility is not a PRIMOS command, you may use it as a command if you have created the private COMMAND$ search rule to TOOLS for the supervisor terminal and SA, using directions given in Chapter 7. You are prompted for arguments after using the command.

Format

TRANSFER_LOG
TLOG

Description

When you invoke the TRANSFER_LOG command, a session similar to the following example takes place:

```
OK,TRANSFER_LOG

[TRANSFER_LOG Rev. 22.0 Copyright (c) Prime Computer, Inc. 1988]
File names may be described as :

A PRIMOS File or Tree/Pathname for disk files
@MTn for tape files, where "n" is the tape drive unit number.

Please enter name of the Source File: SECURITY_LOG.880601.070034
Please enter name of the Destination File : @MT0:1

Transfer complete
OK,
```

If the file is located on disk, you may use a full PRIMOS pathname to indicate the source file. In the above example, the source file is an audit file with a default name, in the form of SECURITY_LOG.*yymmdd.hhmmss*. The suffix *yymmdd.hhmmss* is a time stamp, with the date and time in the same format as for the PRIMOS command function [DATE -FTAG].

The operator has mounted a tape on drive unit number zero and so indicates *@MT0* for a destination. (Valid unit numbers range from 0 through 7.) Immediately after the drive unit number, the operator may optionally specify a particular logical tape on that physical tape by appending a colon and number. In this transfer example, the operator specifies logical tape one on the tape mounted on drive unit zero: *@MT0:1*. A zero after the colon is a command to write immediately. In such a case, the tape begins writing at that immediate point; it does not first advance to beyond the last logical tape.

The above example is a backup transfer. A recovery transfer would indicate @MT*n* for the source file and a PRIMOS pathname for the destination file.

Example of Using the TRANSFER_LOG Utility

The System Administrator wants to back up all the startup audit files stored in STARTS for the month of June. She mounts a scratch tape on Magnetic Tape Unit 0 (MT0) and assigns it to herself.

The Audit File Backup facility addresses a single file at a time for recoveries or backups, so the SA will invoke TRANSFER_LOG once for each file that she wishes to back up. She does the first three backups manually:

```
OK, ASSIGN MT0
device mt0 assigned
OK, TRANSFER_LOG/*SA is already attached to <BANKIT>IRENE>STARTS

[TRANSFER_LOG Rev. 22.0 Copyright (c) Prime Computer, Inc. 1988]
File names may be described as:

A PRIMOS File or Tree/Pathname for disk files
@MTn for tape files, where "n" is the tape drive unit number.

Please enter the name of the Source File : SECURITY_LOG.880601
Please enter the name of the Destination File : @MT0:1

Transfer complete.

OK, TRANSFER_LOG/*Now backing up June 2 to logical tape number 2.

[TRANSFER_LOG Rev. 22.0 Copyright (c) Prime Computer, Inc. 1988]
File names may be described as:

A PRIMOS File or Tree/Pathname for disk files
@MTn for tape files, where "n" is the tape drive unit number.

Please enter the name of the Source File : SECURITY_LOG.880602
Please enter the name of the Destination File : @MT0:2

Transfer complete.

OK, TRANSFER_LOG/*Now backing up June 3 to logical tape number 3.

[TRANSFER_LOG Rev. 22.0 Copyright (c) Prime Computer, Inc. 1988]
File names may be described as:

A PRIMOS File or Tree/Pathname for disk files
@MTn for tape files, where "n" is the tape drive unit number.

Please enter the name of the Source File : SECURITY_LOG.880603
Please enter the name of the Destination File : @MT0:3

Transfer complete.

OK,
```

The System Administrator uses a CPL program to simplify the repetitive process of backing up multiple audit files. The SA has two versions, one for the startup audit files and one for the workday files.

The version for startup audit files has the default base name SECURITY_LOG coded into the program. Each month the SA must alter the code for that month. The CPL program contains other warning comments, as shown below:

```
/*CPL program to simplify monthly backups of disk files to tape.
/*SABU.CPL  (STARTUP-AUDIT-BACK-UPS)
/*  Each invocation of TRANSFER_LOG requires two responses to
/*       interactive prompts for Source File and Destination File.
/*       User hereby may give abbreviated responses on the command
/*       line without waiting for the subsystem.
/*
/*Procedure: All Source Files begin with SECURITY_LOG.88mm
/*             The mm digits designate the month presently
/*             undergoing backup.  User must supply the
/*             the %day_num% digits after mm for each source
/*             getting backed up.
/*           All Destination Files begin with @MT0:
/*             after the colon is the %tp_file_num% for this tape backup
/*
/*WARNINGS: Before running it for a given month, change the two digits
/*             before %day_num% below to the digits for the month
/*             undergoing backup.
/*          Be sure that your tape is mounted on MT0 and that you
/*             have already assigned MT0.
/*
/*    USAGE:  SABU day_num tp_file_num
/*      (If the program is not in TOOLS, use RESUME etc.)
/*
&ARGS DAY_NUM; TP_FILE_NUM
/*
&DATA TRANSFER_LOG                /* Invoke TRANSFER_LOG
   SECURITY_LOG.8806%day_num%     /* Source file
   @MT0:%tp_file_num%             /* Logical tape number on backup tape
&end
&return
```

The SA can resume the above program. However, since she created it as a private C2 utility, she has added it to the top-level directory TOOLS. She therefore invokes it as a command. The following example shows some ensuing TRANSFER_LOG backups of the June audit files, now using the above CPL program.

```
OK, /* Program provides TRANSFER_LOG the  file names it needs:
OK, SABU 06 4 /* Now backing up June 6 to logical tape number 4.

[TRANSFER_LOG Rev. 22.0 Copyright (c) Prime Computer, Inc. 1988]
File names may be described as:

A PRIMOS File or Tree/Pathname for disk files
@MTn for tape files, where "n" is the tape drive unit number.

Please enter the name of the Source File : SECURITY_LOG.880606
Please enter the name of the Destination File : @MT0:4

Transfer complete.
```

```
OK, SABU 07 5 /* Now backing up June 7 to logical tape number 5.

[TRANSFER_LOG Rev. 22.0 Copyright (c) Prime Computer, Inc. 1988]
File names may be described as:

A PRIMOS File or Tree/Pathname for disk files
@MTn for tape files, where "n" is the tape drive unit number.

Please enter the name of the Source File : SECURITY_LOG.880607
Please enter the name of the Destination File : @MT0:5

Transfer complete.

OK, SABU 08 6/* Now backing up June 8th to logical tape number 6.

[TRANSFER_LOG Rev. 22.0 Copyright (c) Prime Computer, Inc. 1988]
File names may be described as:

A PRIMOS File or Tree/Pathname for disk files
@MTn for tape files, where "n" is the tape drive unit number.

Please enter the name of the Source File : SECURITY_LOG.880608
Please enter the name of the Destination File : @MT0:6

Transfer complete.
OK, /* The SA backs up each audit log for that month.
    .
    .
    .
Transfer complete.
OK,
```

THE CRASH AUDIT RECOVERY FACILITY

The Crash Audit Recovery facility enables the recovery of audit trails held in system buffers at a system halt. The CRASH_AUDIT utility ensures that the system buffers holding these audit trails may be written to a new file that complements the original security audit file. The file of recovered audits must have a name different from the original security audit file. For example, SECLOG1.RECOV.[DATE] holds the recovered audit trails that complete the original security audit file SECLOG1.870706@@. The SA uses the CRASH_AUDIT utility as the last step in using the Crash Audit Recovery facility.

In order to use the Crash Audit Recovery facility, you must

- Maintain a RING0.MAP loadmap file on the system, normally located in the directory LOAD_MAPS*. (Maps for revisions of PRIMOS previous to Rev. 22.0 may exist in either PRIRUN or MAPS.)
- Take a tape dump after every unplanned system halt, before a cold start of the system. See your CPU handbook for tape dump directions.
- After the system is cold started, submit the tape dump to the CRASH_AUDIT utility.

The CRASH_AUDIT Utility

Th CRASH_AUDIT utility is supplied in the directory TOOLS to customers who have purchased the Security Audit facility. While not a PRIMOS command, the CRASH_AUDIT utility may be used as a command if you have created the private COMMAND$ search rule to TOOLS for the supervisor terminal and SA, using directions given in Chapter 7.

Format

CRASH_AUDIT *arguments*

Arguments

-MT *n*
Specifies that the tape dump is to be read from magnetic tape unit *n*, where values for *n* range from 0 through 7.

-DUMPFILE *pathname*
Specifies that the tape dump is to be read into the disk file with the specified pathname.

-OUTFILE *pathname*
Specifies that the audit records are to be written to the security audit file with the specified pathname. The name must differ from the name of the original security audit file interrupted by the system halt.

-MAP *pathname*
Specifies the pathname for RING0.MAP. If you do not give this pathname, the default is LOAD_MAPS*>RING0.MAP.

Example of Using the CRASH_AUDIT Utility

On the afternoon of Tuesday, July 5, the SA notices that

- A halt message is displayed at the supervisor terminal.
- The supervisor terminal is in Control Panel mode, as indicated by the CP> prompt.
- The STOP light on the front panel is on, indicating that the CPU is not running.

The SA is certain from these observations that a system halt, not just a hang, has occurred. The SA prepares to do a tape dump. First the SA loads a tape onto Magnetic Tape Unit 0. Then the SA issues the following Virtual Control Panel command to assign the tape unit and to write memory buffers out to tape:

```
CP> TAPEDUMP 0
```

The audit trails, and all other data in memory, are now stored on the tape.

Later, after the system problem is corrected, the SA cold starts the system. When the system is fully stabilized, the SA issues the following single-line command to invoke the CRASH_AUDIT utility:

```
OK, CRASH_AUDIT -MT 0 -DUMPFILE RECOV.880705
-OUTFILE <BANKIT>AUDITS>SECLOG1.RECOV.880705 -MAP LOAD_MAPS*>RING0.MAP
```

First the tape dump is written from the tape, mounted on Magnetic Tape Unit 0, to the pathname on disk specified after -DUMPFILE. After the dumpfile is written to disk, the CRASH_AUDIT utility finds the unrecorded audit trails and copies them to the file specified after -OUTFILE. The SA has maintained the file RING0.MAP in the directory LOAD_MAPS*. The CRASH_AUDIT utility uses this file, specified after the -MAP option, to retrieve those audit trails.

Now two completed security audit files hold the audits for the workday of July 5. Furthermore, when the system is cold started for the second time that day, the command line for SECURITY_MONITOR within PRIMOS.COMI initializes a third audit file for that day, using the default name of SECURITY_LOG.[DATE] and residing in CMDNC0.

APPENDICES

A

EXTERNAL LOGIN AND LOGOUT PROGRAMS

Since Rev. 19.0, you can write separate external programs to monitor and control logins and logouts. (Prior to Rev. 19.0, 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

In addition to the internal login facility provided by EDIT_PROFILE and PRIMOS, the external login program may also regulate the use of the operating system. 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.

You should consider the following factors in designing 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 suffixes are allowed on either name.
- 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.

- Ctrl 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.

SAMPLES: EXTERNAL LOGIN AND EXTERNAL LOGOUT

The following samples illustrate external LOGIN and LOGOUT programs written in Pascal. After each sample program there is an additional sample COMINPUT program that will compile and load each program into its proper location within CMDNC0.

The external LOGIN program is as follows:

```
{ External Login Program: LOGIN.PASCAL                                  }
{   See the ensuing COMINPUT program for program compilation, its       }
{   SEG -LOAD, and its relocation into CMDNC0 as the renamed "LOGIN".   }
{ Purpose:                                                              }
{ This program screens incoming logins.  It immediately admits local    }
{ users, but it requires an extra validation code from those entering   }
{ from the network (whether PRIMENET or NTS).                           }
```

```
PROGRAM MAIN (INPUT,OUTPUT);

CONST
   turn_echo_off = -8192;      { Terminal in half duplex, XOFF enabled }
   convert_to_upcase = 1;      { Convert validation code to upper case }
   len = 6;                    { Length of passwd in chars             }
   from_remote_user = 3;       { Value for any user coming thru net    }
   success = 0;                { To verify Subroutine return codes     }
   low_nts_line = 1024;        { lowest-numbered NTS line              }
   high_nts_line = 1536;       { highest-numbered NTS line             }

TYPE
   val_code_type = PACKED ARRAY [1..len] OF CHAR;
   lts_name_type = STRING[16];
   mac_type = PACKED ARRAY[1..16] of char;

VAR
   code : integer;             { status returned: as$lin, etc.         }
   this_user_is : integer;     { user login status stored here         }
   valcode : val_code_type;    { password entered by user              }
   correct_valcode : val_code_type;  { Must be "len" chars long.       }
     {Sites create a customized PRIMENET val_code within the program. }
     {As a variable, it permits being overwritten by nts_valcode.     }
   nts_valcode : val_code_type;      { Must be "len" chars long.       }
     {Sites create a customized NTS val_code  within the program.     }
   user_duplex : integer;      { initial terminal values               }
   dummy : integer;            { object for function duplx$            }
   primos_line_num : integer;  { needed to configure NTS               }
   media_form : integer;       { identifies IEEE 802.3 LAN             }
   lts_name : lts_name_type;   { configured LAN Terminal server        }
   lts_line : integer;         { port used on LTS                      }
   mac_address : mac_type;     { machine address of LTS                }
   nts_login : integer ;       { flag initialized at program start     }

PROCEDURE as$lin (A,B:INTEGER);  EXTERN;
PROCEDURE nt$lts (A: INTEGER; VAR B: INTEGER; VAR C: LTS_NAME_TYPE;
    VAR D: INTEGER; VAR E: MAC_TYPE; VAR F: INTEGER); EXTERN;
FUNCTION duplx$  (A:INTEGER): INTEGER; EXTERN;
PROCEDURE utype$ (VAR this_user_is:INTEGER); EXTERN;
PROCEDURE case$a (A:INTEGER; VAR valcode: VAL_CODE_TYPE; C:INTEGER); EXTERN;
PROCEDURE logo$$ (A,B,C,D:INTEGER; E:LONGINTEGER; VAR code:INTEGER);
    EXTERN;


BEGIN          { External LOGIN program }
   { First some initialization:...                                    }
   correct_valcode := 'XXXXXX';  {Sites may substitute a valcode...}
        { that is customized for PRIMENET.                           }
   nts_valcode := 'YYYYYY';      {Sites may substitute a valcode...}
        { that is customized for NTS.                                }
   nts_login := 1;        { Flag initialized to "failure"           }
```

```
{And now, with initializations complete...                    }
as$lin (primos_line_num, code);          { Do NTS line check: }
IF (primos_line_num > low_nts_line) AND    { If within spread...}
   (primos_line_num < high_nts_line) THEN      { for NTS lines, }
  BEGIN                  { then complete NTS configuration...  }
     nt$lts (primos_line_num, media_form,
             lts_name, lts_line, mac_address, code);
     IF (code <> success) THEN      { Unsuccessfully configured?}
        BEGIN
           writeln;  { Warn user of configuration problem...    }
           writeln('Failure in NTS access'); { within NTS,...   }
           logo$$(0,0,0,0,0,code);   {before logging out user.  }
        END   { of NTS failed line check }
     ELSE
       nts_login := 0;  {Flag successful NTS login configuration}
  END; { of NTS configuration issues }

utype$ (this_user_is);      { Now check how the user logged in. }
IF (this_user_is <> from_remote_user) THEN   {If the user is... }
  BEGIN   {NOT a PRIMENET login, then admit the LOCAL login,... }
    writeln;     {but provide a little security reminder:       }
    writeln('Notify the SA if PRIMOS reports failed login attempts');
    writeln('recorded against your user ID that you cannot explain!');
  END       {successful login for local line}

ELSE             {If the user IS from PRIMENET, then do these...}
  BEGIN  {validation procedures for a non-local line:}
    user_duplex := duplx$ (-1);   {Store current terminal values}
    dummy := duplx$ (turn_echo_off);        { Turn off Echo     }
    reset (INPUT, '-INTERACTIVE');          { Allow erase char  }
    write ('Validation Code: ');            { Now Prompt user...}
    readln (valcode);                       { Read in value     }
    dummy := duplx$ (user_duplex);          { Reactivate Echo   }
    case$a (convert_to_upcase, valcode, len); { Make all Caps   }

    {Now sort out the information you have on the remote user.  }
    {NTS users must be checked against the nts_valcode, while...}
    {other PRIMENET users must be checked against the value for }
    {correct_valcode that is originally set:                    }
    IF (nts_login = success) THEN {If it's an NTS login, then...}
      correct_valcode := nts_valcode; {switch to nts_valcode value}
    IF (valcode <> correct_valcode) THEN  {and then validate!   }

      BEGIN  {If invalid, then reject the user:                 }
        writeln;
        writeln('Sorry, invalid validation code...');
        logo$$(0,0,0,0,0,code);        { and then log user out. }
      END     { of handling wrong validation code.}

    ELSE   {Welcome the successfully logged-in remote user, but }
      BEGIN  {gove a security reminder about remote logins:     }
        writeln('Welcome!');
        writeln('Notify the SA if PRIMOS reports any unexplained');
        writeln('failed login attempts against your user ID!');
      END    { of successful login for non-local user}
    END   { of the extended IF statement  }
END.      { end of LOGIN.PASCAL }
```

The preceding program requires a small PMA module to make an interface for the two new direct entrypoints AS$LIN and NT$LTS:

```
* Interface Module: NUDNTS.PMA
    SEG
    DYNT AS$LIN
    END
    SEG
    DYNT NT$LTS
    END
```

The following COMINPUT program compiles both the LOGIN program and its PMA interface module. It then uses SEG -LOAD techniques to compress all (procedures, data, and RUNIT program) into a SAM file called EX4000. It then copies the SAM file 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.

The COMINPUT program is as follows:

```
/*  extlogin.comi
/*  compile and load sample external login program
/*  then copy to cmdnc0>login (name must not have a suffix)
pascal login -64v  /* login.pascal must be in the present directory
pma nudnts  /* nudnts.pma must be in the present directory
seg -load
split 167777 4000 150000 /* Split segment '4000, and locate the
/* procedure below '16777 and the stack starting at '150000
mix
s/lo login 0 4000 4000
s/lo nudnts 0 4000 4000
d/li vapplb
d/li paslib
d/li
map
save
return
share
ex
quit
copy ex4000 cmdnc0>login -nq
co -end
```

The following sample illustrates a LOGOUT program:

```
{ External Logout Program:  LOGOUT.PASCAL                              }
{ Note:                                                                }
{ This program must be compiled and "SEG -LOAD"ed into a SAM file      }
{ that can be relocated into CMDNC0 with the name LOGOUT, as shown     }
{ in the ensuing EXTLOGOUT.COMI file.                                  }
{ Purpose:                                                             }
{ The program says "Goodbye" to the user before the internal logout    }
{ procedure completes the disconnection.  You may wish to make         }
{ additions to the program.                                            }

PROGRAM MAIN (OUTPUT);

BEGIN
   writeln;
   writeln('Goodbye!');
   writeln;
END. {of LOGOUT.PASCAL}
```

The following COMINPUT program compiles the LOGOUT program, uses SEG -LOAD to create a SAM file called EX4000, then copies the SAM file into CMDNC0 under the name LOGOUT:

```
/*  extlogout.comi
/*  compile and load sample external logout program
/*  then copy to cmdnc0>logout (name must not have a suffix)
pascal logout -64v  /* login.pascal must be in the present directory
seg -load
split
mix
s/lo logout 0 4000 4000
d/li vapplb
d/li paslib
d/li
map
save
return
share
ex
quit
copy ex4000 cmdnc0>logout -nq
co -end
```

B

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 italicized (for example, *user_id*).

The bracketed word at the end of each explanation indicates that 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 message indicates that an error was caused by one of the following:

- 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 either case, 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 restoration of the SAD was done with MAGRST.
- An incomplete copy of the SAD was done with COPY.

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.`
You have a password SAD that you created at the supervisor terminal. It offers limited security that could be improved if you used ACLs instead of a password on the SAD. [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 specified.) 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 access project DEFAULT, protection has been damaged.`
The ACLs have been changed on this project within the SAD. If the problem is with the setting for $REST, then the system itself corrects the problem, and you need only reissue the command. If a more substantial change in ACLs has been made, you may need to use the SET_DEFAULT_PROTECTION subcommand within EDIT_PROFILE. [COMMAND]

`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 are returned to the EDIT_PROFILE > prompt. [COMMAND]

`Command aborted; type "QUIT" to exit.`
You used Ctrl 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 Analyst because it indicates an EDIT_PROFILE programming error. [FATAL]

`*** Group group_name not legal for this project.`
When you tried to assign 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 databases. *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, divide 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 omitted a hyphen from an option name. The excess object is indicated on the following line by a caret (^). [COMMAND]

`Unrecognizable command` *command*.
Either you issued a command (while in Project Administrator mode) that is restricted to the System Administrator or you issued a command completely unknown to EDIT_PROFILE. [COMMAND]

`Unrecognizable option in command.`
You used either 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]

*** Error - lifetime out of range. Use -1 to 99000.
You used the -PASSWORD_LIFETIME option with an accompanying value outside this range, or you keyed an invalid character. Use ADD_USER with a password lifetime value within the proper range; -1 indicates Infinite, 0 indicates Default, and values 1 through 99,000 indicate the lifetime in days. [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 (PVF) 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 (UVF) 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 is allowed 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. Report it 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 database. 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 rebooted 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.
You specified the -ADD or -DELETE option as part of the new group list, but the list began with a group name. If you specify either -ADD or -DELETE, one of these options must be the first item in the list. [COMMAND]

*** Error - lifetime out of range. Use -1 to 99000.
You specified the -PASSWORD_LIFETIME option with a value outside this range, or you keyed in an invalid character. Use the option with a password lifetime value within valid range; -1 indicates Infinite, 0 indicates Default, and values 1 through 99,000 indicate the lifetime in days. [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 either 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, you must give the user an Initial Attach Point to log in to that project. [NOTICE]

User *user_id* updated *date/time*.
The command executed successfully. [NOTICE]

DEFAULT_PASSWORD_LIFETIME MESSAGES

*** Error - lifetime out of range. Use -1 (infinite) or 1 to 99000.
You keyed an invalid character, or you specified a default password lifetime outside this range. The value -1 sets a default password lifetime to infinite; values from 1 to 99,000 set a lifetime in days. Note that a value of 0 is invalid. Reissue DEFAULT_PASSWORD_LIFETIME followed by a value within the proper range. [COMMAND]

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 you were prompted for EPF attributes. [RETRY]

***EPF Attributes are not set
You have not defined the EPF attributes for either a user or a project. [NOTICE]

Invalid number of *resource-units*:
You specified a number that was either 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 1024. [RETRY]

`User attribute may not be null.`
You did not enter a value. After this message appears, the prompt is repeated. [RETRY]

`User attribute should be numeric.`
You cannot enter a non-numeric value 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 you 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 you specified either the -PROJECT or the -ALL option, 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 database 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 made, restore the files in question from their backup copies. The cause of this problem is often a serious physical disk or hardware error that you should report 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 PRIMOS was 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. If the error is repeated, you should report it 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.`
You may specify either a user ID or the -ALL option, but not both. [COMMAND]

`Warning: user` *user_id* `found on system(s):` *list*
The user ID you specified was found on at least one other system. All systems on which the ID was found are listed. [NOTICE]

C

DETAILED DESCRIPTION OF AUDIT RECORDS

Chapter 11 describes the Security Audit facility. As the first function within this facility, an Audit Collection facility gathers audit information on certain actions considered noteworthy within the system. These actions are called events. All events may be categorized under four general event types.

DESCRIPTION OF AUDIT RECORDS

Table 11-1 provides a summary of all events, categorized under event types. The following table, Table C-1, provides a detailed description of audit records for events. It specifies for each event:

- A brief event description
- The event number identifying that event
- The data items recorded for that particular event
- The abbreviated event type classification for that particular event

Numbers within the table are given in decimal, unless otherwise noted.

TABLE C-1. Detailed Description of Audit Records for Events

Event Description	*Event Number*	*Data Items Recorded*	*Event Type*
Set acccess category	3	Object on which ACL was to be set ACAT used	FS
Set default ACL	4	Object on which ACL was to be set	FS
Revert an ACL	5	Object on which ACL was to be reverted	FS
Set an ACL	6	Object on which ACL was to be set	FS
Calculate access rights	7	Object on which access was to be determined Rights requested	FS
Delete access category	8	Access category to be deleted	FS
Change open mode	9	Object on which open mode was to be changed	FS
Close file by pathname	10	Object to be closed	FS
Close file by entryname	11	Object to be closed	FS
Close file by unit	12	Unit number to be closed	FS
Change name	13	Old name of object New name	FS
Create directory	14	Directory to be created	FS
Delete file	15	File to be deleted	FS
Check existence of file	16	File to be checked	FS
Open file	17	File to be opened Unit number requested Unit number assigned	FS
Force write to disk	18	File to be written Unit number on which file is open	FS
Read directory password	19	Directory on which password is to be read	FS
Allocate record for ldev	20	Logical device number	FS
Read quota information	21	Directory	FS
Set quota field	22	Directory	FS
Delete ROAM file	23	File to be deleted	FS

TABLE C-1. Detailed Description of Audit Records for Events (continued)

Event Description	*Event Number*	*Data Items Recorded*	*Event Type*
Set file attributes	24	Object for which attributes are to be set	FS
Delete segment directory	25	Segment directory name Unit on which segdir is open	FS
Open segment directory	26	Entry number Unit number Key, as for SGD$OP	FS
Manipulate segment directory	27	Segment directory Key, as for SGDR$$	FS
Set directory password	28	Directory name	FS
Assign/unassign device	29	Device name Command (ASSIGN or UNASSIGN)	SYS
Assign/unassign magtape	30	Magnetic tape unit Command (ASSIGN or UNASSIGN)	SYS
Assign/unassign asynchronous line	31	Asynchronous line Command (ASSIGN or UNASSIGN)	SYS
Change password	32		SYS
Copy segment	33	Source octal segment number Destination octal segment number	SY
Check device access	34	Device name Octal device unit	SYS
Allocate segment	35	Octal segment number	SYS
Initialize user	36	User number Project name	SYS
Perform batch operation	37	Operation to be performed	SYS
Log out	39	Connect time (minutes) CPU time (seconds) I/O time (seconds)	SYS
Normal login line	41	Terminal line number	SYS
Phantom login	42	User type, as described for UTYPE$	SYS

TABLE C-1. Detailed Description of Audit Records for Events (continued)

Event Description	*Event Number*	*Data Items Recorded*	*Event Type*
Restore static mode program	43	Program to be restored Key, as for REST$$ Unit on which program was opened Segment to be restored to	SYS
Return segment access rights from Ring 3	44	Octal segment number Ring 3 access, as for RSEGAC$	SYS
Return segment/delete EPF	45	Octal segment number	SYS
Set segment access	46	Octal segment number Access, as for RSEGAC$	SYS
Perform spooler operation	47	Operation	SYS
Initialize VMFA or segment/map EPF	48	Unit on which EPF is open Number of segments to be mapped	SYS
Set information for asynchronous line	93	Line number	SYS
Add disk	49	Physical device number in octal	PRIV
Configure asynchronous line	50	Line number (octal)	PRIV
Perform CHAP command	56	Operation or user number New priority New timeslice	PRIV
Change System Administrator	57	New administrator user ID	PRIV
Perform old-style command	58	Command	PRIV
Perform EDIT_PROFILE command	59	Command	PRIV
Shut down Login Server	62		PRIV
Log out user	64	User ID Key: -1 Log out all (supervisor terminal only) 0 Log out self 1 Log out *nn* 2 Log out user ID (supervisor terminal only)	PRIV

TABLE C-1. Detailed Description of Audit Records for Events (continued)

Event Description	*Event Number*	*Data Items Recorded*	*Event Type*
Perform MAXUSR command	65	Number of users	PRIV
Delete priority ACL	66	Partition on which priority ACL was to be deleted	PRIV
Set priority ACL	67	Partition on which priority ACL was to be set	PRIV
Check for privileged process	68	Operation or source of check	PRIV
Set scheduler variables	69	Command (MAXSCH or ELIGTS) Value with the command	PRIV
Change number of AUDITOR buffers	70	New number of buffers	PRIV
Specify audit file	71	New audit file	PRIV
Turn on/off audited events or event types	72	Option (-ON or -OFF) Events to be operated on Event types to be operated on	PRIV
Enter event from Ring 3 into audit trail	75	Operation or event	PRIV
Shut down AUDITOR	76	Number of buffers written	PRIV
Start up AUDITOR	77		PRIV
Enable/disable auditing of users	78	Option (-ON or -OFF), for SECURITY_MONITOR command List of user IDs	PRIV
Request Security AUDITOR status	79	Active options for present auditing session	PRIV
Share a segment	83	Segment number in octal Access, as for the SHARE command	PRIV
Start up Login Server	84		PRIV
Perform USRASR command	85	Target user for supervisor terminal	PRIV
Attach to directory	86	Directory to attach to	ATCH
Attach/scan directory	87	Directory that is top-level directory for the scan	ATCH

TABLE C-1. Detailed Description of Audit Records for Events (continued)

Event Description	*Event Number*	*Data Items Recorded*	*Event Type*
Attach home	88		ATCH
Attach by logical device number	89	Logical device number Master file directory	ATCH
Perform relative attach	90	Subdirectory to be attached to	ATCH
Perform SECURITY_MONITOR command	91	Command	PRIV
Record Auditor event	92	Event description	PRIV
Switch audit files	96	Old audit file pathname New audit file pathname	PRIV
Close audit file	97	Audit file pathname Total number of buffers written to audit file	PRIV
Perform numbered semaphore operation	98	Semaphore number Description of operation	SYS
Get information about open units	99	User number for owner of open units Unit number (may be input or output, depending on key) Key indicating the information being sought: 1 For specified unit 2 For current attach point 3 For home attach point 4 For initial attach point -1 For next open file unit or for next open file whose pathname contains the prefix specified in pathname below 6 For como unit Pathname for unit (may be input or output)	PRIV

FORMAT OF A SECURITY AUDIT FILE

The audit file consists of a version stamp followed by audit event records. A record is of variable length, depending on the number of data items it carries. An audit file has the general layout shown in Figure C-1.

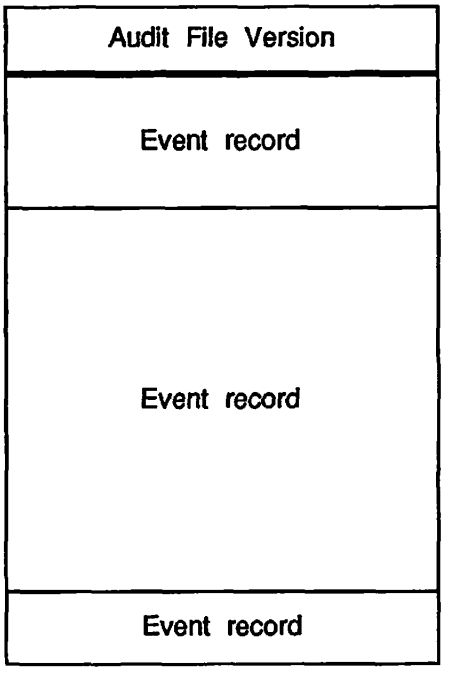

FIGURE C-1. Audit File Layout

Note that event records of varying size are shown. The Audit Reporting facility recognizes the end of an audit file by reading an end-of-file marker. The audit file has only one such marker.

Data Structures for Audit File Format

The following data structures describe the audit file format.

```
/* Declaration of file header--both tape and disk. */
dcl 1 file_header based,
      2 version              fixed bin(15);      /* version of structures in this file.  */
                                                 /* current version = 1.                 */

/* Declaration of version 1 event structure. */
dcl 1 event_v1 based,
      2 timestamp            fixed bin(31),      /* date/time stamp in binary format.    */
      2 userid               char(32) var,       /* userid that generated this event.    */
      2 process_num          fixed bin(15),      /* process # that generated this event. */
      2 optional_elements    fixed bin(15),      /* number of optional elements          */
                                                 /* (including obj and aux)              */
      2 group                fixed bin(15),      /* event group.                         */
      2 number               fixed bin(15),      /* event number.                        */
      2 type                 fixed bin(15),      /* event type.                          */
      2 error_status         fixed bin(15),      /* error status at time of probe call.  */
      2 object,                                  /* object data.                         */
        3 len                fixed bin(15),      /* length of data field in 16-bit words */
                                                 /* or characters if a char data type.   */
        3 type               fixed bin(15),      /* data type.                           */
        3 data               fixed bin(15);      /* actual type can be other than        */
                                                 /* fixed bin(15). That is used as a     */
                                                 /* placeholder only.                    */
   /* auxiliary data comes immediately after object data, but the actual                 */
   /* relative location is undetermined until runtime--it depends on                     */
   /* the length of object.data.                                                         */

/* Based structure (version 1) for any auxiliary data present.                          */

dcl 1 aux_v1 based,
      2 len                  fixed bin(15),      /* length of data field in 16-bit words */
                                                 /* or characters if a char data type.   */
      2 type                 fixed bin(15),      /* type of data field.                  */
      2 data                 fixed bin(15);      /* actual type can be other than        */
                                                 /* fixed bin(15). That is used as a     */
                                                 /* placeholder only.                    */

/* Events */

%replace File_System         by 1;
%replace SYStem              by 2;
%replace PRIV_ops            by 3;
%replace ATtaCH              by 4;
```

```
/* Event Types */

%replace SUCCess        by 1;        /* Error code from PRIMOS gate = 0.              */
%replace NO_ACCess      by 2;        /* Error code from PRIMOS gate = E$NRIT.         */
%replace FAILure        by 3;        /* Error code from PRIMOS gate is non-zero       */
                                     /*   and not E$NRIT.                             */


/* Data types for SEC_PROB obj and aux data items. */

%replace K$FB15         by 1;        /* one word integer.                             */
%replace K$FB31         by 2;        /* two word integer.                             */
%replace K$NVCHAR       by 3;        /* non-varying character string.                 */
%replace K$VCHAR        by 4;        /* varying character string.                     */


/* Format of a varying character string. */
dcl 1 varstring,
      2 length          fixed bin(15), /* length of string in characters.             */
      2 characters char(*);            /* characters in the string.                   */
```

Content of Audit Records

The data structure *event_v1* shows the present version for an audit record of any event. Most of its substructures (*timestamp*, *userid*, *process_num*, and so forth) describe items of information that are displayed for every audit trail. Certain other substructures give the length of a particular audit trail. The length of the audit trail varies with the number of data items needed to record that particular event. (See Table C-1.)

Timestamp: The timestamp can be converted to readable format by using the routine CV$FDA.

Length of Audit Records

The length of an audit record for any event is the sum of the length of the nonvarying part of the record plus the length of the varying part. The length of the varying part is determined by the data, if any. The value of *optional_elements* tells how many data elements there are, if any. The definitive length of an item of data is derived from the data type and, if the data is a varying string, the length as specified in the first word of the string. By determining the length of the object data (*object*) and its optional auxiliary data (*aux_v1*), the Audit Reporting facility identifies the beginning of the audit record for the next event (*event_v1*).

D

NUMBERED SEMAPHORE ACLS

The *Department of Defense Trusted Computer System Evaluation Criteria* specifies in one of its C2 requirements that *all* named system resources must have access control. Formerly, numbered semaphores in Prime systems were a named system resource that did not have access control. To meet the DoD criteria PRIMOS was modified to include ACLs on numbered semaphores in a manner similar to Device ACLs. (Named semaphores already have ACLs, so no modification was necessary in that area.)

Caution

Administrators who are not interested in maintaining the strict C2 configuration should not use Numbered Semaphore ACLs, because in non-C2 environments they are of little use and the performance penalty for semaphore operations is not justified. Do not issue the command NUMSEMACL -ON if your system does not require a strict C2 configuration.

When the file C2.INIT.COMI is executed, numbered semaphore ACLs will be enabled. On a C2-secure system, never issue the command NUMSEMACL -OFF (NSACL -OFF) while users are on the system or allowed to log in.

On a system maintaining strict C2 security, there is a top-level directory on the command device called NUMSEM*. This directory must be present. Within NUMSEM*, there is a directory called DEFAULT. The System Administrator may also choose to create directories called SEM1, SEM2, SEM3, ... SEM*n*, where *n* has a maximum of 64. When a numbered semaphore is operated on, the access to the corresponding directory in NUMSEM* is checked. If the user has at least U (Use) rights, the operation proceeds. If the user does not have U rights, the operation fails with an error code of insufficient access. If there is no SEM*n* directory, then access to the directory DEFAULT is used. Thus, the DEFAULT directory must always be present. The System Administrator should determine which users are allowed to operate on numbered semaphores and set appropriate access on the directories.

All users need U rights to NUMSEM* itself in order to successfully check their access to its subdirectories. If they do not have access to NUMSEM*, users will receive an insufficient access error even though they might have access through a specific ACL to a subdirectory.

INDEX

INDEX

Symbols

A

B

C

D

M

N

O

P

Q

T

U

V

W

X

SURVEYS

READER RESPONSE FORM

System Administrator's Guide, Volume III
DOC10133-2LA

Your feedback will help us continue to improve the quality, accuracy, and organization of our publications.

1. How do you rate this document for overall usefulness?

   ☐ *excellent* ☐ *very good* ☐ *good* ☐ *fair* ☐ *poor*

2. What features of this manual did you find most useful?

   _____
   _____
   _____
   _____
   _____
   _____

3. What faults or errors in this manual gave you problems?

   _____
   _____
   _____
   _____
   _____
   _____

4. How does this manual compare to equivalent manuals produced by other computer companies?

   ☐ *Much better* ☐ *Slightly better* ☐ *About the same*
   ☐ *Much worse* ☐ *Slightly worse* ☐ *Can't judge*

5. Which other companies' manuals have you read?

   _____
   _____

Name:_____ Position:_____

Company:_____

Address:_____

_____

_____ Postal Code:_____

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

System Administrator's Guide, Volume III
DOC10133-2LA

Your feedback will help us continue to improve the quality, accuracy, and organization of our publications.

1. How do you rate this document for overall usefulness?

   ☐ *excellent* ☐ *very good* ☐ *good* ☐ *fair* ☐ *poor*

2. What features of this manual did you find most useful?

   _____
   _____
   _____
   _____
   _____
   _____

3. What faults or errors in this manual gave you problems?

   _____
   _____
   _____
   _____
   _____
   _____

4. How does this manual compare to equivalent manuals produced by other computer companies?

   ☐ *Much better* ☐ *Slightly better* ☐ *About the same*
   ☐ *Much worse* ☐ *Slightly worse* ☐ *Can't judge*

5. Which other companies' manuals have you read?

   _____
   _____

Name:_____ Position:_____

Company:_____

Address:_____

_____

_____ Postal Code:_____

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

System Administrator's Guide, Volume III
DOC10133-2LA

Your feedback will help us continue to improve the quality, accuracy, and organization of our publications.

1. How do you rate this document for overall usefulness?

   ☐ *excellent* ☐ *very good* ☐ *good* ☐ *fair* ☐ *poor*

2. What features of this manual did you find most useful?

   _____
   _____
   _____
   _____
   _____
   _____

3. What faults or errors in this manual gave you problems?

   _____
   _____
   _____
   _____
   _____
   _____

4. How does this manual compare to equivalent manuals produced by other computer companies?

   ☐ *Much better* ☐ *Slightly better* ☐ *About the same*
   ☐ *Much worse* ☐ *Slightly worse* ☐ *Can't judge*

5. Which other companies' manuals have you read?

   _____
   _____

Name:_____ Position:_____

Company:_____

Address:_____

_____

_____ Postal Code:_____

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

System Administrator's Guide, Volume III
DOC10133-2LA

Your feedback will help us continue to improve the quality, accuracy, and organization of our publications.

1. How do you rate this document for overall usefulness?

   ☐ *excellent* ☐ *very good* ☐ *good* ☐ *fair* ☐ *poor*

2. What features of this manual did you find most useful?

   _____
   _____
   _____
   _____
   _____
   _____

3. What faults or errors in this manual gave you problems?

   _____
   _____
   _____
   _____
   _____
   _____

4. How does this manual compare to equivalent manuals produced by other computer companies?

   ☐ *Much better* ☐ *Slightly better* ☐ *About the same*
   ☐ *Much worse* ☐ *Slightly worse* ☐ *Can't judge*

5. Which other companies' manuals have you read?

   _____
   _____

Name:_____ Position:_____

Company:_____

Address:_____

_____

_____ Postal Code:_____

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