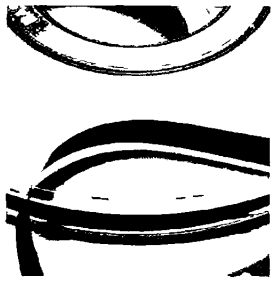
A3.

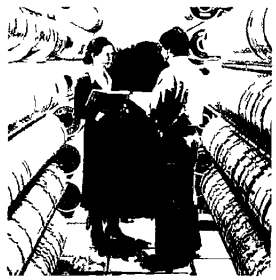
[redacted]

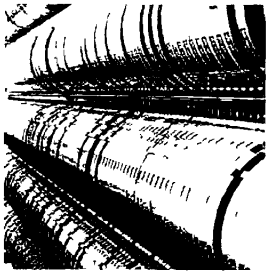
Prime Computer, Inc.

DOC5037-190P
System Administrator's Guide
Revision 19.0

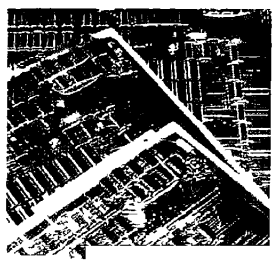













System Administrator's Guide

DOC5037-190

First Edition

by

System Administrator's Guide Task Force:

Alice Landy, Anthony R. Lewis, Emily M. Stone, Sarah Lamb, A. Paul Cioto, Anne W. Patenaude, Michael J. Karp, Stephen R. Marra

This guide documents the software operation of the Prime Computer and its supporting systems and utilities as implemented at Master Disk Revision Level 19.0 (Rev. 19).

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

COPYRIGHT INFORMATION

HOW TO ORDER TECHNICAL DOCUMENTS

U.S. Customers

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

Prime Employees

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

Customers Outside U.S.

Contact your local Prime subsidiary or distributor.

PRIME INFORMATION

Contact your Prime INFORMATION dealer.

PRINTING HISTORY — SYSTEM ADMINISTRATOR'S GUIDE

Edition	Date	Number	Documents Rev.
First Edition	July 1982	DOC5037-190	19.0

Changes made to the text since the last printing have been indicated with change bars in the margin. Change bars with numbers indicate technical changes. Those without numbers indicate rewrites for clarification or additional information. Change bars are used in Chapters 3, 7, 8, and 16 which have been changed from earlier versions. All others chapters and appendixes are either new or completely rewritten.

SUGGESTION BOX

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

Alice Landy
Technical Publications Department
Prime Computer, Inc.
500 Old Connecticut Path
Framingham, Massachusetts 01701

Contents

16 ADDING AND MODIFYING SYSTEM SOFTWARE

PART III - NETWORKS

17 PRIMENET OVERVIEW

18 ADMINISTERING PRIMENET

APPENDIXES

A PHYSICAL DEVICE NUMBERS

About This Book

This book is a guide to the administration of your Prime computer system. It is intended to help you make the decisions and set up the procedures that will:

- Provide your users with a smoothly functioning system.
- Enable your operators to deal with the day-to-day running of the system.
- Help users and operators deal with unexpected problems.

You are expected to have some familiarity with Prime systems before reading this book. If you are not familiar with the system, you should read the Prime User's Guide (DOC4130-190) before reading this book. The Prime User's Guide explains Prime's file management system, and provides introductory and tutorial information about essential commands and utilities.

You will also need:

- System Operator's Guide, (DOC5038-190). This book explains the day-to-day workings of the system and contains detailed information on operations procedures and commands.
- Rev. 19.0 Planning and Installation Guide, (DOC6426-190). This book provides a detailed reference on how to install your Rev. 19.0 software, whether you are a new user or are converting your system from Rev. 18.
- PRIMOS Commands Reference Guide, (FDR3108-190). This book provides a detailed reference for user commands.

WHAT IS SYSTEM ADMINISTRATION?

System Administration is, basically, the organization and management of computer systems. Planning is particularly important for the Administrator, since good planning makes day-to-day operations run more smoothly.

The person responsible for system administration is called the System Administrator. In general, the System Administrator is the person to whom users and operators come when anything goes wrong, or when something happens unexpectedly. The System Administrator is also responsible for the security of the system.

Although reference is frequently made in this book to the System Administrator, in your installation there may be no single person who is responsible for system administration. The job may be shared, or the System Administrator may also double as an operator or a user. It is not necessary that there be a single, specific System Administrator. If you have any share in administrative duties, this book is probably for you.

HOW TO USE THIS BOOK

This book contains three major parts and several appendixes. The parts are:

- Part I — Creating Your System
- Part II — Maintaining Your System
- Part III — Networks

Part I, CREATING YOUR SYSTEM, contains information on planning for your system: planning for user registration, system configuration, disk partitioning, and the allocation of hardware and software resources.

Part II, MAINTAINING YOUR SYSTEM, contains information on planning for and carrying out the tasks that keep a system running smoothly: the care of the computer room, system monitoring, backups, and tending to users' needs. It also contains information on how to add and modify software for your system.

Part III, NETWORKS, is of interest to those administrators whose computers form part of a network. It discusses network planning and configuration.

The History of This Book

Prior to Rev. 19, the book called the System Administrator's Guide (PDR3109) dealt both with system administration issues and with system operations issues. At Rev. 19.0, two books exist. The System Administrator's Guide (DOC5037-190) provides information aimed at the needs and interests of the administrator, while the System Operator's Guide (DOC5038-190) provides detailed reference for operations matters. Both books are essential for any installation.

PRIME DOCUMENTATION CONVENTIONS

The following conventions are used in command formats, statement formats, and in examples throughout this document. Examples illustrate the uses of these commands and statements in typical applications. Terminal input may be entered in either uppercase or lowercase.

Convention	Explanation	Example
UPPERCASE	In command formats, words in uppercase indicate the actual names of commands, statements, and keywords. They can be entered in either uppercase or lowercase.	SLIST
lowercase	In command formats, words in lowercase indicate items for which the user must substitute a suitable value.	LOGIN user-id
abbreviations	If a command or statement has an abbreviation, it is indicated by underlining. In cases where the command or directive itself contains an underscore, the abbreviation is shown below the full name, and the name and abbreviation are placed within braces.	LOGOUT {SET_QUOTA / SQ}
underlining in examples	In examples, user input is underlined but system prompts and output are not.	OK, RESUME MY_PROG This is the output of MY_PROG.CPL OK,

Brackets [ ]	Brackets enclose a list of two or more optional items. Choose none, one, or more of these items.	SPOOL [-LIST -CANCEL]
Braces { }	Braces enclose a list of items. Choose one and only one of these items.	CLOSE {filename ALL}
Ellipsis ...	An ellipsis indicates that the preceding item may be repeated.	item-x{,item-y}...
Parentheses ( )	In command or statement formats, parentheses must be entered exactly as shown.	DIM array (row,col)
Hyphen -	Wherever a hyphen appears as the first letter of an option, it is a required part of that option.	SPOOL -LIST

PART I
Creating Your System

1
Planning For Your System

OVERVIEW

If you are acting as System Administrator for your system, you will want to do some planning before you actually install Rev. 19. Rev. 19 offers many new features in the area of system administration, and you will want to take advantage of them. In particular:

- You will want to decide what aspects of the new User Profiles system you want to use; and you will want to plan your user data base. This chapter will introduce you to User Profiles, and show you how to plan for them. Further details will be found in Chapter 4, EDIT_PROFILE, and Chapter 15, SECURITY.

- If you are a new user, you will probably want your system to be protected by Access Control Lists (or ACLs) from the start. But if you are already running Rev. 18, you may want to phase in the use of ACLs. In either case, you will want to plan your strategy now, so that you can carry it out in an orderly manner.

  This chapter will introduce you to ACLs; later chapters, and the Prime User's Guide, will provide fuller discussions.

- If you are a new user, you will want to plan your system configuration. If you are already running Rev. 18, you will want to know what new features Rev. 19 provides in this area.

This chapter will provide guidelines on configuration for the new user, plus some notes on new features for current users. Full details are given in Chapter 3, CONFIGURATION DIRECTIVES.

- If you are a new user, you may want to think about how you will divide your disk space among your various users and system needs. You will find an introduction and guidelines for this task in Chapter 6, DISKS.

- You may also want to consider the use of quotas on file system directories. If you use quotas, you can limit the amount of disk space each top-level directory can use. If you do not use quotas, you allow users to compete for space on a first-come, first-served basis. Explanations of quotas may be found in Chapter 6 and in the Prime User's Guide.

- Finally, if you are currently running Rev. 18, you will want to know about the new features for backups at Rev. 19. These include an enhanced ability to do backups under PRIMOS; new badspot handling provided by COPY_DISK and by PHYSAV and PHYRST; and a new file maintenance utility, FIX_DISK. Some guidelines for backups are provided in Chapter 13, BACKUPS. Full details on backups and on FIX_DISK are given in the System Operator's Guide.

This chapter will concentrate on your two main areas of planning: user profiles and system configuration. Some User Profile data base must be in place before any users can log in at Rev. 19; and some configuration file must exist before your system can be brought up.

USER PROFILES

Why Use User Profiles?

User profiles:

- Provide a secure method of identifying and validating users.

- Provide administrative control over users.

- Interface with the Access Control List mechanism for file system protection.

- Allow the grouping of users with similar characteristics for accounting and file system purposes.

- Allow the creation of a unique environment for each user.

At login time, each user is:

- Identified by a user-id.
- Validated by the use of a login password.
- Attached to the appropriate directory (known as the user's origin directory, or Initial Attach Point).
- Defined as a member of a particular project.
- Defined as a member of up to 32 specified ACL groups.

A site-specific external login program may then be run, if desired, to further define the user to the system (by requiring extra validation), or to run a site-specific accounting package.

Finally, the user's own login program may be run, to create a customized environment by performing such tasks as:

- Setting terminal characteristics.
- Setting erase and kill characters.
- Modifying system prompts.
- Activating abbreviation and/or global variable files.

Note that the first of these programs is always run, automatically by PRIMOS. The choice of whether or not the second (external login) program is run is up to the System Administrator. And the choice of whether or not the third (the individual login program) is run is up to each user. This combination provides flexibility and security simultaneously.

How Are User Profiles Handled?

The System Administrator, or someone designated by the System Administrator, plans the user profile data base (as explained in this chapter). The administrator then creates the data base using the EDIT_PROFILE utility, as explained in Chapters 2 and 4. This data base contains names (user-ids) and attibutes for each user. The sum of a user's attributes constitute his "profile". Users must be registered in this data base before they can log into the system. Therefore, EDIT_PROFILE allows the administrator to keep the data base up to date.

The data base also contains profiles for one or more projects. A project is composed of a subset of users who share certain characteristics. Usually, they are grouped together because they share common attributes or are accounted for together.

From the system's viewpoint, projects are required: each system must have at least one. From your viewpoint, projects are optional. If you are not interested in setting up projects on your system, you simply:

- Allow EDIT_PROFILE to create a default project for you. (The project will be named DEFAULT.)
- Allow DEFAULT to remain the only project on the system. As long as DEFAULT is your only project:
  - — EDIT_PROFILE automatically registers all users as members of project DEFAULT when you add them to the system.
  - — EDIT_PROFILE asks you no questions relating to projects.
  - — Users never have to specify project-ids at login.

If you do want to use projects on your system, you can let them further customize users' environments. Since users can be members of more than one project, their attributes and environment can vary according to which project-id they supply at login time. Thus, users who need different directories, different access, and so on, for different jobs can receive them automatically at login time by logging in as members of one or another 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. This allows the responsibilities -- and the workload — of administration to be shared in an orderly manner among a number of people.

What Are ACLs?

Since ACLs will be mentioned frequently in the discussion of attributes, we'd better define them now.

An ACL (or Access Control List) is a list of users (or of groups of users), together with a list of access rights belonging to each user or group. An ACL can protect a file, a group of files, a directory, or an entire subtree. The rights that can be granted are shown in Table 1-1. A sample ACL might be:

```
JAMES: LURA
JOAN: ALL
JOHN: NONE
$REST: LUR
```

If this ACL protected directory J, then JOAN would have ALL access to J. JOHN would have no access to J whatever. JAMES could use the directory in pathnames, list its contents, read its files, and add new files. All other users of the system ($REST) could use the directory

in pathnames, list its contents, and read its files. They would have no further rights to it.

Table 1-1
ACL Access Rights

Symbol	Right	Applies To	Meaning
R	Read	Files	File may be read.
W	Write	Files	File may be modified.
U	Use	Directories	User may attach to directories.
L	List	Directories	Directory contents may be listed.
A	Add	Directories	Directory entries may be added.
D	Delete	Directories	Directory entries may be deleted.
P	Protect	Directories	Access rights may be changed.
ALL		Files and Directories	All of the above rights.
NONE		Files and Directories	No access allowed.

What Are 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 dot. Thus, it is easy, when reading an ACL, to tell which IDs represent individual users and which represent groups.

There are two kinds of ACL groups: system-based and project-based. A system-based ACL forms part of the user's entry in the system data base. It is active every time the user logs in, no matter what project the user logs into. System-based ACL groups are often user for global system access. For example, ".SUPER_USER" might have ALL access to system UFDs.

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

Often, a project-id will have a corresponding ACL group which contains all members of the project. For example, the project OPERATIONS might use an ACL group .OPERATIONS for its members. In addition, project-based ACL groups may be used to distinguish among project members terms of the rights each group within the project needs.

In a given ACL, individual rights override group rights. Thus, if JOHN were a member of group .JARS, and the ACL on a directory read:

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

JOHN would have only LUR rights to the directory. Group rights, however, are additive. For example, assume the following ACL protects a directory:

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

Any user who is both a leader and a member is given PDALURW (that is, all) access.

How Is an ACL Group Defined?

The System Administrator enters the name of the ACL group in the system database. The System or Project Administrators then define various users as members of the group. Groups and their memberships can be altered as needed. Thus the data base can be kept up to date, to reflect the current needs of the system.

Chapter 16 of the Prime User's Guide contains a good introduction to ACLs. If you are intending to use ACLs on your system, you should read this chapter before you go further in your planning.

Should You Use ACLs?

You will probably want to use ACLs to provide at least part of the file system security on your system. ACLs allow you to:

- Improve file system security.
- Provide an easy to use interface for users and programs to set and modify file system access.

- Provide common access for specified groups of users under administrative control.

If you do not use ACLs:

- You will have very little security on your User Profile data base.
- You will not be able to use projects on your system.
- You may have lessened security on other subsystems.

If you are converting your system from Rev. 18, you may not want to convert your entire system to ACLs immediately. But you will almost certainly want to make some use of ACLs as soon as possible; and you may very well want to convert your entire system eventually. (For a comparison of the security provided by ACLs and passwords, see Chapter 15.)

Note

If your system is part of a network which contains both Rev. 18 systems and Rev. 19 systems, you may want to keep the directories most frequently accessed by remote users from the Rev. 18 systems password protected. Chapters 17 and 18 explain network security in more detail.

What Attributes Can You Specify for Users?

For each user, you specify:

- A user-id
- A login password
- A default project affiliation (optional)
- Membership in up to 16 system-wide ACL groups (optional)

These are system attributes. They are valid every time the user logs in. They are stored as part of the system data base.

In addition, you specify another set of attributes for each project of which the user is a member. This second list contains:

- An Initial Attach Point, or origin directory. (This is the directory to which the user is attached at login.
- Membership in up to 16 project-specific ACL groups (optional).

Note

You don't have to define these attributes for each member of the project. You can create a default Initial Attach Point and default ACL groups for each project. Any user for whom you don't define specific project attributes then uses the default attributes.

Project attributes are valid only when the user logs in as a member of that particular project. There are three ways in which a user can log in as a project member:

- The user can supply the project-id as part of the login command.
- If the user supplies no project-id at login, and a default project has been defined for him as one of his system attributes, then the user is logged in as a member of that default project.
- If the only project in use on the system is project DEFAULT, all users are automatically logged in as members of that project.

(For further details on how projects are handled at login time, see Chapter 15.)

EXAMPLES OF SYSTEM PLANNING

Let's look at three imaginary systems, to see how profiles, projects, and ACLs might work for them:

Example 1:

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

1. The System Administrator looks at the disk. There are about 30,000 records for the teams to share.
2. The SA creates three UFDs, naming them A, B, and C. He sets a quota of 10,000 records on each UFD. That establishes the space for each team.
3. The SA now creates his data base for the three teams. Using EDIT_PROFILE, he creates three projects: ALPHA, BETA, and GAMMA.

4. The SA registers all members of TEAM A as users of the system, and as members of project ALPHA. Their Initial Attach Point is UFD A. (IF UFD A had subdirectories, any of its subdirectories could serve as Initial Attach Points for team members.)

   The SA also creates a Project Administrator, AMY, for project ALPHA.

   Finally, the SA sets up three access groups for project ALPHA to use. The first group, .TEAMA, contains all the project members. The other two, .SOMEA and .OTHERA, are left empty for the Project Administrator to use as she wishes. With these, she can limit access within the project's directories to particular subgroups of project members.

5. The SA registers the members of projects BETA and GAMMA in the same way that he registered members of Project ALPHA.

6. The SA now sets access protection on the three top-level UFDs, A, B, and C. He does this by creating Access Control Lists, or ACLs.

   The ACL for UFD A looks like this:

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

   This ACL gives Project Administrator AMY all rights to her project's UFD, including the right to set protection on any subdirectories she may create. All other project members 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.) No one else has any rights; they cannot attach to UFD A or gain any information about its contents.

   ACLs for UFDs B and C are similar.

7. The SA keeps full control of the MFD. Its ACL reads:

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

   This allows the system's users to attach to the disk, but does not let them list or read the contents of the MFD. It also denies the Project Administrators the right to change access to their top-level UFDs. (To do that, they would need "LU" — List and Use -- access to the MFD.)

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

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

Similarly, all members of team A can work at will within their own directory and its 10,000 records. But the other two directories are invisible to them. Team A members can't attach to directories B or C;, they can't list or read any information from them; they can't copy information in or out of them. They are completely isolated from the other directories by the ACLs which have been established.

Note that ACLs and projects last only as long as you want them to. Suppose that Teams A, B, and C suddenly had to cooperate on a new, large project. The System Administrator could set up a new project, DELTA. Users then could belong to two projects: ALPHA and DELTA, perhaps, or BETA and DELTA. Users would specify which project they wanted to work on by including the project-id in the login command, as in:

```
LOGIN ALAN -PROJECT DELTA
```

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

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

This method of proceeding would be especially appropriate if any of the following applied:

- The three older projects were still going on.
- There was additional disk space for project DELTA to occupy.
- Accounting wanted to keep the four projects separate.

Example 2:

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

This group uses the simplest possible system. They have one "default" project (automatically named DEFAULT) to which everyone in the group belongs. No access groups are defined. The ACLs on their MFDs say simply:

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

ACLs on top-level UFDs read:

```
$REST:ALL
```

If anyone needs special protection on a particular directory, he sets it himself.

One person would be known to the system as System Administrator. But there would be nothing to prevent other members of the group from using the System Administrator's user-id and doing administrative tasks.

Example 3:

If the group described in Example 2 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 would add one access group, .US, to the system. He would register all the system's users as members of that group.

2. The SA would then change the ACLs on the system's MFDs to read:

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

   The new ACL would not restrict the rights of the original users, since they are all members of the group .US. Users of other systems would have restricted privileges. They could attach to directories on these disks, 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 4:

The math department at a small college has bought a 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 so on. The department head will act as SA.

1. On this system, the SA sets up a default project for faculty members, graduate students working on research products, and whatever "guests" may visit the system. He also sets up projects for each of the six math courses whose students will

use the computer. As research projects are defined, he may set up projects for them as well.

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

2. The SA sets up one system-wide access group, .FACULTY. He places all faculty members in the group. He defines project-based access groups for each math course: .M105, .M210, etc. For the graduate courses, he defines a few other access groups that may be used for joint projects.

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

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

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

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

   The faculty members then create (and protect) subdirectories as they need them. In particular, they 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 his students to work cooperatively on projects. He wants his course directory ACL to read:

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

   Professor White wants no sharing of information to take place among students in his course. He wants his course directory's ACL to read:

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

Professor White will then create an individual directory for each student to work in. He will set an ACL on each directory that says:

(student-id): DALURW

In this way, the students will not be able to see each other's work. However, they will be able to read the messages Prof. White places in the course directory. They will also be able to place messages there themselves.

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

   Because their Initial Attach Point must be controlled by their project affiliations (and because one student may be enrolled in more than one course), students will have to specify project-ids when they log in. For example:

   LOGIN J2943 -PROJECT M105

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

   Note that the students can remain in the system data base until they graduate. While they are enrolled in courses, their project affiliation and their presence in access groups allows them to work on the system. At other times, they will have either very limited access or no access, depending on whether or not the SA has set the system to require a valid project-id for login.

THE USER PROFILE DATA BASE

What does a User Profile data base actually look like? From the system's point of view, it's a directory called the SAD (the System Administration Directory) that resides in the MFD of the system's command device. (For information on the SAD and its structure, see Chapter 15, SECURITY.) From your point of view, it's a collection of lists resembling the illustration in Figure 1-1.

The collection contains two "master lists": a list of every project name you define for your system, and a list of every ACL group name you define for your system. (EDIT_PROFILE uses these lists to keep track of which names are valid and which are not.)

Note

If you're not using ACLs or projects on your system, these two lists will not be part of your data base.

Next comes the system data base. It contains an entry for every user you define to the system, beginning with the System Administrator. The entry lists the user's attributes: user-id, password, and (optionally) default project and ACL groups.

Finally, there are the project data bases. There is a separate project data base for each project you define. The project data base contains four types of material:

- If you are using ACLs, it contains a list of all the ACL group names that you have designated for this project. (This list, called the "limits" of the project, provides a pool of project-specific group names which the Project Administrator can assign as he or she chooses.)

- It contains the name (that is, the user-id) of the Project Administrator. (The Project Administrator does not have to be a member of the project.)

- It contains a record of the project default Initial Attach Point (or IAP) and the project default ACL groups. If these are present, then users who are assigned no specific IAP or ACL groups use the project's defaults, instead.

  Note

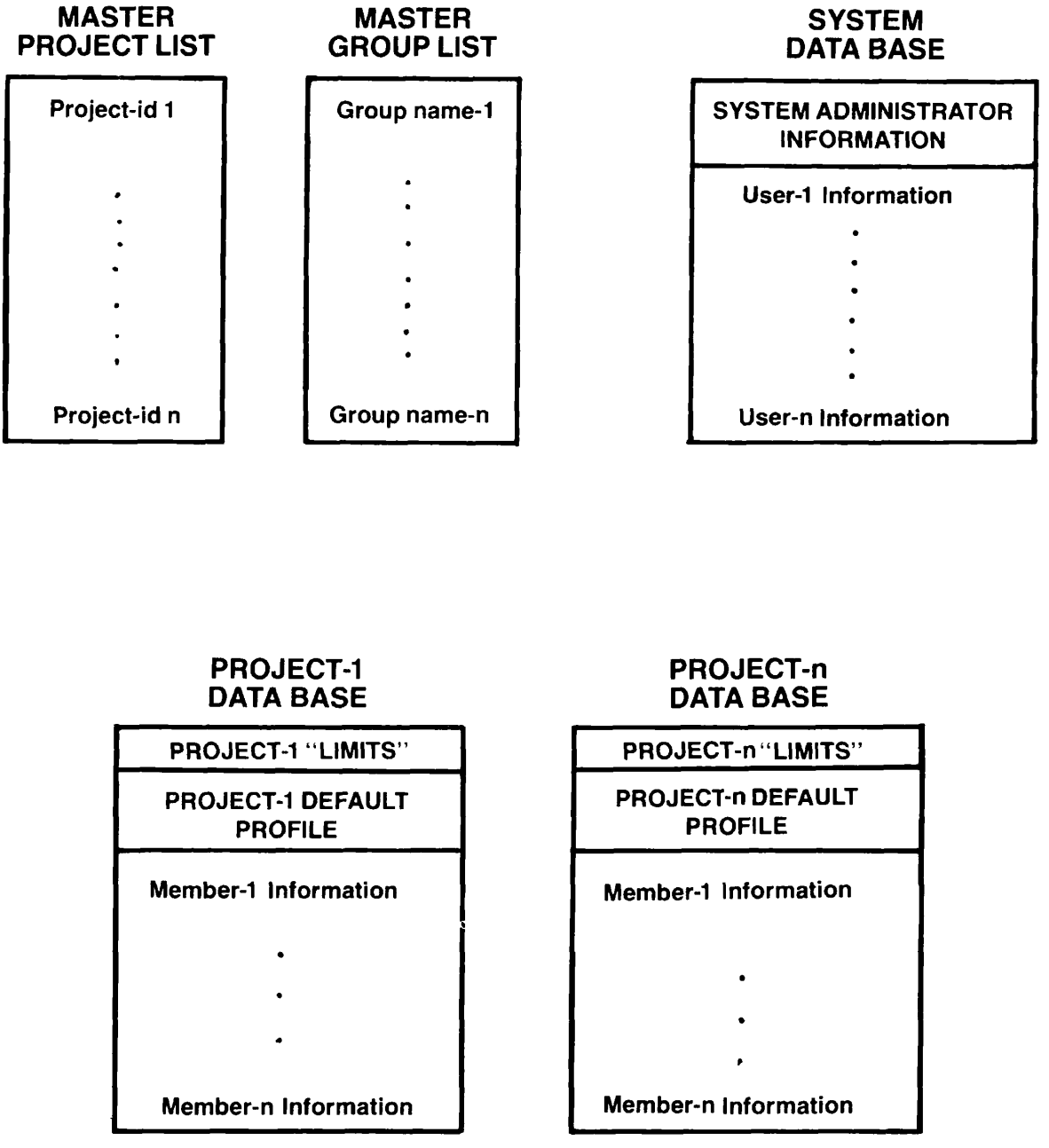
  If there is no default IAP, you must assign each user an IAP, or the user will not be able to log in.

- It contains an entry for each user who is a member of the project. The entry contains that user's project attributes: Initial Attach Point and project-specific ACL groups.

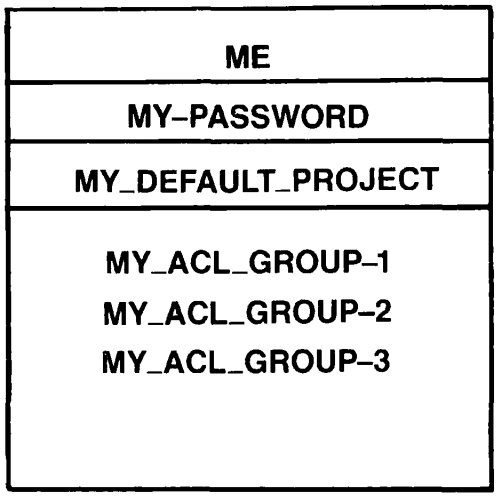
From the user's point of view, the data base resembles that sketched in Figure 1-2: a collection of the user's attributes, one set for use at the system level during every session, and one or more project-specific sets, to be used when logged in as a member of that project.

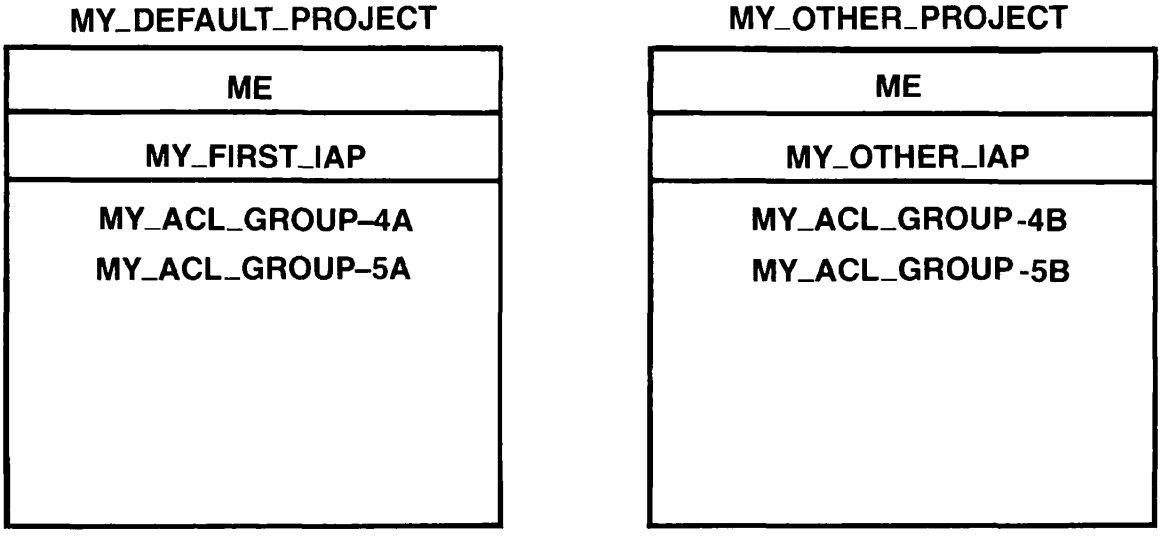
Project Data Bases

You may have noticed in the list above that system attributes are always assigned to each user on an individual basis, but that project attributes may be assigned to many project members "by default". It is by this means that projects allow you to combine the security of individual IDs and passwords with the convenience of group access to the file system. As an example of this, consider the imaginary project data base portrayed in Figure 1-3.



User Profile Data Base from Administrator's Viewpoint
Figure 1-1





User Profile Data Base from User's Viewpoint
Figure 1-2

Project Administrator: CLAUDIUS
Project Limits — ACL Groups: .DANES .PRINCES .OTHER .SPECIAL .CHARACTERS
Project Defaults: Default IAP: <DRAMA>DENMARK>ELSINORE Default ACL Groups: .DANES
ID: CLAUDIUS IAP: ACL Groups:
ID: HAMLET IAP: ACL Groups:
ID: GERTRUDE IAP: ACL Groups:
ID: HORATIO IAP: ACL Groups:
ID: POLONIUS IAP: ACL Groups:
ID: LAERTES IAP: ACL Groups:
ID: OPHELIA IAP: ACL Groups:

Imaginary Project Data Base
Figure 1-3

As the figure shows, all members of the project share a single origin directory, <DRAMA>DENMARK>ELSINORE. They also share membership in a common access group, .DANES.

Both the directory and the access group are project defaults: they were defined for the project by the System Administrator, and so did not need to be defined for each member of the project. Nonetheless, each of the project's members, like all other users on the system, has his or her own unique ID and password. This provides good login security. It also makes it possible to change any user's profile if the need for special privileges arises.

For example, suppose that project leader CLAUDIUS determines that he and HAMLET need special access rights to a group of files. He edits the project's data base, using EDIT_PROFILE, and assigns the two men to the group .PRINCES, one of the access groups provided by the System Administrator as part of the project's limits. Figure 1-4 shows the project data base once this is done. Most members still share the default attributes. HAMLET and CLAUDIUS still share the default origin directory; but they now have their own access groups, rather than the default ones.

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.

SKETCHING OUT YOUR DATA BASE

Think About Your System

With these sketches and the previous examples in mind, you can begin to think about your own system and how you might organize its data base. What groups do your users seem to fall into? Are there some "natural" dividing lines you might use to divide users into projects, or to use when assigning 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? This sort of thinking is the first step towards setting up your data base.

Consider Degrees of Security

The next question you want to consider is, what degree of security do you want for your system?

The type of User Profile Data Base you create may well depend on your answer to this question. In particular, the use of projects often correlates with the degree of security desired on the system.

```
Project Administrator:  CLAUDIUS

Project Limits — ACL Groups:
    .DANES
    .PRINCES
    .OTHER
    .SPECIAL
    .CHARACTERS

Project Defaults:

  Default IAP:  <DRAMA>DENMARK>ELSINORE
  Default ACL Groups:  .DANES

ID:  CLAUDIUS
IAP:
ACL Groups:
    .DANES
    .PRINCES

ID:  HAMLET
IAP:
ACL Groups:
    .DANES
    .PRINCES

ID:  GERTRUDE
IAP:
ACL Groups:

ID:  HORATIO
IAP:
ACL Groups:

ID:  POLONIUS
IAP:
ACL Groups:

ID:  LAERTES
IAP:
ACL Groups:

ID:  OPHELIA
IAP:
ACL Groups:
```

Imaginary Project Data Base After Changes
Figure 1-4

Essentially, there are three main types of system:

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

  Such a system was shown as Example 2 in EXAMPLES OF SYSTEM PLANNING, earlier in this chapter.

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

  This type of system was shown in Example 1 of EXAMPLES OF SYSTEM PLANNING.

- A mixed system, which combines tight security on some projects (and for some users) with a friendly environment 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).

  This type of system was shown as Example 4 in EXAMPLES OF SYSTEM PLANNING.

Draw Up Some Lists

Once you have an idea of how you want your system (and its data base) to be organized, you will probably find it helpful to draw up some lists of users and projects. These lists will help you visualize your system more precisely. Also, you can use the lists as reminders when you are creating the data base with EDIT_PROFILE. Many people find working with EDIT_PROFILE easier if they have written copies of the information they want to enter. (Chapter 2 and Chapter 4 contain step-by-step examples of how EDIT_PROFILE prompts for information.)

Drawing Up Lists for Project-based Systems: If you are designing a project-based system, you may want to use the following steps:

1. If you are using projects, your first list may well be a master list of the projects you want on your system. Include, for each project:

   - The project's name
   - The name and the user-id of the Project Administrator

- The default Initial Attach Point (if you intend to use one)
- The default ACL groups (if you intend to use them)
- A list of other ACL groups you want to make available as project-specific groups for the project.

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

   At the end of this step, you should have a master list of projects (from Step 1), a master list of users, and a project-specific list of users for each project. (Each project can contain up to 2000 users.)

   Figure 1-5 shows a sample form for creating a master user list. Figure 1-6 shows a sample form for creating a project data list.

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

   a. For each user, define a user-id and a temporary password. (You may choose to have users share IDs and passwords, or you may want a separate ID and password for each user.) You can either assign the IDs yourself, or send around 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 Chapters 17 and 18.

   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 system-wide ACL groups (if any) to which you want this user to belong.

SYSTEM USER LIST

Drawn up by: [signature]

Date: 7/8/82

SYSTEM NAME: SYSX

ID: FROG PASSWORD: GREEN DEFAULT PROJECT: — OTHER PROJECTS: SWAMP HOLLYWOOD	ACL GROUPS: . AMPHIB
ID: PIG PASSWORD: BEAUTIFUL_STAR DEFAULT PROJECT: — OTHER PROJECTS: HOLLYWOOD IOWA	ACL GROUPS: • VIPS • PIGS • BEAUTIES
ID: POSSUM PASSWORD: — DEFAULT PROJECT: SWAMP OTHER PROJECTS: —	ACL GROUPS:
ID: DOG PASSWORD: ARF DEFAULT PROJECT: — OTHER PROJECTS: HOLLYWOOD	ACL GROUPS: • BEST_FRIENDS

Sample System List
Figure 1-5

PROJECT DATA LIST

Drawn up by: AAR

Date: 7/12/82

PROJECT NAME Hollywood

PROJECT ADMINISTRATOR: NAME Ann L. Rooney

USER-ID Ann

ACL GROUPS DEFINED FOR THIS PROJECT:

. STARS . HEROES . VILLAINS . OTHERS
. SUPERSTARS . HEROINES . OUTERSPACE

DEFAULT PROFILE:

INITIAL ATTACH POINT: <HOLLYWOOD>Movies
ACL GROUPS: .STARS

USERS

ID: Frog
IAP: _
ACL Groups: —

ID: Pig
IAP: _
ACL Groups: . STARS
. SUPERSTARS

ID: Dog
IAP: _
ACL Groups: _

Sample Project List
Figure 1-6

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.

Drawing Up Lists for a Friendly System: If you are not using projects on your system, you will need to construct only a master list for users. This list should contain, for each user:

- A user-id
- A temporary password (optional)
- A list of ACL groups to which the user will belong (optional)

Rules for User Attributes

The following rules govern user attributes:

User-id:

- May be 1-32 characters in length
- Must begin with an alphabetic character
- May contain letters, numbers, and the special characters ., _, and $.

Password:

- May be 1-16 characters in length
- May contain any characters except PRIMOS reserved characters. (Reserved characters are defined in the glossary of the Prime User's Guide.)

System Administrator:

- There may be only one user-id representing the System Administrator on a system, at any given time. (Any number of people may share administrative duties by sharing the use of that ID.)

Project Administrators:

- There may be only one Project Administrator for any project at any given time.
- There may be up to 24 PA's on a given system.
- A PA may administer more than one project at one time.
- The System Administrator may also act as Project Administrator for one or more projects.
- If the only project in use is DEFAULT, the System Administrator is automatically its Project Administrator.

Project Names:

- Project names follow the same rules as user-ids.

ACL Group Names:

- May be from 2 to 32 characters long
- Must begin with a dot
- May include letters, numbers, and the special characters ., _, and $.

When your lists are complete, you are ready to set up the User Profile data base for your system. Chapter 2, INSTALLATION, and Chapter 4, EDIT_PROFILE, will show you how to create your data base when you install your system.

Note

If you have a large number of users on your system, you may want to construct a smaller, temporary data base to begin with. Chapter 4 of this book and the Rev. 19 Planning and Installation Guide give examples of how to create a temporary data base, how to create your "real" data base while using the temporary one, and how to shift over to using the "real" data base.

The Rev. 19 Planning and Installation Guide also explains how to shift most easily from a Rev. 18 system of using UFD-names as usernames to a Rev. 19 system of user-ids.

PLANNING FOR SYSTEM CONFIGURATION

Why Is Configuration Needed?

PRIMOS is the operating system for the 50 Series computers. It contains the code which manages:

- Time-shared access for up to 128 users
- Segmented virtual address space for programs up to 32 megabytes per user
- Input/output control
- File system
- Interactive terminal access and phantom user non-interactive jobs
- Communications systems

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

PRIMOS is delivered in a single version which is configured at every cold start to support between 1 and 128 users. PRIMOS takes its configuration information from a file (usually named CONFIG) which defines the number of users to be supported, the resources to be made available to each user (for example, the number of nonshared segments in each user's working space), the disposition of AMLC lines, and other system parameters.

Because these specifics vary from site to site, you will want to decide how you want your own system configured.

The Configuration File

A configuration file is composed of a series of configuration directives, one per line. It is usually constructed under PRIMOS II (Prime's single-user operating system), using the nonshared editor, NSED. The file is usually named CONFIG. It must be stored in CMDNCO.

It is possible to create a CONFIG file, using only the minimum directives, and use that file to bring up PRIMOS for the first time. You may then modify the CONFIG file under PRIMOS, using the regular editor, ED. The next time you cold start the system, the modified file will be used for configuration, and the modifications will take effect.

This chapter explains which directives are essential to have in a CONFIG file. Chapter 2 shows examples of creating a temporary CONFIG file at installation time. Chapter 3 provides a full reference to all configuration directives.

CONFIGURATION DIRECTIVES

Configuration directives may be thought of as falling into five categories:

- Necessary directives which must be set for the system to function at all.
- Useful directives which need not be set, but which if set correctly make the system function better.
- Default-changing directives which the system doesn't care about but the System Administrator might.
- Equipment-specific directives which should be used if you have certain equipment attached to the computer.
- Rarely-used directives which are used for system debugging or which are functionally obsolete; they should be avoided.

Details on all these directives, their arguments and uses, are in Chapter 3. All numerical arguments to configuration directives are octal numbers. Decimal equivalents are provided for ease of calculation.

Necessary Directives

Command Device: This is the device on which CMDNCO is searched when a user invokes an external PRIMOS command. It is set with the COMDEV directive. The argument is the physical device number of the partition which will initially be assigned as logical device 0 — the command device. See Appendix A for details on constructing physical device numbers. If the command device is the same as the paging device, the disk is said to be split; split command disks are not commonly used.

Paging Device: A physical device (or partition) must be set aside for paging. This is set with the PAGDEV directive. On current systems, with storage module devices or other reasonably-sized disks, it is not necessary to specify the number of records to be allocated for paging. Just make sure that the partition is large enough by following the calculations in Chapter 6. The operating system will calculate the size of NSEG, the total number of system segments.

On smaller systems you may have to specify an alternate paging device with the ALTDEV directive. In this case, the size of the primary paging device should be explicitly specified; this sets the point at which paging switches from the primary to the alternate device.

The total space for paging cannot exceed 65536 ('200000) records.

Number of Users: There are four categories of users — terminal, phantom, remote, and slave. The latter two types are for networks only. The total number of configured users of all types must be less than or equal to 128.

The number of terminal users is set by the NTUSR directive. There is no default value; this directive must be included in the file. You must set the value to at least the number of terminals you will have connected to the computer. If you set the value higher than necessary it may make it easier to add terminals in the future. However, it will also increase the size of memory required for PRIMOS (wired memory), decrease the memory available for paging, and degrade system performance in direct relation to the number of excess terminal users configured.

Phantom users may be thought of as users at imaginary terminals. The number of phantom users is set by the NPUSR directive; the default value is 0. You must configure phantoms for:

- Each spooler to be used (1 per spooler).
- The Batch monitor.
- Batch queues (up to 1 per queue).
- The network server NETMAN (if you are using networks).

You may want to configure some phantoms to be available for terminal users. If so, about 1 phantom for each 5 terminal users is a reasonable number to start with. If your terminal users complain that phantoms are not available, you may want to increase the number configured. If there are no complaints, you may want to decrease the number configured until there are complaints and then increase it slightly.

Remote users are terminal users on other systems who wish to log into your system through their computer which is networked to yours. The number of remote users is set with the NRUSR directive. The default value is 0. If you set the value to 0 or omit it from the file, no one can log in remotely to your system regardless of any network connections. You can allow up to 32 ('40) remote users on your system. If you specify any remote users, you must include the NET ON directive or the system will start up but will not allow any network activity.

Slave users are processes on your system which handle requests for file access, attaching, etc. by users on other systems. The number of slave users is set by the NSLUSR directive. The default value is 0. If you set the value to 0 or omit it from the file, no one can access files on your system from other systems networked to yours. If you specify any slave users, you must also include the NET ON directive or the system will start up but will not allow any network activity.

You should consult with your Prime System Analyst to set initial values for remote and slave users. The number needed will vary greatly depending upon your specific network, computers, and work being done.

Networks: Networks are enabled by including the NET ON directive in the configuration file. This causes the startup program to read the file CMDNCO>NETCON to determine the configuration of the network. This file is created by the NETCFG utility described in Chapter 18. If you have configured either remote or slave users, you must include the NET ON directive in the configuration file.

End of File: The end of the configuration data file is marked by the directive GO. This directive must be the last noncomment line of the configuration data file.

Useful Directives

Memory Validation: Memory validation occurs at each cold start. The amount of physical memory to be used (in 2048-byte pages) is specified by the MAXPAG directive. If the directive is not included, only 512K bytes of memory can be used. The table below is a quick reference for setting MAXPAG.

Memory (MBytes)	Memory (pages)	MAXPAG argument
8	4096	10000
6	3072	6000
4	2048	4000
3	1536	3000
2	1024	2000
1-1/2	768	1400
1	512	1000
1/2	256	400

Per-user Segments: Each user is given 32 ('40) segments of virtual address space by default. If large software packages are being used or large programs being developed, it may be necessary to increase the number of segments available to each user. This is done with the NUSEG directive. Each user may have as many as 240 ('360) segments. However, as the number of segments per user increases, the expected amount of paging space will also increase. The System Administrator then has to make a trade-off between decreased performance and an increase in the amount of paging space available. (See the PAGDEV directive.)

Assignable AMLC Lines: Assignable AMLC lines are used for serial devices such as serial printers. The number of buffers for assignable AMLC lines is set by the NAMLC directive. The default value is 0; the constraint is that the number of assignable AMLC lines plus the number of terminal users cannot exceed 127 ('177). The PRIMOS AMLC command actually specifies which lines are assignable. This is done by setting bits 9-16 of the lword argument to 0. Complete details of the AMLC command are in the System Operator's Guide. Such lines often need to have their buffer sizes changed from the default values. This is done with the AMLBUF directive, described below under Changing Buffer Sizes.

Event Logging: There are two types of event logging mechanisms — system and network. They are discussed in detail in Chapter 12 of this book and in the System Operator's Guide. These mechanisms can be enabled or disabled separately with the LOGREC (system) and NETREC (network) directives; the arguments and meanings are the same for both directives.

If the argument is 0, event logging is enabled. If the argument is positive, event logging is enabled but a warning message is printed at the supervisor terminal. Prior to Rev. 19, quotas could be set on the event logging files with a positive argument to these directives; at Rev. 19, file sizes can be controlled by the more general quota system. Users with pre-Rev. 19 configuration files will find that event logging will occur as before. However, it is recommended that they change any positive arguments to LOGREC or NETREC to 0. If a negative value is given (such as 177777), event logging is disabled.

Note

Event logging can be enabled or disabled while PRIMOS is running with the EVENT_LOG command. However, if the command device is write-protected, event logging should be disabled with the LOGREC and NETREC directives since the event loggers will try to write the fact of PRIMOS startup to the event logging file on the command device, causing an error.

Configuration Memory Requirements: This value can be approximated by including the WIRMEM directive in the file. When the system is cold started, the amount of wired memory, in 2048-byte pages, is printed at the supervisor terminal. Although this value changes during operation, it does give some indication of the memory used for a particular system configuration.

Default-changing Directives

Abbreviation Processor: Normally users are allowed to create abbreviation files and use these files on the command line. If you do not wish users to have this facility, you can disable the abbreviation

processor by including the directive ABBREV NO in the configuration file. If you do nothing (or include ABBREV YES), users will be allowed to have command line abbreviations.

Erase and Kill Characters: Prime supplies " and ? as the erase (one character) and kill (entire line) characters for the system. You may change these characters with the ERASE and KILL directives. If you do change either the erase or kill character, be sure that all system users are aware of this, since Prime's documentation talks in terms of the Prime-supplied defaults.

Note

Whether or not you change the erase or kill characters system-wide, each user has the ability to make a change for his terminal while logged in by using the -ERASE or -KILL option of the PRIMOS TERM command. Details of TERM are given in the Prime User's Guide.

Logging In and Logging Out: There are four directives that modify the login and logout procedure somewhat.

When a user logs in or out, a message is normally printed at the supervisor terminal to this effect. This allows the System Administrator to have a record of these transactions. However, you may decide that such detailed information is not necessary, especially since it may use a significant amount of paper with a hard-copy supervisor terminal. These messages can be disabled at the supervisor terminal by including LOGMSG NO in the configuration file. Doing nothing (or including LOGMSG YES) causes these messages to be printed at the supervisor terminal.

If users are logged in, they are normally permitted to give the LOGIN command. They will then be logged out and logged in again according to the arguments of the LOGIN command. (See the discussion under LOGLOG in Chapter 3 for details of procedure if external LOGIN and/or LOGOUT programs exist.) If you wish to force users to log out explicitly before being able to log in again, include the LOGLOG NO directive in the configuration file. If you do nothing (or include LOGLOG YES), users will be able to use LOGIN while logged in.

If the AMLC terminal line is disconnected (or the terminal is turned off), the user is not logged out. If you wish the user to be logged out in such a situation, include DISLOG YES in your configuration file. If you do nothing (or include DISLOG NO), users will not be logged out upon disconnection.

You can decide how long a terminal can sit idle before it is automatically logged out (inactivity timeout). Prime supplies a default time of 1000 ('1750) minutes, that is, 16 hours 40 minutes. If you do nothing, this is the inactivity time that will be allowed. You

will probably want to set this value to a considerably shorter time, say 1 hour. This is done with the LOUTQM directive. For example, LOUTQM 74 sets the inactivity timeout to 1 hour (1 hour equals 60 minutes, whose octal value is 74).

Printing Configuration Directives: Normally configuration directives are not printed at the supervisor terminal as they are processed. If you want to see these directives printed at the supervisor terminal, include the directive TYPOUT YES in the file. All directives subsequent to TYPOUT YES will be printed at the supervisor terminal until either the directive TYPOUT NO or the directive GO is encountered.

Equipment-specific Directives

Changing Buffer Sizes: Although many devices operate with the default buffer sizes it is often desirable (and, in some cases, necessary) to change these sizes.

Buffer sizes for each terminal user and each assignable AMLC line can be modified with the AMLBUF directive. There is an extensive discussion of this in Chapter 8. Most terminals use the default input buffer size of 128 ('200) but increase the output buffer size from 192 ('300) to 512 ('1000) and the DMQ buffer size from 32 ('40) to 128 ('200). Terminals for special purposes such as OAS, FORMS, or DPTX may have performance improved by different modifications to the buffer size. For example, a PT65 terminal being used for OAS might have its input buffer changed to 832 ('1500), output buffer to 512 ('1000), and DMQ buffer to 128 ('200).

Buffer sizes for assignable AMLC lines will vary with the specific device used. Your Prime System Analyst will help you decide which buffer sizes, if any, need to be changed.

Input and output buffer sizes for all remote users are set with the REMBUF directive. (The DMQ buffer size for remote users is set by their local system.) Usually the default input and output buffer sizes of 128 ('200) and 192 ('300) are used. However, to speed remote file handling these values might be increased to 1024 ('2000) for input and 256 ('400) for output. Again, optimum settings for these remote users' buffers will depend upon the specifics of the network and what kinds of operations the users are performing.

If one of the AMLC lines were to be attached to a high-speed input device, data could come into the DMC tumble tables faster than it could be removed. In this case, data would be lost. The input buffer size for the AMLC controller DMC tumble tables can be set with the AMLIBL directive. If the directive is not included the default buffer size is 48 ('60). The maximum size for the buffers depends upon the number of controllers and the amount of space available in the system for buffers; it will be set to this calculated value by either AMLIBL 0 or

AMLIBL with no argument. The need to change table sizes is not quite rare since computer-to-computer links now use SMLC rather than AMLC connections.

AMLC Programmable Clock: There is a software programmable clock in the AMLC hardware. It can be specified in the range from 29 ('35) to 19200 ('45400) baud as required by the device on that AMLC line. To specify the programmable clock for an AMLC line, the configuration argument to the AMLC command must have bits 8-10 set to 4 (bit pattern 100). If bits 8-10 are set to 4 and the AMLCLK directive is not in the configuration data file, the programmable clock will assume the value of 9600 ('22600) baud. See the System Operator's Guide for complete details of the AMLC command.

Telephone Lines: There are three timers associated with dial-up lines. You will probably want to use the defaults for the first two. However, you may want to set the third (gracetime) timer. This is essentially the amount of time the system will allow a carrier to remain active without being connected to a logged-in process. A reasonable value would be 1 minute. This is 60 seconds or 600 tenths of a second. Converting to octal numbers gives a value of '1130 for this argument. The directive would be:

```
AMLTIM 2 3410 1130
```

The first two arguments are the defaults for the other timers. (See the AMLTIM directive in Chapter 3.) You should get complete details on this from your Prime System Analyst.

When the user logs out, the carrier will remain active for the period specified by the gracetime argument. If the DTRDRP directive is included in the configuration file, the carrier is dropped immediately regardless of the value of gracetime. This will prevent a user on a dial-in line from successfully logging in while already logged in.

Nonstandard Supervisor Terminal: The regularly-supplied supervisor terminal has a baud rate of 300, an input buffer of 256 ('400) bytes, and an output buffer of 384 ('600) bytes. If you are using a device for the supervisor terminal which does not conform to these values, it will be necessary to set its rates.

The supervisor terminal baud rate is set with the ASRATE directive. Besides the default baud rate of 300, baud rates of 110, 1200, and 9600 are also supported. If the ASRATE directive is used, it must be the first directive in the configuration file.

If it is necessary to change either the input or output buffer size for the supervisor terminal, use the ASRBUF directive.

Synchronous Lines: SMLC lines to other computers and devices are enabled and configured with the SMLC directives. Which one is used and what values are assigned depend upon the specific hardware and controllers on your system. There is a detailed discussion in Chapter 3. Your Prime System Analyst will help you with the hardware.

Uninterruptible Power Supply: If you have an uninterruptible power supply, you can specify how long to wait to bring the system up after a warm start following a power failure. This delay is to allow the disks to come up to full speed before attempting to access them. This is done with the UPS directive. A positive value gives the time in seconds to wait -- typically '100 for a storage module. An argument of 0 results in a warm start followed by a halt. If you don't have an uninterruptible power supply, leave this directive out of the configuration file.

Rarely-used Directives

Prepaging: When a page fault occurs and there are no unused memory pages, not just one but several least-recently-used pages are written out to disk from memory making them available. Normally 3 pages are written out at a time. This value can be changed with the PREPAG directive. Unless your Prime System Analyst recommends changing this, use the default value by omitting this directive from the configuration file.

One-line Configuration: Prior to Rev. 15, system configuration was done with the one-line CONFIG command rather than with a configuration data file. For compatibility, this is still supported both as a command and as a configuration directive.

New systems should not use this directive; already existing systems are encouraged to convert to the appropriate individual directives.

Per-user File Units: PRIMOS normally guarantees that at least 16 ('20) file units will be available to each user and that a user might have up to 127 ('177) file units open at one time. Since this situation is almost always acceptable, it is not expected that the FILUNT directive will be used often to change these defaults. Prior to Rev. 19, the total number of file units available to be open on the system could be reduced with a third argument to FILUNT. This is no longer necessary. If you already have a configuration file using three arguments to FILUNT, the third argument will be ignored and the system value of 3247 ('6257) will be used.

System User Segments: The total number of segments required by the system is the sum of those required by the operating system (PRIMOS) and by all the configured users. The number of segments is specified

by the NSEG directive. However, the number of page maps available (as specified by PAGDEV and ALTDEV) may be fewer than the number of possible user segments. In this case, the number of segments specified by NSEG could not be allocated and the number would be automatically reduced to conform with the paging space available. Since this reduction is done automatically at Rev. 19, there is no longer any reason to manually change the default value from 1022 ('1776), which is also the maximum value.

File System Read/Write Lock: The default file system read/write lock is set to allow N readers or 1 writer. If the system-wide value is changed from this, a number of utilities and subsystems will not work. If it is necessary to change the read/write lock of any file or set of files, use the PRIMOS RWLOCK command (rather than the RWLOCK directive) for this.

Kernel Debugger: Prime's assembly language debugger can be wired into PRIMOS at system startup with the VPSD directive. This is useful for debugging the operating system but not for specifying any useful system configuration in a production environment. It is not expected that customers will have any use for this directive.

2
Installation

INTRODUCTION

If you are using a Prime system for the first time, your initial installation of software will be the installation of Revision 19 software. If you already have a Prime system, you will be converting your current version of software to Revision 19.

This chapter only outlines the steps you must take for either the initial installation or the conversion, whichever applies to your system. Full information on both procedures is found in the Rev. 19.0 Planning and Installation Guide.

INITIAL INSTALLATION

All Prime software products shipped with new Prime computer systems are stored on a storage module disk pack called the Master Disk. The Master Disk contains the operating system (PRIMOS), the utilities, the nonchargeable software products, and any separately priced software products you have ordered. After your computer system has been installed at your site, you are responsible for installing the products from the Master Disk.

The following list outlines the procedure for the initial installation of the software from the Master Disk.

1. Turn on power to equipment.

2. Mount the Master Disk.

3. Load (bootstrap) PRIMOS II, Prime's single-user operating system, into the system.

4. Attach to UFD CMDNC0 and then:

   - Create a configuration file, using the nonshared Editor NSED.

   - Copy the command file C_PRMO.TEMPLATE from UFD PRIRUN into CMDNC0 and rename it C_PRMO. The C_PRMO.TEMPLATE is listed in Figure 2-1.

   - Modify the command file C_PRMO for the initial system startup, again with the nonshared Editor NSED.

5. Attach to UFD PRIRUN and resume PRIMOS.

6. Set maximum users to 0.

7. Set system date and time.

At this point you have installed PRIMOS in the command partition. However, before any users can log in, the System Administrator must:

8. Set up the User Profile data base with EDIT_PROFILE.

9. Set security on your system with Access Control Lists (ACLs).

10. Set quotas on top-level user UFDs, if you wish, with the SET_QUOTA command.

11. If you have chargeable software to install, attach to UFD SYSTEM and then:

    - Modify and run the command file CREATE.ALL.COMI.

    - Modify and run the command file INSTALL.ALL.COMI.

12. Attach to UFD CMDNC0 and modify the command file C_PRMO again, to suit your needs.

13. Shut down and cold start the system at the supervisor terminal.

```
/* C_PRMO.TEMPLATE, PRIRUN, JNK, 07/21/81
/* TEMPLATE FOR MAKING C_PRMO FILE FOR BRINGING UP PRIMOS
/* Copyright (C) 1980, Prime Computer, Inc., Wellesley, MA  02181
/*
CONFIG -DATA                /* specify CONFIG file after -DATA
ADDISK                      /* specify local disks to be added
AMLC TTY                    /* specify AMLC lines
OPR 1                       /* SHARE REQUIRES OPR 1
SHARE SYSTEM>ED2000 2000         /* SHARE the editor  -  ED
SHARE SYSTEM>S2050 2050 700
R SYSTEM>S4000 1/1               /*  SHARE FORTRAN LIBRARIES
SHARE 2050
SHARE SYSTEM>SP2121 2121
R SYSTEM>SP4000 1/10               /* SHARE SPL LIBRARY
SHARE SYSTEM>S$2167 2167
R SYSTEM>S$4000 1/12               /* SHARE SPOOL LIBRARIES
OPR 0
PROP PR0 -START                  /* START SPOOLER PHANTOM
BATCH  -START               /* STARTUP BATCH MONITOR
CO SYSTEM>BASICV.SHARE.COMI 7    /* SHARE BASICV COMPILER
CO SYSTEM>COBOL.SHARE.COMI 7     /* SHARE COBOL COMPILER AND LIBRARY
CO SYSTEM>DBG.SHARE.COMI 7       /* SHARE DEBUGGER
CO SYSTEM>DBMS.SHARE.COMI  7       /* SHARE DBMS
CO SYSTEM>DPTX-DSC.SHARE.COMI 7    /* SHARE DPTX-DSC
CO SYSTEM>DPTX-TCF.SHARE.COMI 7    /* SHARE DPTX-TCF
CO SYSTEM>EMACS.SHARE.COMI 7        /* SHARE EMACS
CO SYSTEM>FED.SHARE.COMI 7         /* SHARE FED
CO SYSTEM>FORMS.SHARE.COMI 7       /* SHARE FORMS LIBRARY
CO SYSTEM>FTS.SHARE.COMI 7         /* SHARE FTS
CO SYSTEM>F77.SHARE.COMI 7         /* SHARE F77 COMPILER
CO SYSTEM>MIDAS.SHARE.COMI 7       /* SHARE MIDAS LIBRARY
CO SYSTEM>PASCAL.SHARE.COMI 7      /* SHARE PASCAL COMPILER
CO SYSTEM>PL1G.SHARE.COMI 7        /* SHARE PL1G COMPILER
CO SYSTEM>POWERPLUS.SHARE.COMI 7        /* SHARE POWER
CO SYSTEM>VISTA.SHARE.COMI 7            /* SHARE VISTA
CO SYSTEM>VRPG.SHARE.COMI 7             /* SHARE VRPG
CLOSE 7
/* SET THE DATE AND TIME **********
CO -END
```

The Installation Template: C_PRMO.TEMPLATE
Figure 2-1

CONVERTING A REV. 18 SYSTEM TO REV. 19

If you are currently running pre-Rev. 19 Prime software and plan to change to Rev. 19, you will receive a tape (or tapes) with the following Rev. 19 software:

- The TOOLS directory, which you will use in the conversion.
- The M190U1 partition of the Master Disk, which contains nonchargeable software.
- The M190V1 partition, which contains the sources of nonchargeable software.
- Those portions of the M190C1 partition (if any) that are in use at your site. The M190C1 partition contains separately chargeable software. This partition will probably be a separate tape; your analyst should know about it.
- Those portions of the M190D1 partition (if any) that are in use at your site. The M190D1 partition contains sources of certain separately chargeable software.

You should then follow the conversion steps outlined below. Full details are in the Rev. 19.0 Planning and Installation Guide.

Note

You must start with a system currently running Rev. 18.3 PRIMOS. If you are running a version lower than 18.3, you must install 18.3 before you attempt to convert to Rev. 19. (Your analyst should be able to help you get to Rev. 18.3.)

Conversion Steps

1. Clear the system of users and set maximum users to 2.
2. Back up the existing command partition.
3. Restore the TOOLS directory from your tape as a top-level UFD in the command partition.
4. Run CONVERT_PROFILE (a program in the TOOLS directory).
5. Restore Rev. 19 Master Disk software (nonchargeable) from your tape(s), using LOAD_19 (another program in the TOOLS directory). This step automatically installs nearly all of the nonchargeable software.
6. Restore from your tape(s) any chargeable software you have purchased. (You will install this at Steps 7 and 12.)

7. Install PRIMENET software (if you use it).

8. Boot Rev. 19.

9. Issue the LOGOUT ALL command (to log out phantoms).

10. Set system date and time.

11. Execute PROFILE_CONVERSION (provided by the CONVERT_PROFILE utility from Step 4).

12. Install any chargeable software (other than PRIMENET) you have purchased.

13. Use FIX_DISK to convert partitions to Rev. 19.

14. Convert to ACLs with CONVERT_ACLS (a third program in the TOOLS directory), if you want to use ACLs.

15. Redefine your Batch queues (if you use Batch).

16. Set quotas on top-level user UFDs, if you wish, with the SET_QUOTA command.

17. Build your Rev. 19.0 C_PRMO file.

18. Bring down your system and bring it up again to activate all your Rev. 19 software.

3
Configuration Directives

CONFIGURING THE SYSTEM

PRIMOS is configured each time the system is cold started. This allows the System Administrator to reconfigure a system as necessary to meet changing needs.

At most times, however, the same configuration is used each time it is cold started. For this reason, the directives that configure the system are written into a data file.

This data file is generally named CONFIG. It is run by the CONFIG command, using the format:

```
CONFIG -DATA configuration_data_filename
```

For example:

```
CONFIG -DATA CONFIG
```

This CONFIG command must be the first command in the command file C_PRMO, which brings up PRIMOS. Once these two files, CONFIG and C_PRMO, are created, bringing up the system becomes simple.

Each Prime system comes with a C_PRMO template, which must be completed to suit the installation. This template is shown in Chapter 2. The CONFIG file varies more widely; so CONFIG files are created by individual administrators. (See the System Operator's Guide for an example of a cold start using these two files.) | 19.0

To alter a system's configuration, simply alter the CONFIG file. The next time the system is brought up, the new configuration takes effect.

Creating the CONFIG File

The CONFIG file is created and altered using the Editor. This chapter procides a detailed explanation of how to use each directive. See Chapter 1 for a discussion of reasons for their use.

Notes

All numerical values in the CONFIG must be in octal unless the documentation specifies otherwise.

19.0 See Appendix A for details on constructing physical device numbers needed as arguments for certain directives.

Network Information

19.0 If the computer is to be part of a network, certain parameters have to be set by CONFIG directives such as REMBUF, NSLUSR, NRUSR, and NETREC. Other information dealing with the computer's interface to the network is stored in the NETCON file. The Administrator creates the NETCON file, using the NETCFG utility. (See Chapter 18.) See the PRIMENET Guide for general network information.

In order for NETCON information to be used when the system is brought up, the Administrator must include the NET ON directive in the CONFIG file.

CONFIG Errors

If there are errors in the CONFIG file, the system will not come up. Instead, the operator will receive a message at the supervisor terminal telling him which CONFIG directive was faulty and requesting that he restart the system. When this happens, the operator must:

- Re-boot PRIMOS II.
- 19.0 Correct the faulty directive in the CONFIG file using the unshared editor, NSED.
- Bring up PRIMOS.

The most common causes of errors are:

- Out-of-range values for parameters (for example, a value of 12 for an SMLC line, which can only take values from 00 to 07).
- Values that are correct in themselves, but conflict with other values. (For example, values for NPUSR+NSLUSR+NRUSR+NTUSR must be <= '200 (decimal 128). Values of '100 for NPUSR and NTUSR, plus '20 for NRUSR and NSLUSR, though each correct in isolation, would produce a sum greater than '200, and would keep the system from starting up. 19.0
- Decimal numbers used by mistake for octal ones.

A list of CONFIG error messages appears at the end of this chapter.

CONFIG DIRECTIVES

Directives for configuration data files are given below. Comment lines may be inserted in this file; they should begin with the characters /*. All numerical parameters are octal unless otherwise specified.

▶ ABBREV { YES | NO }

The YES option (the default) allows users to use ABBREV files. The NO option prohibits the use of ABBREV files.

▶ ALTDEV physical-device [records]

Specifies the alternate paging device and, optionally, its size.

physical-device	The physical device number of the paging disk or partition.
records	The size of the alternate paging device. If specified, and the equivalent parameter is specified in PAGDEV, then the sum of PAGDEV's and ALTDEV's records arguments is used to calculate NSEG — the total number of segments in the system. (records is a 16-bit non-zero positive unsigned integer.) The total primary and alternate paging device records must be <= 65536.

The alternate paging device is used only if the primary paging device (PAGDEV) has records specified, or if the primary paging device is a storage module type disk.

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

Sets terminal I/O buffer sizes.

line	The AMLC line number for which buffer sizes are to be set. For terminal users, this value is the physical line number. For assignable AMLC lines, this value is NTUSR+NRUSR-1 for the first assignable AMLC line, NTUSR+NRUSR for the second, up to NTUSR+NRUSR+NALMC-2 for the last assignable AMLC line. Use the actual line number to change the size of the DMQ buffer on an assignable line. An invalid line number will result in the message: BAD LINE # IN AMLBUF COMMAND.
in-buff-size	The terminal input buffer size in words (two characters per word). If 0 is specified, buffer size is unchanged. The default value is '200 (128 decimal).
out-buff-size	The terminal output buffer size in words. If 0 is specified, buffer size is unchanged. The default value is '300 (192 decimal); the minimum value is '100 (64 decimal).
dmq-size	The size in words for the DMQ buffer (only meaningful if the system has DMQ AMLC controllers). If omitted or specified as 0, its value is not changed. The default value is '40 (32 decimal).

19.0 The AMLBUF directive cannot be used to modify input or output buffer size for remote users; the REMBUF directive should be used instead.

Total size of input buffers plus output buffers may not exceed 128K words. Exceeding this limit produces the message: NO ROOM. AMLBUF (TFLADJ).

No individual buffer may exceed '7777 (4095 decimal) words. Total size of DMQ buffers must be a power of 2 and may not exceed 64K words. Failing to meet the first specification produces the message: BAD DMQ AMLC CONFIGURATION. Failing to meet the second specification results in the message: UNABLE TO INITIALIZE DMQ AMLC (AINIT).

19.0 See Chapter 8 for details on configuring AMLC buffers.

▶ AMLCLK baudrate

baudrate specifies the desired baudrate for the software programmable clock in the AMLC hardware. The value specified must not be less than '35 (29 decimal) nor greater than '45400 (19200 decimal). (This operation was performed previously by changing the variable "AMLCLK" in the AMINIT module.)

▶ AMLIBL buffer-size

Defines the size of the DMC input tumble tables at cold start. AMLIBL explicitly sets the size of the input buffers or automatically allocates the maximum size allowed by the available buffer space.

buffer-size is the number of words allocated to each input buffer. There are two buffers for each AMLC controller and all buffers are made the same size. Except for the special value of 0 described below, the number must be greater than '20. The upper bound is variable depending on the number of controllers configured and the amount of space available in the system for buffers. If buffer-size is 0 or omitted, the size of the buffers is automatically calculated as the maximum possible. If the AMLIBL directive is not specified, the default buffer size is '60 (decimal 48). 19.0

If buffer-size is too small, the error message:

BAD AMLIBL PARAMETER (CINIT)

will be generated during cold start initialization. If buffer-size is too large, the error message:

INPUT BUFFERS TOO LARGE (AMINIT)

is generated at cold start initialization. Modify the parameter to be a value within the permissible range as described above.

▶ AMLTIM [ticks] [disctime] [gracetime]

The AMLTIM directive controls three variable event timers.

ticks — The time interval (in decimal tenths of a second) between carrier check operations. At the end of each period, the AMLC process checks each line for carrier loss. If a loss has occurred, the process forces the line's Data Terminal Ready (DTR) signal to be inactive until the next sample period, thus disconnecting its incoming telephone line. The value for ticks must be greater than 0. Default is 2 (0.2 seconds).

disctime — The time period (in decimal tenths of a second) for forcing the DTR signal to the modem inactive on lines without active carriers. Specifying a value of 0 disables this feature. Otherwise, the value specified must not be less than the value of ticks and is truncated to the nearest multiple of that value. Default is '3410 (1800 decimal — that is, 3 minutes).

gracetime — The minimum grace period (in decimal tenths of a second) for terminal lines that have active carriers but are not connected to logged-in processes. This argument effectively defines the minimum time granted for a caller to establish itself with a logged-in process. (The actual grace period varies randomly from gracetime to twice gracetime.) Default value is 0; the argument is disabled. The specified value (if not 0) must be greater than ticks, and is truncated to the nearest multiple of ticks.

▶ ASRATE control-word

Sets the supervisor terminal Baud rate.

control-word is an octal integer specifying supervisor terminal Baud rate.

control-word	Baud rate (decimal)
110	110
1010	300 (default)
2010	1200
3410	9600

If used, the ASRATE directive must be the first one in the configuration data file. This ensures that any subsequent configuration error messages are printed at the appropriate speed.

▶ ASRBUF line [in-buff-size [out-buff-size]]

Sets ASR terminal I/O buffer sizes.

line — The ASR line number. At present, only 0 is a valid line number.

in-buff-size — The ASR terminal input buffer size in words. If 0 is specified, buffer size is unchanged. The default value is '200 (128 decimal).

out-buff-size The ASR terminal output buffer size in words. If omitted or specified as 0, buffer size is unchanged. The default value is '300 (192 decimal); the minimum value is '100 (64 decimal). 19.0

▶ COMDEV physical-device

physical-device specifies the physical device number of the disk or partition on which the system UFD, CMDNCO, resides. This device becomes the system command device.

The command device must be specified.

▶ CONFIG options

Specifies basic system configuration.

This directive and its options are discussed at the end of this chapter. It is still supported for compatibility with pre-Rev. 15 configurations; its use is not recommended.

▶ DISLOG { YES / NO }

Sets disconnect logout option.

YES The user will be logged out if AMLC line is disconnected.

NO The user is not logged out if AMLC line is disconnected (Default).

▶ DTRDRP

This directive applies to the DTR (Data Terminal Ready) signal associated with an AMLC line. Specifying this CONFIG directive automatically forces the dropping of the DTR for any user when that user logs out, no matter what period of gracetime has been set by the AMLTIM directive. This directive is only useful for installations with 'port selectors' or dial-up modems. (See also the DROPDTR command in the System Operator's Guide.)

▶ ERASE { character | octal-value }

Specifies system default erase character; supplied default is ". Either of the following examples sets the system default erase character to !.

character — Printing ASCII character, such as " (which is the default value). For example:

```
ERASE !
```

octal-value — The octal value of any ASCII character. For example:

```
ERASE 241
```

▶ FILUNT reserved-unit max-unit

Defines the number of file units available to the user.

19.0 reserved-unit — Maximum number of file units guaranteed to be available to each user. Default is 16 ('20).

max-unit — Maximum number of units any one user may have open at one time. Default is 127 ('177).

If the FILUNT directive is not specified in the configuration file, the default values are used.

The maximum total number of units that may be open simultaneously by all users is '6257 (decimal 3247). The numbe· of file units guaranteed to be available to each user is 16; reserved-unit may be used to increase or decrease this quantity.

19.0 Since there are not enough file unit table entries to permit all users to have 127 file units open simultaneously (128*127=16256), the PRIMOS file management system subroutine SRCH$$ may return the error code E$FUIU (all units in use). If multiple cooperating processes (users) depend on having a certain number of file units available, the possibility of a deadlock exists. reserved-unit must be specified so that there are sufficient units available to prevent deadlock. That is reserved-unit*n <= '6257, where n is the total number of configured users.

Note

Prior to Rev. 19, the total-number of file units available to be open on the system could be reduced with a third argument to FILUNT. This is no longer necessary. A third argument to this directive will now be ignored. The value set by the system is '6257. 19.0

▶ GO

Marks end of configuration data file; any subsequent lines are ignored. The configuration data file must include a GO directive.

▶ KILL { character / octal-value }

Specifies system default KILL character; supplied default is ?

character A printing ASCII character, such as +, ?, *, etc.

octal-value The octal value of any ASCII character.

For example, KILL 230 sets the system default kill character to CONTROL-X.

▶ LOGLOG { YES / NO }

Allows LOGINs while logged in.

YES Users can use the LOGIN command while logged in (Default).

NO The LOGIN command is inhibited for a logged-in user.

This directive sets the variable LOGOVR in FIGCOM.

Note

If a user logs in when either he or another user is already logged in, the following will occur:

- If there is neither an external LOGOUT nor an external LOGIN program, the current user will be logged out with the internal LOGOUT program and the new user logged in by the internal LOGIN program.
- If there is an external LOGOUT program, but no external LOGIN program, the external LOGOUT program will run and log out the current user. The new user will be logged in by the internal LOGIN program.
- If there is no external LOGOUT program and there is an external LOGIN program, the external LOGIN program will run to first log out the current user and then log in the new user.
- If there is an external LOGOUT program and an external LOGIN program, the external LOGOUT program will run to log out the current user. Then the external LOGIN program will run to log in the new user.

▶ LOGMSG { YES / NO }

Prints LOGIN/LOGOUT messages.

YES LOGIN and LOGOUT messages are printed at the supervisor terminal (default).

NO LOGIN and LOGOUT messages are suppressed.

▶ LOGREC value

Enables/disables event logging.

If value is set to 0 or a positive number, event logging is enabled. If value is set to a negative number, event logging is disabled; this should be used when running a write-protected disk.

Prior to Rev. 19, a positive argument set the maximum size of the event logging file. This is no longer the case; the size of the event logging file can now be controlled by the more general quota system. If a positive value is used, system logging is enabled. However, the following message appears at the supervisor terminal: 19.0

```
LOGREC CONFIG DIRECTIVE NO LONGER SETS A QUOTA ON THE
SYSTEM EVENT LOGGING FILE.
PLEASE USE 'SET_QUOTA'.  (CINIT)
```

See the description of event logging and the EVENT_LOG command in Chapter 12 for more information. See the NETREC directive later in this chapter for enabling and disabling network event logging.

▶ LOUTQM minutes

Specifies inactivity time for automatic LOGOUT.

minutes is the number of minutes of inactivity (minus 1) allowed, at the terminal, before the user is automatically logged out. The default value is '1750 (1000 decimal). The value must be greater than 1.

▶ MAXPAG number-of-pages

Specifies number of pages of memory to validate. Memory validation occurs at each cold start. This directive must be included in order to use more than 512K Bytes of memory.

number-of-pages is the number of '4000 (2048 decimal) byte pages of physical memory to validate for use.

▶ NAMLC number-of-buffers

Specifies number of reserved buffers to be used for assigned AMLC lines in the system.

number-of-buffers is the number of reserved buffers to be used for
19.0 assigned AMLC lines. The default is 0. NAMLC + NTUSR must be <= '177 (127).

The AMLC command specifies which lines are assignable. (See the System Operator's Guide for details.)

▶ NET ON

This directive specifies that networks are to be configured. If this directive is not included, then networks will not be configured. (See Chapter 18.)

▶ NETREC value

Enables/disables network event logging.

If value is set to 0 or a positive number, network event logging is enabled. If value is set to a negative number, network event logging is disabled; this should be used when running a write-protected disk.

Prior to Rev. 19, a positive argument set the maximum size of the network event logging file. This is no longer the case; the size of
19.0 the network event logging file can now be controlled by the more general quota system. If a positive value is used, network logging is enabled. However, the following message appears at the supervisor terminal:

```
NETREC CONFIG DIRECTIVE NO LONGER SETS A QUOTA ON THE
NETWORK EVENT LOGGING FILE.
PLEASE USE 'SET_QUOTA'. (CINIT)
```

See the description on network event logging and the EVENT_LOG command in Chapter 12 for more information. See the LOGREC directive earlier in this chapter for enabling and disabling event logging.

▶ NPUSR number

Specifies number of phantom users to be configured.

number is the number of phantom users for which the system is to be configured. (Non-negative octal integer.) Default is 0; maximum is '40 (decimal 32). The maximum is '200 (128 decimal), minus the number of terminal, slave, and remote users (NTUSR, NSLUSR, and NRUSR). 19.0

Note

If networking is to be used, at least one phantom user must be configured for NETMAN.

▶ NRUSR number

Specifies the number of processes to be reserved for remote logins.

number is the number of remote users for which the system is to be configured. (Non-negative octal integer.) Default is 0. The maximum is '40 (32 decimal).

Note

If NRUSR is specified, then the NET ON directive must also be specified. 19.0

▶ NSEG number

Specifies the total virtual address space for a system (that is, the variable NSEG in SEGMENT 4).

number specifies the number of page maps to be allocated during system initialization. There may be fewer page maps available than the number of possible user segments. If the paging space specified by PAGDEV (and ALTDEV) does not allow NSEG segments to be allocated, NSEG is reduced to conform to the paging space requirements. The system allows a maximum of '1776 (1022 decimal) page maps. The default value for NSEG is 1022 decimal, the maximum.

It is not expected that this directive will be used often.

▶ NSLUSR number

Sets the number of slave processes (users) configured for a system. Each user accessing files on the local system from remote systems requires a slave process for the duration of the access. These slave processes come out of the PRIMOS 128-process pool.

number is the number of simultaneous remote file accesses the local system wishes to support. If this pool is exhausted when a remote user 19.0 makes an attach request, the E$NSLA (no NPX slaves available) error code is returned. NIUSR + NPUSR + NRUSR + NSLUSR must be <= '200 (128).

Note

If NSLUSR is specified, then the NET ON directive must also be specified.

▶ NIUSR number

Specifies number of terminal users.

number is the number of terminal (local system) users for which the system is to be configured. (Positive octal integer between '2 and '200.) The number of terminal users MUST be specified, either with the NIUSR directive or with the CONFIG command, 0/number.

19.0

NIUSR is added to NPUSR, NSLUSR, and NRUSR to determine the total number of users configured on the system. NPUSR + NRUSR + NIUSR + NSLUSR must be <= '200 (128 decimal).

▶ NUSEG number

Sets the size of the virtual address space for each user. number specifies the number (in octal) of segments available to each user process. PRIMOS system reserves room for a maximum of '360 (240 19.0 decimal) segments per user. The default value of number is '40 (32 decimal).

▶ PAGDEV physical-device [records]

Specifies paging device and size. PAGDEV must must be specified.

physical-device	The physical device number of the disk or partition on which paging is to take place.
records	The size of the paging disk (a non-negative 16-bit integer). If specified, this value (plus ALTDEV's value for records, if any) limits the total number of segments in the system. If records is specified and an alternate paging device has been specified by ALTDEV, then records defines the point at which page space allocation switches from the primary to the alternate paging device.

If records is small (for example, 1), almost all paging is forced to occur on the alternate device. However, the primary device is always used to page the PRIMOS segments and user 1's segments '6000 and '6002.

If the primary paging device is a storage module type disk, the number of paging records on the disk will be calculated correctly, even if records is not specified. After the number of records on the primary paging device is exhausted, paging will switch to ALTDEV (if ALTDEV was specified).

The total number of records on the primary and the alternate paging device must be <= 65536 records.

See Chapter 6 for more details on paging devices. | 19.0

▶ PREPAG pages

Specifies number of pages to prepage.

pages is a positive octal integer specifying the number of pages to prepage out when a page fault occurs. The default value is 3.

▶ REMBUF in-buff-size out-buff-size

Sets terminal input-output buffer sizes for remote users.

in-buff-size The terminal input buffer size in words. If 0 is specified, buffer size is unchanged. Default value is '200 words (128 words decimal, 256 characters). Minimum size is '113 words (75 words decimal, 150 characters).

out-buff-size The terminal output buffer size in words. If 0 is specified, the buffer size remains unchanged. Default size is '300 words (192 words decimal, 384 characters). Minimum size is '100 words (64 words decimal, 128 characters).

▶ RWLOCK value

Specifies file system read/write lock setting.

value is an octal integer specifying the system-wide file read/write lock. The default value is 1. Possible values are:

Value	Meaning
0	1 reader or 1 writer (writer has exclusive control)
1	N readers or 1 writer (writer has exclusive control)
3	N readers and 1 writer

19.0

Caution

N readers and N writers (value = 5) is no longer supported through the RWLOCK directive. This lock value slows all file access. In addition, many subsystems will not work correctly if the system default RWLOCK is not set to 1. Any file needing the N readers and N writers concurrent access can have its lock set to that mode with the PRIMOS-level RWLOCK command.

▶ SMLC ON
SMLC CNTRLR controller-number device-address
SMLC SMLCnn controller-number line-number
SMLC DSC line strap proc recv

Enables and configures SMLC lines. The four formats are discussed below.

SMLC ON: Enables the SMLC in the default ON configuration; the normal OFF default is SMLC disabled.

Default configuration is given in the table below.

Logical Line Number	Logical Controller	Controller Address	Controller Physical Line Number
0	0	'50	0
1	0	'50	1
2	0	'50	2
3	0	'50	3
4	1	'100000	0
5	1	'100000	1
6	1	'100000	2
7	1	'100000	3

Logical lines 0 through 3 are mapped to logical controller 0. Device address is '50, with physical line numbers 0 to 3.

Logical lines 4 through 7 are mapped to physical lines 0-3 on controller 1. Controller 1's address is '100000; thus, the controller is disabled. To enable controller 1, set its address to a valid device address ('50 or '51) with the "SMLC CNTRLR" directive.

The default configuration can be changed with the "SMLC CNTRLR" and "SMLC SMLCnn" directives, explained below. "SMLC CNTRLR" changes the mapping of logical controller to physical address. "SMLC SMLCnn" changes the mapping of a single logical line number. Either directive may be used separately. If both are needed, then the "SMLC CNTRLR" directive(s) must be given first. In other words, the controller must have been assigned its correct physical address before any new SMLC lines are assigned to it.

SMLC CNTRLR controller-number device-address: Specifies physical device number(s) of the SMLC controllers. Thus, it assigns a physical controller to a logical controller-number.

controller-number — The logical controller, either 0 or 1.

device-address — The physical device address of the specified controller. The default values are address '50 for controller 0, disabled address ('100000) for controller 1. If controller 1 is used, its address should not conflict with the address of any other peripheral controller. '51 is suggested.

Giving a logical controller number other than 0 or 1 produces the error message: BAD SMLC CONTROLLER MAPPING COMMAND.

If you map one logical controller to an address to which another controller is already mapped, SMLC automatically disables the previously-mapped controller, setting its address to '100000. The disabled controller may be enabled again by a new mapping directive. For example, the following directives first map controller 1 to address '50 (thus disabling controller 0, which was mapped to address '50 by default), and then map controller 0 to address '51 (thus enabling it again):

```
SMLC CNTRLR 1 50
SMLC CNTRLR 0 51
```

The operator may also disable a logical controller by setting its address to blank or to '100000. For example, controller 0 could be disabled by either of these directives:

```
SMLC CNTRLR 0 100000
SMLC CNTRLR 0
```

SMLC SMLCnn controller-number line-number: Maps logical line numbers to physical line numbers on a specified logical controller.

nn	The logical line number; values are 00 to 07.
controller-number	The logical controller, either 0, 1, or 100000.
line-number	The physical line number of the specified controller onto which the logical line number is mapped. The default values map SMLC00 to SMLC03 to physical lines 0 to 3 on the device at address '50; SMLC04 to SMLC07 to physical lines 0 to 3, on the device at address '51.

For example, the directive:

```
SMLC SMLC04 1 3
```

assigns logical line 4 to physical line 3 on controller 1.

Setting the controller number to a blank or to '100000 disables a logical line number. For example, logical line 7 can be disabled by either of these directives:

```
SMLC SMLC07
SMLC SMLC07 100000
```

Giving any value for controller-number other than 0, 1, blank, or 100000, results in the error message: BAD SMLC LINE MAPPING COMMAND

SMLC DSC line strap proc recv n: Specifies data set control configuration for a synchronous line on the SMLC/HSSMLC/MDLC board. Only the BSCMAN process currently interprets the data structures initialized by the DSC option. Arguments are specified as octal numbers:

line — Logical line number (0 through 7).

strap — Bit set indicates signals that are strapped ON by the software, (default is 1).

'1 Data Terminal Ready (DTR)

'2 Request to Send (RTS)

In addition, speed select in Europe is specified via the '10 bit:

Set to 1 - fast

Reset to 0 - slow

proc — Indicates data set control procedure (to be used for transmitting data) as follows (default is 2):

1 No data set orders. Usually used with DTR and RTS strapped ON, with modems used for 4-wire full duplex service.

2 Use data set orders as follows:
Issue RTS, wait for CLEAR to Send (CTS), send, drop RTS. Usually used with most half duplex modems.

3 Use data set orders as follows:
Wait for .NOT. Carrier Detect (CD), issue RTS, wait for CTS, send, drop RTS. Rarely used, but may be necessary with 201 series modems only if lines are very noisy. Try "2" first.

recv — Indicates whether the receiver is to be turned on before or after transmitting (default is 0):

0 Turn on receiver before transmitting. This provides the fastest response and should be used if possible.

1 Turn on receiver after transmitting. This setting must be used with 2-wire 201 series modems. This setting may be tried on other two wire systems only if problems appear that cannot be solved by other means.

The default setup, if no DSC is specified, is the equivalent of including the following line in the configuration data file:

```
SMLC DSC line 1 2 0
```

▶ TYPOUT { YES / NO }

Controls printing of configuration directives.

YES — Subsequent directives in configuration data file will be printed on the supervisor terminal as they are processed. Printing continues until a TYPOUT NO or GO directive is encountered.

NO — Commands are not printed as they are processed. Printing is suppressed until a TYPOUT YES or GO directive is encountered. TYPOUT NO is the default.

Any number of TYPOUT directives can be used in a configuration data file to print selected commands.

▶ UPS number

This directive enables an automatic warm start after power is restored following a power failure check. It is designed to be used when an Uninterruptible Power Supply (UPS) is used to maintain power to the CPU and memory.

number is the system UPS action variable. This variable determines what actions are taken after a power failure. Valid values of number are:

177777 — No UPS (default).

0 — UPS, but HALT on a warm-start.

>0 — Number of seconds to delay after warm-start. (No operator start is required.)

The number of seconds to delay after a warm-start is the amount of time it takes for the disk(s) to come up to the proper number of revolutions per minute. Typically, this is about 1 minute (about '100 seconds) for a storage module.

▶ VPSD

Including the VPSD directive in the CONFIG file causes the kernel VPSD debugger to be wired (and thus available for use) at system startup. Most installations will not want this debugger enabled during normal use.

▶ WIRMEM

This directive causes the size of wired memory, in pages, to be printed at the supervisor terminal during coldstart. This value changes as the system runs. However, it does give some idea of the relative memory cost of the selected configuration.

PRIMOS PRELOADER AND INITIALIZATION ERROR MESSAGES

Below is a list of all error messages generated by the PRIMOS preloader ('PRIMOS') and the PRIMOS and NETWORK initialization sequences. The majority of the CONFIG messages are fatal and cause configuration to terminate. Any error messages that do not come from the preloader ('PRIMOS') require that PRIMOS II be booted again from the control panel (that is, start over from the beginning); the offending directive (or lack thereof) should be corrected before attempting to bring up PRIMOS again. The module generating the error is shown in parenthesis at the end of the error message; this is part of the error message.

Error Messages

- file-system-error-message config-file (PRIMOS)

A file system error was encountered by the preloader while attempting to open the configuration file config-file.

- file-system-error-message CAN'T ATTACH TO CMDNCO (AINIT)

A file system error was encountered while attempting to attach to CMDNCO for User 1.

- file-system-error-message CMDNCO (PRIMOS)

A file system error was encountered by the preloader while attempting to attach to CMDNCO.

- file-system-error-message C_PRMO (PRIMOS)

A file system error (other than FILE NOT FOUND) was encountered by the preloader while attempting to open the file C_PRMO for command input.

- file-system-error-message PRnnnn (PRIMOS)

A file system error was encountered by the preloader while attempting to open or read the indicated PRnnnn file.

- BAD directive PARAMETER (AINIT)

One or more of the parameters specified for the configuration directive is invalid.

- INVALID CONFIG COMMAND: directive (AINIT)

The directive in the configuration data file is not a recognized configuration directive.

CONFIG Error Messages

- BAD AMLIBL PARAMETER (CINIT)

The DMC input buffer size is too small. Minimum value specified must be greater than '20, if not specified as 0. 19.0

- BAD DMQ AMLC CONFIGURATION (CINIT)

A DMQ buffer size in an AMLBUF directive is not equal to a power of 2.

- BAD LINE # IN AMLBUF CMND (CINIT)

An AMLBUF directive specifies a line number less than 0 or greater than the number of lines configured for the system. Also given if the AMLBUF directive is used to try to modify the size of the input or output terminal buffers for remote users. 19.0

- BAD LINE # IN ASRBUF CMND (CINIT)

An ASRBUF directive specifies an invalid line number. The only valid line number is 0. 19.0

- BAD RECORD ADDRESS IS LESS THAN 16. (BADSP$)

A bad record is found to have an address less than or equal to 16. The first 16 records of a partition contain the MFD, BOOT, and other special files.

- BAD SMLC CONTROLLER MAPPING COMMAND (CINIT)

An SMLC controller mapping directive specifies an invalid or missing controller number. The correct numbers are either '0' or '1'.

- BAD SMLC DATASET PROCEDURE: n (CINIT)

An SMLC DSC directive specifies an invalid or missing dataset procedure of n. Correct values are '1', '2', or '3'.

- BAD SMLC DATASET STRAPPING ORDER: n (CINIT)

An SMLC DSC directive specifies an invalid or missing dataset strapping order of n.

- BAD SMLC LINE MAPPING COMMAND (CINIT)

An SMLC line mapping directive specifies an invalid or missing line number.

- BAD SMLC RECEIVER ON/OFF CONTROL: n (CINIT)

An SMLC DSC directive specifies an invalid or missing receiver on/off control value of n. Correct values are '0' or '1'.

- END OF FILE. MISSING 'GO' CMND (PRIMOS)

The configuration data file does not include the required GO directive.

- FILUNT INVALID (AINIT)

The FILUNT directive specifies incorrect information for proper configuration.

- FIRST COMMAND MUST BE CONFIG

The command typed in response to the 'PLEASE ENTER CONFIG' prompt or the first executable command in C_PRMO was not a CONFIG command. It must be one of these. (Pre-loader)

- ILLEGAL ALTDEV

The device specified as the alternate paging device is not a legal physical device number. (Pre-loader)

- ILLEGAL COMDEV

The device specified for the command device is not a legal physical device number. (Pre-loader)

- ILLEGAL PAGDEV

The device specified for the paging device is not a legal physical device number. (Pre-loader)

- INCORRECT 'BADSPT' FILE FORMAT (BADSP$)

The file BADSPT has an incorrect file format. For example, the starting address (SA) of the saved format file is not equal to '1000.

- INPUT BUFFERS TOO LARGE (AMINIT)

The DMC buffer size is too large. Reconfigure within the permissable range using the AMLIBL directive.

- LOGREC CONFIG DIRECTIVE NO LONGER SETS A QUOTA ON THE SYSTEM EVENT LOGGING FILE. PLEASE USE 'SET_QUOTA'. (CINIT) 19.0

Prior to Rev. 19, the system event logging file could be given a quota size using the LOGREC configuration directive. At Rev. 19 the size of this file can be controlled by setting a quota on its directory with the SET_QUOTA command. See Chapter 6 for a detailed description of quotas.

- MISSING NIUSR, PAGDEV, OR COMDEV

The configuration data file did not specify these required parameters. (Pre-loader)

- NETREC CONFIG DIRECTIVE NO LONGER SETS A QUOTA ON THE NETWORK EVENT LOGGING FILE. PLEASE USE 'SET_QUOTA'. (CINIT)

Prior to Rev. 19, the network event logging file could be given a quota size using the NETREC configuration directive. At Rev. 19 the size of this file can be controlled by setting a quota on its directory with the SET_QUOTA command. See Chapter 6 for a detailed description of quotas. 19.0

- NO ROOM. AMLBUF (TFLADJ)

The total size of the terminal I/O buffers (AMLBUF directives and ASRBUF) exceeds 256K bytes (2 segments).

- NRUSR INVALID (AINIT)

The number of remote users specified by an NRUSR directive exceeds the maximum number of configurable remote users ('40, decimal 32).

- NTUSR+NAMLC TOO BIG

The number of terminal and assigned buffers exceeds the maximum number of configurable buffers (128, '200).

19.0

- NTUSR+NPUSR+NRUSR+NSLUSR TOO BIG (AINIT)

The sum of terminal, phantom, remote, and slave users exceeds the maximum number of configurable users (128, '200).

- RAT UNREADABLE ON ALTDEV (AINIT)

The record that contains the DSKRAT on ALTDEV is not readable.

- RAT UNREADABLE ON COMDEV (AINIT)

The record that contains the DSKRAT on COMDEV is not readable.

- RAT UNREADABLE ON PAGDEV (AINIT)

The record that contains the DSKRAT on PAGDEV is not readable.

- RESTART PLEASE

This message appears following any error message printed by the PRIMOS initialization logic (AINIT). The system will halt at location BOOT0_ in segment 6. PRIMOS II must be reloaded. The offending directive in the configuration data file must be corrected.

- SEEK FAILURE ON ALTDEV (AINIT)

The initial seek to cylinder 0 on the alternate paging device failed.

- SEEK FAILURE ON PAGDEV (AINIT)

The initial seek to cylinder 0 on the primary paging device failed.

- SMLC CTRLR # OUT OF RANGE (AINIT)

An SMLC directive specifies an invalid controller number.

- SMLC LINE # OUT OF RANGE (AINIT)

An SMLC directive specifies an invalid line number.

- SUM OF BAD SPOTS ON THE PRIMARY AND ALTERNATE PAGING DEVICE EXCEEDS 16

PRIMOS supports a maximum of 16 defective tracks (bad spots) on both primary and alternate paging partitions.

- TOO MANY BAD SPOTS IN 'BADSPT' (BADSP$)

Number of bad spots entries in the file BADSPT exceeds 16. PRIMOS supports a maximum of 16 bad spots in both primary and alternate paging partitions.

- TPIOS ERROR

An I/O error occurred while preloading the paging device. (Pre-loader)

- UNABLE TO INITIALIZE DMQ AMLC (AINIT)

The total size of the DMQ buffer sizes specified exceeds 64K words.

- USE physical-device FOR PAGING?

The disk physical-device has been specified as the paging device, but is formatted as a standard PRIMOS disk. A reply of YES is required to enable paging on physical-device. (Pre-loader)

NETWORK INITIALIZATION ERROR MESSAGES

Network errors no longer cause system configuration to terminate.

- file-system-error-message Can't attach to PRIMENET*

Be sure that the PRIMENET* UFD exists on the command device and that user SYSTEM has UR access rights. 19.0

- file-system-error-message Can't start network

Be sure that user NETMAN has ALL access rights to the PRIMENET* UFD and to the file NETWORK_SERVER.COMI.

- file-system-error-message CAN'T START SLAVE (BINIT)

Slaves for FAM II could not be started. The particular file system error message gives more details.

- file-system-error-message CMDNC0
  NETWORK NOT CONFIGURED (NETFIG)

19.0

A file system error has occurred while attaching to CMDNC0 to read the network configuration file.

- file-system-error-message NETCON
  NETWORK NOT CONFIGURED (NETFIG)

A file system error has occurred while opening or reading the network configuration file.

- BAD KEY FPID (Network Block Init)
- BAD KEY HDXQ (Network Block Init)
- BAD KEY MYQ (Network Block Init)
- BAD KEY RNGQ (Network Block Init)
- BAD KEY SLCQ (Network Block Init)
- BAD KEY VCBQ (Network Block Init)

These are internal network errors that do not cause system configuration to terminate.

- BAD NETCON FILE (NETFIG)

The network configuration file has an illegal or obsolete format. Recreate the network configuration file using the most recent version of NETCFG.

19.0

- BAD PARAMETER FPID (Network Block Init)
- BAD PARAMETER HDXQ (Network Block Init)
- BAD PARAMETER MYQ (Network Block Init)
- BAD PARAMETER RNGQ (Network Block Init)
- BAD PARAMETER SLCQ (Network Block Init)
- BAD PARAMETER VCBQ (Network Block Init)

These are internal network errors that do not cause system configuration to terminate.

- IPC NOT SUPPORTED (NETFIG)

IPC networking is obsolete and no longer supported.

- Network Server Logged out during network startup.

This error indicates either that the necessary files do not exist in the PRIMENET* UFD, or that NETMAN does not have ALL access to them. (See Chapters 5 and 18.)

- NO MORE ROOM FOR NETWORK TABLES (NETFIG)

The network configuration requires more table space than exists in the operating system. These errors can only be generated if the -NOCHECK option was used with the NETCFG command.

- No phantoms configured. Can't start network process. (BINIT)

At least one phantom process (user) must be configured exclusively for the use of the network server.

- NO ROOM FPID (Network Block Init)
- NO ROOM HDXQ (Network Block Init)
- NO ROOM MYQ (Network Block Init)
- NO ROOM RNGQ (Network Block Init)
- NO ROOM SLCQ (Network Block Init)
- NO ROOM VCBQ (Network Block Init)

19.0

These are internal network errors that do not cause system configuration to terminate.

- NOT FOUND. SLAVE.COMI; CAN'T START SLAVE (BINIT)

Slaves for FAM II could not be started because the file SLAVE.COMI was not found in UFD PRIMENET*.

- TOO MANY HDX NODES (NETFIG)
- TOO MANY HOSTS (NETFIG)
- TOO MANY NAMES (NETFIG)
- TOO MANY PATHS (NETFIG)
- TOO MANY RING NODES (NETFIG)
- TOO MANY SMLC'S (NETFIG)

The network configuration requires more table space than exists in the operating system. These errors can only be generated if the -NOCHECK option was used with the NETCFG command.

- U$NPX set for wrong process. (BINIT)

This is an internal PRIMOS error that indicates problems with the process allocation tables. FAM II slaves could not be started.

SINGLE-LINE CONFIG COMMAND

The single-line CONFIG command (rather than the configuration data file) can still be used to configure the system. However, the single-line command cannot specify as many features as the configuration data file directives. Users are urged to convert to the data file method of configuring the system.

The command format is:

CONFIG ntusr pagdev comdev [maxpag] [altdev] [namlc]
[npusr] [nrusr] [smlc]

ntusr — An octal integer defining the number of terminal users, including the supervisor (for example, for a 4-user system, ntusr = 5; for a 7-user system, ntusr = 10).

pagdev — A physical device number specifying the device to be used for paging.

comdev — An argument that specifies the physical device number initially assigned as logical device 0. This is the device on which UFD CMDNCO is searched when a user invokes an external PRIMOS command. If comdev and pagdev are the same, the disk is considered to be split into a file system and a paging part. The boundary between the partitions is defined by the DSKRAT header, and it may be set by the MAKE program.

maxpag — An optional argument defining available physical memory storage. It corresponds to the last sector number (octal) to be used.

altdev — CONFIG may specify either one or two disk devices on which paging is to take place. The alternate paging device cannot be a split disk.

namlc — An optional argument defining the number of assignable AMLC lines.

npusr — An optional argument defining the number of phantom users.

nrusr — An optional argument specifying the number of remote users who can run on the system.

smlc — An optional argument enabling the SMLC.

For example, the one-line CONFIG command:

```
CONFIG 60 100460 460 1000 6/4
```

specifies a system with 48 ('60) terminals and 4 (6/4) phantom users. The paging device (100460) and the command device (460) are partitions of a storage module on drive unit 0. The value of MAXPAG ('1000) specifies '400K (256K) of available physical memory storage.

The "6/" prefix is a feature of the command line that allows the omission of arguments. In this case, arguments 4 and 5 have been left at the default values. (The first argument, 60, is argument 0.)

The arguments, with their numerical values, are:

0/ntusr	Number of terminal users
1/pagdev	Paging device
2/comdev	Command device
3/maxpag	Number pages physical memory to use
4/altdev	Alternate paging device
5/namlc	Number assignable AMLC lines
6/npusr	Number phantom users
7/nrusr	Number remote users
10/smlcon	Non-zero value enables SMLC

Note

All numerical values in the command line are octal unless otherwise specified.

4
Using EDIT_PROFILE

INTRODUCTION

This chapter explains how you use the EDIT_PROFILE command. You should not use EDIT_PROFILE until you have read Chapter 1, which discusses the planning you need to do before you use the command. You may also want to read Chapter 15, which discusses security for your system as a whole. EDIT_PROFILE allows you to specify very precisely what degree of security you want, so you must plan your profiles before you create them.

The System Administrator may use EDIT_PROFILE in any of its three modes. These modes are as follows:

- Initialization Mode allows you to create the System Administration Directory (SAD), in which all user profile information is stored. You use this mode to create a new Rev. 19 system, when you boot it, or to create a new SAD.

- System Administrator Mode allows you to create, maintain, and delete profiles for users and groups of users, once the SAD exists.

- Project Administrator Mode allows you to offload some routine administrative work onto Project Administrators, who use EDIT_PROFILE in this mode to perform chores related to their particular projects.

INITIALIZATION MODE

In initialization mode, EDIT_PROFILE leads you through the steps necessary to create the user profile data base, asking you to define the characteristics you want for your system.

Entering Initialization Mode

To enter initialization mode, you give the EDIT_PROFILE command. The SAD that controls access to your system must be stored in the MFD; however, you may also create SADs elsewhere.

To create a new SAD in the MFD, give the command:

EDIT_PROFILE [{-MFD_PASSWD / -MPW} password]

without an argument. If you are creating the SAD in a password-protected MFD, you must specify the owner password for the MFD, using the -MFD_PASSWD option. (XXXXXX is the password as it arrives on the master disk.)

While there is no SAD, no users can log in. This means that you have to run EDIT_PROFILE from the supervisor terminal.

However, if you want to create a SAD elsewhere, and a SAD has already been created in the MFD, you can issue the command from any terminal. To do this, give the command in the form:

```
EDIT_PROFILE pathname
```

In this form of the command, pathname is the full pathname of the parent directory of the new SAD. For example, to create a SAD on the disk SEA, in the subdirectory CHANNEL of the top-level directory ENGLISH, you would give the command as follows:

```
EDIT_PROFILE <SEA>ENGLISH>CHANNEL
```

When you create a SAD in a parent directory other than the MFD, this parent directory must be an ACL directory. (In the example, CHANNEL must be an ACL directory.)

To create a SAD in the ACL directory to which you are currently attached, you can give the command in the form:

```
EDIT_PROFILE *
```

EDIT_PROFILE then goes into initialization mode, provided that no SAD exists in the directory where you are creating one. It responds as follows:

```
Profile editor [rev. 19.0] in initialization mode DD MMM YY HH:MM:SS
```

The figures at the end of the line show the current date and time. EDIT_PROFILE then begins an initialization dialog.

Initialization Dialog

During initialization, EDIT_PROFILE may ask you questions and prompt you for information as follows.

1. SAD does not exist. Create it?

   Answer YES. A NO terminates EDIT_PROFILE.

1a. Do you want to convert the MFD to an ACL directory?

   Answer YES to enable the use of projects and groups.

   Answer NO if you do not want to use ACLs, or projects other than the system default project.

   (This question is asked only in a password-protected MFD.)

2. Do you want SYSTEM-wide groups, PROJECT-based groups or BOTH?

   Answer BOTH for greatest flexibility. This allows you to create and use both system-wide and project-specific access groups on your system.

   Answer SYSTEM if you want only system-wide access groups, PROJECT if you want only project-specific access groups. The criteria for these choices are explored in Chapter 1 of this Guide.

   Whatever your answer, you can use the System Administrator mode subcommands SET_SYSTEM_GROUPS and SET_PROJECT_GROUPS to enable or disable groups at the system or project levels if you want to change this specification later on. See the discussion of System Administrator mode for descriptions of these commands.

   (This question is not asked in a password-protected MFD.)

3. Projected number of users:

   Answer with the total number of users you expect to be using your system. EDIT_PROFILE always allows space for at least 20 users. The default reply to this question is also 20, so that if you expect fewer than 20 users, you can simply enter a blank or a carriage return. EDIT_PROFILE is most efficient with 5000 or fewer user profiles.

   In fact, 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 you add many more users later on, and space gets short, EDIT_PROFILE warns you with the following message:

Warning: User Validation file is overloaded.

and you can then use the System Administrator mode subcommand REBUILD to rebuild the data base. REBUILD is discussed in the section of this chapter describing System Administrator mode.

4. System administrator name:

The answer SYSTEM allows a user at the supervisor terminal to run EDIT_PROFILE. Any other name prevents this; the Administrator will have to log in.

This question is asked only when you are creating a SAD from the supervisor terminal. Otherwise, EDIT_PROFILE automatically enters the name of the person creating the SAD as System Administrator.

5. Create project "DEFAULT"?

This is the only time you can create the system default project.

Answer YES to create the project, which is always called "DEFAULT". You can delete the project later, in System Administrator mode, if you wish to do so.

Answer NO if you are sure that you will never want a system default project on your system. If you do answer no, you must create at least one project on your system. No users will be able to log in until you have done this. (To create a project other than DEFAULT, you use the System Administrator mode command ADD_PROJECT, described later in this chapter.)

A system default project is useful in three situations:

- If you are not going to create separate projects on your system, you must create the default project, to which all system users will belong. While DEFAULT is the only project on your system, any users you register for your system will automatically be added to that project.

- If you expect that any users of your system will not belong to a specific project, you will want the default project to provide a "catch-all" for them.

- If you prefer not to be asked to specify a default login project for each user whom you add to the system, you should create DEFAULT to take care of this for you.

6. Set system-wide attributes for user "user-id":

   (The "user-id" EDIT_PROFILE displays in this question is the System Administrator's, given in answer to question 4.)

   EDIT_PROFILE prompts you to enter the System Administrator's password. Although a null password is acceptable, the System Administrator should not use a null password, for security reasons.

   EDIT_PROFILE then prompts you to enter the names of the System Administrator's system-wide access groups, if you allowed system-wide groups in your answer to question 2.

   If you have chosen not to create the DEFAULT project, EDIT_PROFILE also asks you to specify a default login project for the System Administrator. You may omit this if you want to.

   At this stage, EDIT_PROFILE creates the User Validation, Master Project, and Master Group files, and notifies you that it has done so. If you have chosen to create the DEFAULT project, EDIT_PROFILE now asks you to define it. If not, initialization is complete, but you must add at least one project to your system before anyone can log in.

7. Set limits for project "DEFAULT":

   EDIT_PROFILE prompts you to enter the names of all the access groups that may later be associated with system project "DEFAULT". The term limits indicates that these are all the groups that may at some time be used in the DEFAULT project. The System Administrator can change project limits later on, using the CHANGE_PROJECT command in System Administrator mode, but a Project Administrator cannot change project limits.

8. Set attributes for user "user-id" in project "DEFAULT":

   (Again, the "user-id" displayed is the System Administrator's.)

   Next, you define the project-based attributes for the System Administrator when using the DEFAULT project. First, if you enabled project-based groups in answer to question 2, you enter the ids of the project-based groups to which the System Administrator will belong. Second, you specify his or her Initial Attach Point, if any, in the project.

9. Set profile attributes for project "DEFAULT":

   EDIT_PROFILE prompts you to specify the project-based access groups specific to project DEFAULT. This question is asked only if you enabled project-based groups in answer to question 2. The groups you specify must have been included in your list of groups supplied as project limits in answer to question 7.

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. It also depends on where you create the SAD. The following examples illustrate these different dialogs.

Initializing an ACL System

The following example of EDIT_PROFILE in Initialization mode shows how to create a SAD for a system using ACLs, projects, and groups. In this example, the administrator is working in a password-protected MFD, and converts it to an ACL directory. EDIT_PROFILE therefore asks what types of groups the administrator wants.

```
OK, EDIT_PROFILE -MFD_PASSWD XXXXXX
Profile editor [rev 19.0] in initialization mode 18 May 82 11:12:40.
SAD does not exist.  Create it?  YES
Do you want to convert the MFD to an ACL directory?  YES
Do you want SYSTEM-wide groups, PROJECT-based groups, or BOTH?  BOTH
*** Creating User Validation File.  Projected number of users: 30
System administrator name:  SYSTEM

Create project "DEFAULT"?  YES

Set system-wide attributes for user "SYSTEM":
    Password:  ME
    Groups:  .WRITERS .ADMIN .ENG .TYPISTS
*** New group added to system: ".WRITERS".
*** New group added to system: ".ADMIN".
*** New group added to system: ".ENG".
*** New group added to system: ".TYPISTS".

User Validation file created 18 May 82 11:14:32
    44 entries in prime area; file is 3 records long.

Master Project File created 18 May 82 11:14:32

Master Group File created 18 May 82 11:14:32

Set limits for project "DEFAULT":
    Groups:  .TYPISTS .ADMIN

Set attributes for user "SYSTEM" in project "DEFAULT":
    Groups:  .TYPISTS
    Initial attach point:  <MARKET>BOOKS

Set profile attributes for project "DEFAULT":
    Groups:  .TYPISTS
*** New group added to project: ".TYPISTS".
Project "DEFAULT" created.
    44 entries in prime area; file is 3 records long.
Check entry?  YES
```

```
**************************************************************************
Project: DEFAULT                          Administrator: SYSTEM
    One entry in use out of 44.

Master project limits:
    Groups: .TYPISTS

Project profile:
    Groups: .TYPISTS
    Initial attach point: <MARKET>BOOKS
**************************************************************************

>
```

Initializing a Non-ACL System

On a system which does not use ACLs, you cannot create any project except DEFAULT, and the System Administrator has to administer that project. Without ACLs, you cannot use access groups, so EDIT_PROFILE does not ask you any group-related questions during initialization.

EDIT_PROFILE only works correctly when the SAD has a null owner password. Security of the user profile data base is much less complete on systems which do not use ACLs.

In this example, the administrator chooses not to use ACLs. Project DEFAULT is therefore created automatically. Fewer than 20 people are expected to use the system, so the administrator presses the carriage return when asked to enter the projected number of users. The administrator then defines his own characteristics in project DEFAULT, and the attributes of the project itself.

```
OK, EDIT_PROFILE -MFD_PASSWD XXXXXX
Profile editor [rev. 19.0] in initialization mode 07 Aug 82 20:43:56
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.
*** Creating User Validation File.  Projected number of users:
System administrator name: SYSTEM

Set system-wide attributes for user "SYSTEM":
    Password: ME

User Validation File created 07 Aug 82 20:45:32
    20 entries in prime area; file is 1 record long.

Master Project File created 07 Aug 82 20:45:32

*** Creating project "DEFAULT"

Set attributes for user "SYSTEM" in project "DEFAULT":
```

```
    Initial attach point: <KATIE>MILD>ABRAMS

Set profile attributes for project "DEFAULT":
    Initial attach point: <KATIE>MILD
Project "DEFAULT" created.
    20 entries in prime area; file is 1 record long.
Check entry? YES

*************************************************************************
Project: DEFAULT
    One entry in use out of 20.

Project profile:
    Initial attach point: <KATIE>MFD
*************************************************************************
Change entry? NO
```

Creating a SAD outside the MFD

EDIT_PROFILE allows you to create a SAD in any ACL-protected directory that does not already contain one. This is useful for two reasons:

- For testing purposes, you can create a new SAD without disrupting other users of your system. In fact, you need not do this yourself. You can delegate the task to someone else, and check that it has been done properly before using it as a "live" SAD in the MFD. You can also use this method to practice using EDIT_PROFILE.

- 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.

In the following example, a user creates a SAD in her current directory. She chooses to enable only system-wide groups, and specifies 5000 users. Her user-id is JUNE, and EDIT_PROFILE enters this id as the System Administrator's. (EDIT_PROFILE does not ask for the System Administrator's id because the SAD is not being created in the MFD.)

June chooses not to create project DEFAULT. This means that she will have to use the ADD_PROJECT command in System Administrator mode, to create at least one project on the system.

EDIT_PROFILE then asks her to define the system-wide attributes for the System Administrator. She gives the password as JULY, and specifies that the administrator will belong to a system-wide group called .ADMINISTRATORS.

Because June did not create project DEFAULT, EDIT_PROFILE then asks if the administrator will belong to a default login project. June does not specify one.

EDIT_PROFILE then creates the validation, group, and project files, and initialization is complete, as shown by the prompt >.

```
OK, EDIT_PROFILE *
Profile editor [rev 19.0] in initialization mode 17 Jun 82 11:31:32.
SAD does not exist.  Create it? YES
Do you want SYSTEM-wide groups, PROJECT-based groups, or BOTH? SYSTEM
***Creating User Validation File.  Projected number of users: 5000
System administrator = "JUNE".

Create project "DEFAULT"? NO

Set system-wide attributes for user "JUNE":
    Password: JULY
    Groups: .ADMINISTRATORS
*** New group added to system: ".ADMINISTRATORS".
    Default login project:

User Validation File created 17 Jun 82 11:33:48
    7516 entries in prime area; file is 353 records long.

Master Project File created 17 Jun 82 11:33:48

Master Group File created 17 Jun 82 11:33:48

>
```

Now that EDIT_PROFILE is in System Administrator mode, June uses the HELP subcommand to remind herself of the format of the ADD_PROJECT command, and then uses the command to create the first project on the system, without which users will not be able to log in. EDIT_PROFILE asks her if she wants to check the entry; she does so, and then quits by typing the letter q.

```
> HELP AP
Add_Project [<project_id> [-PA <pa_name>] [-CReate_pa]
             [-LIKE <like_reference>] [-PROFile]
             [-SIZE <entry_count>] [-No_Query]]
> ADD_PROJECT BAG -PA JUNE -CR -PROFILE -SIZE 500

Set attributes for user "JUNE" in project "BAG":
    Initial attach point: <TEST>BAG>JUNE

Set profile attributes for project "BAG":
    Initial attach point: <TEST>BAG
Project "BAG" created.
    772 entries in prime area; file is 37 records long.
Check entry? YES
```

```
**************************************************************************
Project: BAG                                  Administrator: JUNE
    One entry in use out of 772.

Project profile:
    Initial attach point: <TEST>BAG
**************************************************************************
Change entry? NO
> Q
```

Leaving Initialization Mode

After you have answered all the questions posed by EDIT_PROFILE, it prompts you to enter a command. (The prompt is a right-angle bracket, >, as illustrated in the previous example.)

This prompt tells you that the user profile data base has been initialized, and that you are now in System Administrator mode, described in the following section of this chapter.

When you are ready to quit, you do so by typing QUIT (or Q) in response to the prompt.

When EDIT_PROFILE initialization is complete, a SAD containing the Master Group File, Master Project File and User Validation File has been created.

Care of your SAD

In systems using ACLs, EDIT_PROFILE automatically generates the ACL protecting the MFD, if the MFD is not ACL-protected already. The System Administrator is given ALL rights. Everybody else (identified as $REST) is given only List (L) and Use (U) rights.

It is extremely important that you and anyone else acting as System Administrator observe the following rules:

- Do not alter the ACLs protecting the SAD or its contents. Any change in the ACL may allow breaches in the security of your system, or cause problems for EDIT_PROFILE.
- Do not alter the read/write locks protecting the contents of the SAD.
- Do not try to copy individual parts of the SAD.
- Keep a copy of your SAD in case it gets damaged. A copy of the SAD on a back-up disk would serve the purpose.

If the ACLs on the SAD or its contents, or the read/write locks on its contents are altered, you should restore them to their original condition, using the SET_DEFAULT_PROTECTION command in System Administrator mode, described below.

Copying your SAD: If you need to create a copy of the SAD, you must copy its entire contents, using the -COPY_ALL option of the COPY command.

SYSTEM ADMINISTRATOR MODE

Once you have initialized the user profile data base, you use EDIT_PROFILE in System Admininistrator mode. In this mode, you can use EDIT_PROFILE's subcommands to add, change, and delete attributes of users and projects.

Table 4-1 shows the subcommands 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 subcommands can be divided into three sets:

- System-level Commands provide control of the system as a whole. Using these commands, the System Administrator can enforce system requirements for the handling of passwords, maintain system-wide access groups, and do other system-related jobs.

- Project-level Commands provide control of all the projects on the system, including project DEFAULT, usually managed by the System Administrator. The System Administrator is the only person who can add or delete projects, and set project groups, 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, however, 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 can specify a Project Administrator to manage the project. That person can then use EDIT_PROFILE in Project Administrator mode, discussed later in this chapter. The System Administrator can administer all projects, whoever the Project Administrator is.

The following sections explain how to use each of EDIT_PROFILE's subcommands.

Table 4-1
EDIT_PROFILE Subcommands

Command	Used by	Function
System Command		
CHANGE_SYSTEM_ADMINISTRATOR	SA	Changes id of SA
FORCE_PASSWORD	SA	Disallows passwords on login line
HELP	SA or PA	Displays EDIT_PROFILE information
LIST_SYSTEM	SA	Displays System and other attributes
NO_NULL_PASSWORD	SA	Disallows use of null passwords
REBUILD	SA or PA	Rebuilds validation files
SET_DEFAULT_PROTECTION	SA	Restores protection to SAD
SET_SYSTEM_GROUP	SA	Enables or disables system groups
Project Command		
ADD_PROJECT	SA	Creates a new project
ATTACH_PROJECT	SA or PA	Specifies a "current project"
CHANGE_PROJECT	SA or PA	Changes a project's attributes
DELETE_PROJECT	SA	Removes a project from your system
DETACH_PROJECT	SA or PA	Detaches the "current project"
LIST_PROJECT	SA or PA	Lists attributes of a project
SET_PROJECT_GROUPS	SA	Enables or disables project groups
User Control Command		
ADD_USER	SA or PA	Adds user to system or projects
CHANGE_USER	SA or PA	Changes a user's system or project attributes
DELETE_USER	SA or PA	Removes user from system or project
LIST_USER	SA or PA	Lists user's system or project attributes
VERIFY_USER	SA	Checks existence of user-id on network systems

SYSTEM-LEVEL COMMANDS

The CHANGE_SYSTEM ADMINISTRATOR Command

You use this command to change the user-id of the System Administrator. This may be necessary if a different person is going to take over the job of 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.

After you have entered the user-id of the System Administrator during initialization, you cannot change the System Administrator until you have re-booted the system, because PRIMOS reads the System Administrator's id only when the system is booted, and won't let the System Administrator be changed unless it recognizes the old administrator making the change.

The format of the command is as follows:

```
{CHANGE_SYSTEM_ADMINISTRATOR}  [user-id] [-ALL]
{CSA                        }
```

The user-id identifies the new System Administrator. If you do not specify it, EDIT_PROFILE prompts you to enter it.

The -ALL option makes the new System Administrator the Project Administrator of any projects administered by the previous System Administrator. -ALL is assumed if your only project is DEFAULT.

After you give the command, EDIT_PROFILE asks you to confirm that you really meant it. If you reply yes (or y), the System Administrator is changed.

When this happens, EDIT_PROFILE changes all the ACLs protecting the SAD and its subdirectories to reflect the new System Administrator's user-id. It does this from scratch, so that if you changed any of these ACLs, the changes are lost. (As noted earlier in this chapter, it is not advisable to alter these ACLs in any case.)

EDIT_PROFILE automatically terminates after the System Administrator has been changed.

The FORCE_PASSWORD Command

You use this command to prevent users from entering their passwords on the same line as the LOGIN command. This means that a user has to wait for a system prompt before typing a login password, and that the password is not echoed at the user's terminal. This prevents passwords from being seen by unauthorized people.

The format of the command is as follows:

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

The -ON option forces password prompts. This option is the default. The -OFF option allows passwords on the LOGIN line.

See also: NO_NULL_PASSWORD

The HELP Command

You use HELP to display the arguments, options, and option arguments, for one or all EDIT_PROFILE subcommands. The format of the command is:

```
HELP [command-name]
```

where command-name identifies an EDIT_PROFILE subcommand. If you specify a command name, EDIT_PROFILE displays the format of that command, showing its argument, if any, and all its options with their arguments, if any.

If you don't specify a command name, EDIT_PROFILE lists all subcommands, with the main argument and options for each of them, as illustrated in the following example.

```
> 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    Option
------------                    --------    ------

Add_Project                     project     -PA, -CReate_pa, -SIZE
                                            -No_Query, -LIKE
Add_User                        user        -LIKE, -PROJect, -PROFile, -No_Query
                                            -SYStem, -DeFauLT
                                            -PassWord, -Verify_NS
ATtach_Project                  project     none
Change_Project                  project     -PROFile, -SIZE, -LIST
                                            -PA, -LIMits
Change_System_Administrator
                                SA name     -ALL
Change_User                     user        -PROJect -LIST
                                            -SYStem -PassWord
Delete_Project                  project     none
```

Delete_User	user	-PROJect
DeTach_Project	project	none
Force_PassWord	none	-ON, -OFF
HELP	command	none
List_Project	project	-PROFile, -USER, -ALL -OUTput, -TTY, -APPend
List_System	none	-USers, -GRoups, -PROJects, -ALL -OUTput, -TTY, -APPend -DETail
List_User	user	-PROJect, -ALL
No_Null_PassWord	none	-ON, -OFF
REBuild	none	-PROJect, -SIZE
Set_Default_PRotection	none	-CoNVert
Set_Project_Groups	none	-ON, -OFF
Set_System_Groups	none	-ON, -OFF
Verify_User	user	-ALL

The LIST_SYSTEM Command

You can use this command to display system, group, project, and user attributes, depending on the options you specify. Among system attributes displayed may be the following:

- SAD not ACL protected
- System-wide groups enabled (only on ACL systems)
- Project-based groups enabled (only on ACL systems)
- Non-DEFAULT projects in use (only on ACL systems)
- Passwords always requested at login (FORCE_PASSWORD has been used)
- Null passwords not allowed (NO_NULL_PASSWORD has been used)

The format of the command is as follows:

```
{LIST_SYSTEM}
{LS         }  [options]
```

The options for the LIST_SYSTEM command are as follows:

Option	Meaning
-USERS	Lists system-wide attributes of all system users.
-GROUPS	Lists all groups on the system.

-PROJECTS	Lists all projects on the system, with their attributes.
-ALL	Lists all the information provided by the combination of the -USERS, -GROUPS, and -PROJECTS options.
-DETAIL	Lists additional information depending on the other options you select. With the -DETAIL option, the -USERS list includes the list of projects to which each user belongs; the -GROUPS list shows the membership of users and projects in each of the groups; and the -PROJECTS list shows which users and groups belong to each of the projects.
-OUTPUT pathname	Directs the output of the command into the file you specify with pathname. If you specify a simple filename rather than the full pathname, the file is opened in the SAD.
-APPEND	Adds output of the command at the end of the file you specified with the -OUTPUT option. You use this option only in conjunction with the -OUTPUT option. Unless you use -APPEND, any contents of the specified output file are overwritten.
-TTY	Directs output of the command to your terminal. Output is displayed at your terminal by default. Therefore, you need only use -TTY when you want to check output at your terminal, and have redirected output to a file with the -OUTPUT option.

If you give the command without any options, EDIT_PROFILE displays the System Administrator's id and a summary of system attributes, as shown in the following example.

```
Profile editor [rev 19.0] in system administrator mode 19 Mar 82 15:15:
56.
> LIST_SYSTEM
*********************************************************************
System                          Administrator: SYS_ADMIN
    45 entries in use out of 68 in prime area; 2 in overflow area.
    System-wide groups enabled.
    Non-DEFAULT projects in use.
    Passwords always requested at login.
*********************************************************************
> QUIT
```

The NO_NULL_PASSWORD Command

You use the NO_NULL_PASSWORD command either to disallow the use of null passwords on your system, or to allow their use.

Disallowing null passwords improves system security, since it ensures that user-ids on your system cannot be compromised by simple trial-and-error methods. After you have disallowed 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.

The format of the command is as follows:

```
{NO_NULL_PASSWORD}  [-ON  ]
{NNPW            }  [-OFF ]
```

You use the -ON option (which is the default) to disallow null passwords. If any users of your system have null passwords when you issue the command, EDIT_PROFILE displays a list of these users, so that you can assign passwords to them.

You use the -OFF option to allow the use of null passwords. PRIMOS does allow their use unless you explicitly forbid it using the -ON option.

See also: FORCE_PASSWORD.

The REBUILD Command

You use this command to rebuild the user profile data base, at the level either of the whole system or of an individual project. You may want to do this for the following reasons:

- If you have added many users to the system or to a particular project, EDIT_PROFILE issues a warning message indicating that a file is overloaded, which means you should rebuild it.
- If you expect to add many users, you may want to rebuild in anticipation of the increase.
- If you want the user profile data base to be cleaned up, REBUILD accomplishes this for you by removing any "dead" entries. For example, suppose you used project-based groups on your system, and then disabled them (with the -OFF option to the SET_PROJECT_GROUPS command). Even though the groups aren't used, they are not actually removed until you issue a REBUILD command.
- If you need to conserve disk space, REBUILD allows you to do so, both by cleaning up redundant material, and by allowing you to specify the size of files.

Note

Never use REBUILD while users can log in to your system. MAXUSR 0 should be in effect before you give the REBUILD command.

The format of the command is as follows:

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

You use the -PROJECT option to rebuild files related to an individual project. If you leave out the project-id, EDIT_PROFILE assumes your current project (see the ATTACH_PROJECT command), unless you have none. In this case, it prompts you to specify a project-id.

If you don't use the -PROJECT option, EDIT_PROFILE rebuilds the user profile data base for the whole system. When you use REBUILD on a system with only one project, EDIT_PROFILE automatically rebuilds the project-related files every time you rebuild the system-related files.

You use the -SIZE option to specify how many users you expect to need space for, either in the system or the project-related data base. EDIT_PROFILE always allows space for at least 20 users, both for the system and for each project, and can accommodate up to 2000 user profiles.

If you leave out the -SIZE option, EDIT_PROFILE expands the system or project validation file to the next size up from its current size.

The following example shows an administrator rebuilding the entire user profile data base. She gives the command without any options, allowing EDIT_PROFILE to select the new size of the user validation file.

```
Profile editor [rev 19.0] in system administrator mode 26 Aug 82 09:02:
52.
> rebuild                          /* SA wants to clean up UVF

*** UVF backed up into file "UVF.OLD" 26 Aug 80 09:02:56
*** MPF backed up into file "MPF.OLD" 26 Aug 80 09:03:00
*** MGF backed up into file "MGF.OLD" 26 Aug 80 09:03:04
*** MPP for project "DEFAULT" backed up into
    file "DEFAULT>BACKUP>MPP" 26 Aug 80 09:03:08.
*** MPP for project "EDUCATION" backed up into
    file "EDUCATION>BACKUP>MPP" 26 Aug 80 09:03:16.

*** Rebuild complete 26 Aug 80 09:04:00! ***

Delete old files? yes                    /* Trusting soul, that SA!

> quit
```

The SET_DEFAULT_PROTECTION Command

You use this command to restore the default ACL protection in the SAD. (As noted in the previous section, you should if possible ensure that these ACLS are never altered from their default state.) SET_DEFAULT_PROTECTION also restores the default read/write lock settings in the SAD, both for password- and ACL-protected systems.

You also use the command to convert a password SAD to an ACL SAD. The format of the command is as follows:

```
{SET_DEFAULT_PROTECTION}  [-CONVERT]
{SDPR                  }
```

You specify the -CONVERT option only if you are converting a password SAD to ACL protection.

The SET_SYSTEM_GROUPS Command

You use this command to allow or disallow the use of system-wide groups on your system. The format of the command is as follows:

```
{SET_SYSTEM_GROUPS}  [-ON ]
{SSG              }  [-OFF]
```

The -ON option, which is the default, enables system-wide groups.

The -OFF option disables system-wide groups. In fact, all that happens immediately is that PRIMOS stops assigning system-wide groups to users when they log in. Although inactive, user- and project-related group information remains on the system until the System Administrator issues a REBUILD command.

PROJECT-LEVEL COMMANDS

The ADD_PROJECT Command

You use this command to create a new project on your system. When you give the command, EDIT_PROFILE creates a new project directory in the SAD, and defines the project according to the options you select. The format of command is as follows:

```
{ADD_PROJECT}  [project-id [options]]
{AP         }
```

You must specify the project-id, which is the name of the project to be created, if you specify any options.

The options for the ADD_PROJECT command are as follows:

Option	Meaning
-PA user-id	Specifies the user-id of the Project Administrator of the new project. If you do not use this option, or leave out the user-id, EDIT_PROFILE prompts you for this information.
{-CREATE_PA -CR}	Specifies that you want to define the attributes of the Project Administrator as a member of the new project. (A Project Administrator does not have to belong to the project which he or she administers.)
-SIZE entry-count	Allows you to specify how many users you expect to belong to the project. If you leave out 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 that the data base can efficiently handle, and gives you the opportunity to rebuild the data base, specifying a new size if you wish.
{-NO_QUERY -NQ}	Stops EDIT_PROFILE asking you whether you want to check or change the newly-created project definition.
-LIKE reference	Allows you to specify a reference identifying the id of an existing project, when you want the new project to have the same attributes as an existing project.
-PROFILE	Specifies that you want to define the profile of the new project while you are creating it. If you do not use this option, the profile will be set up with null entries.

If, when you start an EDIT_PROFILE session, the only project on your system is DEFAULT, that is defined to be your "current project". (For information on current project, see the ATTACH_PROJECT command.) However, as soon as you create another project using the ADD_PROJECT command, DEFAULT ceases to be your current project. You then have no current project unless you give an ATTACH_PROJECT command to specify one.

The following example shows how a System Administrator might use the ADD_PROJECT command to define a new project on his system.

```
> ADD_PROJECT                                   /* SA creating new project
Enter project id: DEALERS
Project administrator name: DLR_MGR

Set limits for project "DEALERS":
    Groups: .CARS .PARTS                        /* 2 groups will be allowed
*** New group added to system: ".CARS".         /* New project, new groups
*** New group added to system: ".PARTS".
Create administrator's entry? NO
Create project profile? YES

Set profile attributes for project "DEALERS":
    Groups: .CARS .PARTS
*** New group added to project: ".CARS".
*** New group added to project: ".PARTS".
    Initial attach point: <MARKET>DEALERS
Project "DEALERS" created.
    20 entries in prime area; file is 1 record long.
Check entry? Y

*************************************************************************
Project: DEALERS                              Administrator: DLR_MGR
    No entries in use out of 20.

Master project limits:
    Groups: .CARS .PARTS

Project profile:
    Groups: .CARS .PARTS
    Initial attach point: <MARKET>DEALERS
*************************************************************************
Change entry? N
>
```

The ATTACH_PROJECT Command

You use this command to identify a particular project as your current project. A current project serves as a default; that is, if you use one of the EDIT_PROFILE subcommands which allows you to specify a project-id, and you do not specify the id, the subcommand is performed on your current project.

EDIT_PROFILE recognizes a current project in one of three ways:

- While 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 is the current project.
- If you use the ATTACH_PROJECT subcommand, the project you specify becomes the current project.

The format of this command is:

```
{ATTACH_PROJECT}  [project-id]
{ATP           }
```

You need not specify the project-id on the command line. If you leave it out, EDIT_PROFILE asks you to enter it.

See also: DETACH_PROJECT

The CHANGE_PROJECT Command

You use this command to change the attributes or size of a project. The format of the command is as follows:

```
{CHANGE_PROJECT}  [project-id [options]]
{CP            }
```

where project-id identifies the project to be changed. You must specify the project-id if you specify any other options on the command line. The other options are as follows:

Option	Meaning
-PA [user-id]	Specifies that you are changing the administrator of the project. If you leave out the user-id of the new Project Administrator, EDIT_PROFILE prompts you for it.
-PROFILE	Specifies that you want to change the profile of the project. For example, you would use this option if you wanted to associate another access group with the project.
-LIMITS	Specifies that you want to change the master project limits. (Limits are the set of access groups which may be associated with the project.)
-SIZE [entry-count]	Specifies that you want to change the amount of space reserved in the user profile data base for information related to the project. entry-count specifies the number of project members for whom you wish space allocated. If you leave out the entry-count, EDIT_PROFILE prompts you for it. Using -SIZE is the only way to control the entry-count with the CHANGE_PROJECT command. However, you can also use the REBUILD command

to change the entry-count, and this is preferable if you are not changing other project attributes.

When you use -SIZE, the part of the user profile data base containing information on the specified project is rebuilt, so you should try not to use it often. However, it does provide a useful way to conserve disk space.

-LIST — Displays the project attributes after other changes you specify in the command line have been made.

The following example illustrates the use of the CHANGE_PROJECT command.

```
Profile editor [rev 19.0] in system administrator mode 25 Aug 80 11:59:
20.
> CP
Enter project id: DEFAULT                  /* Change the default project
Change administrator? YES                  /* Set a new PA
Project administrator name? ADMIN_1
Change project profile? YES                /* Create the profile

Project profile:
    Groups: <none>
    Initial attach point: <none>

Set profile attributes for project "DEFAULT":
    Groups: .DEFAULT .SYSTEM_WIDE
    Initial attach point: <CMDISK>DEFAULT_DIRECTORY
Change project limits? NO                  /* Leave MPP as is
Project "DEFAULT" updated 25 Jul 82 12:00:24.
> QUIT
```

The DELETE_PROJECT Command

You use this command to remove a project from your system. If any project members are using the project when you issue the command, EDIT_PROFILE offers you the chance to change your mind. If you delete a project that is the default login project for any users, they will then have no default login project.

The format of the command is:

```
{DELETE_PROJECT}  [project-id]
{DP            }
```

where project-id identifies the project to be deleted.

If you leave out the project-id, EDIT_PROFILE deletes your current project, unless you have none. (Current project is described under the ATTACH_PROJECT command.)

If you have no current project, and omit the project-id, EDIT_PROFILE asks you what project you want to delete, and then deletes it.

You cannot use this command on a non-ACL system.

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

```
Profile editor [rev 19.0] in system administrator mode 25 Aug 82 11:12:
32.
> DELETE_PROJECT                              /* Delete current project
Project "DUMMY" currently contains 5 entries.
Do you want to delete it? YES
*** Project "DUMMY" deleted 25 Aug 82 11:12:44.
    (3 default projects reset.)
> quit
```

The DETACH_PROJECT Command

You use this command to clear the setting of a current project set by a previous ATTACH_PROJECT or other EDIT_PROFILE command. The format is:

{DETACH_PROJECT / DTP} [project-id]

You do not have to specify the project-id. If you choose to do so, you must specify the correct id of your current project.

Once you have given this command, you have no current project. If you give subsequent commands within EDIT_PROFILE, you must therefore specify in the command line what project the command applies to.

See also: ATTACH_PROJECT

The LIST_PROJECT Command

You use this command to list the attributes of a project. Attributes listed always include the project limits, and may include user and other attributes, depending on the options you select.

The format of the command is:

{LIST_PROJECT / LP} [project-id [options]]

where project-id identifies the project to be listed. You must specify it if you use any other options. These options are as follows:

Option	Meaning
-PROFILE	Lists the project profile, which shows project-based groups and the Initial Attach Point.
-USER user-id	Lists the profile of the specified project member. To list only user attributes, without project attributes, you should use the LIST_USER command, described below.
-ALL	Lists the profiles of all project members.
-OUTPUT pathname	Directs the output of the command into the file you specify with pathname. If you specify a simple filename rather than the full pathname, the file is opened in the SAD. This option is particularly useful with the -ALL option, which may produce voluminous output.
-APPEND	Adds output of the command at the end of the file you specified with the -OUTPUT option. You use this option only with the -OUTPUT option. Unless you use -APPEND, any contents of the specified output file are overwritten.
-TTY	Directs output of the command to your terminal. Output is displayed at your terminal by default. Therefore, you need only use -TTY when you want to check output at your terminal, and have redirected output to a file with the -OUTPUT option.

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.

```
> LP DEFAULT -PROFILE                    /* We want to see the profile too

*************************************************************************
Project: DEFAULT                         Administrator: ADMIN_1
    3 entries in use out of 268.

Master project limits:
    Groups: .DEFAULT .SYSTEM_WIDE

Project profile:
    Groups: .DEFAULT .SYSTEM_WIDE
    Initial attach point: <MARKET>DEFAULT_DIRECTORY
*************************************************************************
> QUIT
```

The SET_PROJECT_GROUPS Command

You use this command to allow or disallow the use of project-based access groups on your system. The format of the command is as follows:

```
{SET_PROJECT_GROUPS}  [-ON ]
{SPG               }  [-OFF]
```

The -ON option, which is the default, enables project-based groups.

The -OFF option disables project-based groups. In fact, all that happens immediately is that the system notes that groups are not to be used. User- and project-related group information remains on the system until the System Administrator issues a REBUILD command.

USER CONTROL COMMANDS

The ADD_USER Command

You use this command to add a user to the system, a project, or both, and to create the user's profile. The format of the command is:

```
{ADD_USER}  [user-id [options]]
{AU      }
```

where user-id identifies the user to be added. You must specify the user-id if you use any of the other options. These options are as follows:

Option	Meaning
-LIKE reference	Specifies that the new user will have the same attributes as an existing user, identified by reference. If you also specify a project with the -DEFAULT or -PROJECT options, then the existing user must belong to that project.
{-NO_QUERY -NQ}	Stops EDIT_PROFILE asking you whether you want to check or change the newly-created user profile.
-PASSWORD [password]	Allows you to specify a login password for the new user whom you are adding to the system. This option implies -SYSTEM, described below. When you add a user to the system you must specify a password; if you leave it out, EDIT_PROFILE prompts you for it. (You may specify a null password by entering a

carriage return in response to the prompt, if you are allowing the use of null passwords on your system.)

-PROFILE — Specifies that you want to create the user's profile explicitly, in response to prompts from EDIT_PROFILE. If you leave out this option, the profile is set up from the default attributes in the project profile.

-SYSTEM — Specifies that you are adding the user to the system, which is the default in System Administrator mode. This implies both -PASSWORD and -DEFAULT.

-PROJECT [project-id] — Allows you to specify the project to which you are adding the user. You can only add a user to one project at a time, although a user can belong to several projects. This option does not affect the user's default login project. You cannot specify both -PROJECT and -DEFAULT.

If you leave out the project-id, EDIT_PROFILE assumes your current project (see the ATTACH_PROJECT command), unless you have none. In this case, it prompts you to specify a project-id.

{-DEFAULT / -DFLT} [project-id] — Allows you to specify the project to which you are adding the user, and makes that project the user's default login project. -DEFAULT implies the -SYSTEM option. You cannot specify both -PROJECT and -DEFAULT.

If you don't specify this option when adding a user to the system, EDIT_PROFILE asks you for the user's default login project, unless the only project on your system is the system DEFAULT project. When this is so, DEFAULT is the user's default login project.

If you leave out the project-id, EDIT_PROFILE assumes your current project (see the ATTACH_PROJECT command), unless you have none. In this case, it prompts you to specify a project-id.

{-VERIFY_NS / -VNS} — Searches the SADs of any systems, to which your system is attached by network, which recognize user-ids defined on your system, to see whether the user-id you have

specified already exists on another system. If so, EDIT_PROFILE prints a warning message, listing the PRIMENET nodenames of the systems where it found duplicates. This option makes it easier to avoid confusing duplication of user-ids across the network.

Since -SYSTEM is the default in System Administrator mode, the user you specify is added only to the system, except in the following situations:

- If you specify -DEFAULT or -PROJECT, you explicitly add the user to a project.
- If the only project on your system is the system DEFAULT project, EDIT_PROFILE automatically adds all users to that project when you add them to the system.
- If you specify no options at all, and you have a current project (see the ATTACH_PROJECT command), then EDIT_PROFILE adds the user to that 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.

The following example shows how a user with id ARTHUR might be added to a system. Since ARTHUR is a new user, the administrator specifies the -PASSWORD and -DEFAULT options. -SYSTEM is assumed, and the administrator uses the -PROFILE option to create ARTHUR's profile explicitly.

The system is on a network, so the administrator uses the -VERIFY_NS option. Despite the fact that the same id is found on two other systems, the administrator goes ahead and creates the profile.

```
> ADD_USER ARTHUR -PW XYZZY -DFLT DEFAULT -PROFILE -VERIFY_NS
Warning:  user "ARTHUR" found on system(s):
    NNK
    NNS

Set system-wide attributes for user "ARTHUR": /* Totally new user
    Groups: .PA_GROUP                         /* PA's in this group
*** New group added to system: .PA_GROUP      /* SA created new group

User "ARTHUR" added to system.
Check entry? NO                               /* We'll look later.

Set attributes for user "ARTHUR" in project "DEFAULT":
    Groups: .DEFAULT .SYSTEM_WIDE
*** New group added to project: .DEFAULT      /* First occurence
*** New group added to project: .SYSTEM_WIDE  /* in this project
    Initial attach point: <CMDISK>PROJECT_1_UFD
```

```
User "ARTHUR" added to project "DEFAULT".
Check entry? YES
************************************************************************
System-wide attributes for user "ARTHUR":
    Groups: .PA_GROUP
    Default project: DEFAULT

Attributes for user "ARTHUR" in project "DEFAULT":
    Groups: .DEFAULT .SYSTEM_WIDE
    Initial attach point: <CMDISK>PROJECT_1_UFD
************************************************************************
Change entry? no                                  /* Looks good
> QUIT
```

The CHANGE_USER Command

You use this command to change a user's attributes. You can alter system-wide attributes, project-based attributes, or both. The format of the command is as follows:

```
{CHANGE_USER}  [user-id [-PASSWORD [password]]
{CU          }  [-PROJECT [project-id]] [-SYSTEM] [-LIST]]
```

where user-id identifies the user whose attributes you want to change. You must specify the user-id if you include any other options. These options are as follows:

Option	Meaning
-PASSWORD [password]	Allows you to specify a new login password for the user. If you leave outv the password itself, EDIT_PROFILE prompts you to enter it.
-PROJECT [project-id]	Specifies that you are changing the user's project-based attributes in the project identified by project-id. If you leave out the project-id, EDIT_PROFILE assumes your current project (see the ATTACH_PROJECT command), unless you have none. In this case, it prompts you to specify a project-id.
-SYSTEM	Specifies that you are changing the user's system-wide groups or default login project.
-LIST	Lists the user's attributes after the specified changes have been made.

The following example shows how an administrator might change the system-wide attributes of a user called ARTHUR.

```
Profile editor [rev 19.0] in system administrator mode 25 Aug 80 10:45:
28.
> cu arthur -system                          /* Let's change global groups
System-wide attributes for user "ARTHUR":
    Groups: .PA_GROUP
    Default project: DEFAULT

Set system-wide attributes for user "ARTHUR":
    Groups: .pa_group .adventurers
*** New group added to system: .ADVENTURERS
    Default project: <cr>                      /* Blank line means no change
User "ARTHUR" updated 25 Jul 82 10:45:32.
```

The DELETE_USER Command

You use the DELETE_USER commmand to remove a user from your system, or from a project. When you delete a user from the system, EDIT_PROFILE also removes that user from any projects to which he or she belonged.

The format of the command is as follows:

```
DELETE_USER   [user-id [-PROJECT [project-id]]]
DU
```

where user-id identifies the user whom you are deleting. If you do not specify this, EDIT_PROFILE prompts you to enter it.

Unless you specify -PROJECT, the user is removed from the system. When you specify -PROJECT, EDIT_PROFILE removes the user only from the individual project.

If you leave out the project-id, EDIT_PROFILE assumes your current project (see the ATTACH_PROJECT command), unless you have none. In this case, it prompts you to specify a project-id.

The following examples show how to use DELETE_USER at both the system and the project level. In the first example, the administrator removes a user from the administrator's current project, and therefore does not have to specify the project-id.

```
> du jo_turkey -project
*** User "JO_TURKEY" deleted from project "DEFAULT" 25 Nov 81 11:05:48.
```

In the second example, the administrator removes a user from some other project, so she specifies the project-id explicitly.

```
> du tom_turkey -proj thanks
*** User "TOM_TURKEY" deleted from project "THANKS" 25 Nov 81 11:05:56.
```

In the third example, the administrator removes a user called JIMMY from the system.

```
> du jimmy
*** User "JIMMY" deleted from system 25 Nov 81 11:07:00.
*** User "JIMMY" deleted from project "EDUCATION" 25 Nov 81 11:07:00.
*** User "JIMMY" deleted from project "BAD_BOYS" 25 Nov 81 11:07:04.
    (Project "BAD_BOYS" is now empty.)
*** User "JIMMY" deleted from project "DEFAULT" 25 Nov 81 11:07:08.
> quit
```

The LIST_USER Command

You use this command to list a user's attributes, either as a member of one project, or as a member of each of the projects to which he or she belongs. The format of the command is as follows:

$$\left\{\begin{matrix}\text{LIST\_USER}\\ \text{LU}\end{matrix}\right\} \left[\text{user-id} \left[\begin{matrix}\text{-PROJECT [project-id]}\\ \text{-ALL}\end{matrix}\right]\right]$$

where user-id identifies the user whose attributes you want to list. If you do not supply it, EDIT_PROFILE prompts you to enter it.

You use the -PROJECT option to specify that you want to list the user's attributes as a member of a particular project. EDIT_PROFILE assumes -PROJECT if DEFAULT is the only project on your system.

If you leave out the project-id, EDIT_PROFILE assumes your current project (see the ATTACH_PROJECT command), unless you have none. In this case, it prompts you to specify a project-id.

You use the -ALL option to list the user's attributes in all the projects she belongs to.

In the following example, an administrator lists the attributes of a user called CAROLINA in all her projects.

```
Profile editor [rev 19.0] in system administrator mode 10 Nov 80 11:11:
12.
> lu carolina -all
********************************************************************
System-wide attributes for user "CAROLINA":
    Groups: .DEFAULT .EDUCATIONAL .SYSTEM_WIDE
    Default project: EDUCATION

Attributes for user "CAROLINA" in project "EDUCATION":
    Groups: .TEACHERS .GAMES .LIBRARIANS
    Initial attach point: <EDUCTN>CAROLINA
```

```
Attributes for user "CAROLINA" in project "DEFAULT":
    Groups: <none>
    Initial attach point: <none>
**********************************************************************
> quit
```

The VERIFY_USER Command

You use the VERIFY_USER command only if your system is on a network. If so, you can use the command to find out whether user-ids on your system also exist on networked systems which recognize the ids defined on your systems. If EDIT_PROFILE finds an id duplicated elsewhere, it prints a list of the PRIMENET nodenames of the systems where it found duplicates.

VERIFY_USER, like the -VERIFY_NS option of the ADD_USER command, makes it easier to avoid confusing duplication of user-ids across the network. The format of the command is as follows:

```
{VERIFY_USER} [user-id]
{VU         } [-ALL   ]
```

If you specify a user-id, EDIT_PROFILE searches the SADs of the other systems only for that id, and prints a list of duplicates if found.

You use the -ALL option to search the SADs of other systems for all the ids on your system. Again, EDIT_PROFILE prints a list of all the duplicates it finds.

PROJECT ADMINISTRATOR MODE

A system that does not use ACLs can support only one project, DEFAULT, and the System Administrator has also to be the administrator of that project. The System Administrator can use EDIT_PROFILE in System Administrator mode, which is more powerful than Project Administrator mode. The discussion of Project Administrator mode is therefore irrelevant to administration of non-ACL systems.

On an ACL system, the System Administrator may define more than one project, and delegate some of the work of maintaining these projects. When creating a project, the System Administrator has to specify the user-id of someone designated as that project's administrator. The Project Administrator can then use a limited set of EDIT_PROFILE commands in Project Administrator mode.

A Project Administrator can only change the attributes of members of the particular project or projects that he or she administers. The following discussion is addressed to Project Administrators.

Entering Project Administrator Mode

To use EDIT_PROFILE in Project Administrator mode, you must specify the -PROJECT option with the project-id of your project. The Project Administrator therefore gives the command in the following form:

```
EDIT_PROFILE [pathname] -PROJECT project-id
```

You supply a pathname 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, then you give the name of the disk partition (MFD) in which your project is kept.

For example, suppose HARRY is Project Administrator for a project called HARKNESS in a partition called HAMPER, which is on a system called SYS.H. If HARRY is logged in to another system, he gives the EDIT_PROFILE command as follows:

```
EDIT_PROFILE <HAMPER> -PROJECT HARKNESS
```

PROJECT ADMINISTRATOR COMMANDS

As Project Administrator, you can use the following EDIT_PROFILE subcommands:

Command	Meaning
ADD_USER	Adds a new member to your project.
CHANGE_PROJECT	Changes the profile of your 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 subcommands available in Project Administrator mode.
LIST_PROJECT	Lists the attributes of your project, and of one or more project members.
LIST_USER	Lists the attributes of an individual project member.
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 explaining project-level commands.

The ADD_USER Command in Project Administrator Mode

The Project Administrator adds users to projects and sets up user profiles with the ADD_USER command. The command format is:

```
{ADD_USER}  [user-id] [-PROFILE] [-LIKE reference]
{AU      }            [-NO_QUERY]   [-PROJECT project-id]
                      [-NQ       ]
```

If you specify only a user-id, that user is added to your project as a new member with the default attributes described in the project profile. To establish a new profile, use the -PROFILE option. The system then queries you to provide the profile you want.

Normally, the user-id argument and the -PROFILE option will be the most useful. Project Administrators may also use the following options:

Option	Meaning
-LIKE reference	Specifies existing user attributes the new user will assume. The reference identifies the user who has the existing attributes.
{-NO_QUERY} {-NQ}	Suppresses the "Check" and "Change" questions posed by the system after the new user is added.
-PROJECT project-id	Specifies the project to which the user is added. You only use this option if you administer several projects.

The CHANGE_PROJECT Command in Project Administrator Mode

The Project Administrator uses this command to change the project profile. The format is:

```
{CHANGE_PROJECT}[project-id] [-PROFILE] [-LIST]
{CP            }
```

You need not specify the project-id unless you administer several projects.

The -LIST option displays the latest version of project attributes. You must specify the -PROFILE option, which specifies that you want to change or list the project profile.

The CHANGE_USER Command in Project Administrator Mode

As Project Administrator, you use this command to change the profile of an individual member of your project. Note that your System Administrator may restrict the attributes that you can change. For example, a Project Administrator may only assign access groups for project members from the list of groups assigned to that project by the System Administrator. The format of the command is:

```
{CHANGE_USER}  [user-id] [-PROJECT project-id] [-LIST]
{CU          }
```

The user-id identifies the project member whose attributes you wish to change. If you do not supply a user-id, the system prompts you to enter one.

The -PROJECT option is useful only if you administer several projects.

The -LIST option prints the user's attributes after you have changed them.

The DELETE_USER Command in Project Administrator Mode

You use this command to delete a user from your project. The command format is:

```
{DELETE_USER}[user-id] [-PROJECT project-id]
{DU         }
```

The -PROJECT option is useful only if you administer several projects.

The LIST_PROJECT Command in Project Administrator Mode

This command lists the attributes of the specified project, as well as those of either one or all users in the project. The list always includes the project limits imposed by your System Admininstrator. The command format is:

```
{LIST_PROJECT}  [project-id]  [-PROFILE] [-ALL          ]
{LP          }                          [-USER user-id]
                [-OUTPUT pathname]  [-APPEND]  [-TTY]
```

You need not specify a project-id unless you administer several projects.

The options are as follows:

Option	Meaning
-ALL	Lists profiles for all users in the project.
-OUTPUT pathname	Directs the output from the command into the file specified by pathname. This option is very useful if the output that would be listed from the -ALL option is extremely long. If you do not supply a pathname, the command will fail.
-APPEND	Adds command output at the end of the file you specified with -OUTPUT. Otherwise, if you have specified an existing file and do not use -APPEND, the contents of that file will be overwritten.
-TTY	Displays output of the command at the terminal. If you use the -OUTPUT option and also want to see the output at your terminal, use the -TTY option as well. Unless you specify -TTY when you use -OUTPUT, your terminal will not display the output file.
-PROFILE	Displays the project profile.
-USER user-id	Lists the profile of the project member identified by the user-id. You can only specify one user-id; if you want to list only user attributes, use the LIST_USER command.

The LIST_USER Command in Project Administrator Mode

You use this command to display the attributes of an individual member of your project. The format of the command is:

```
{LIST_USER}  [user-id]  [-PROJECT  project-id]  [-ALL]
{LU       }
```

The user-id identifies the member of your project whose attributes are to be listed. If you do not specify it, the system will prompt for one.

You need not use the -PROJECT option unless you administer several projects.

You use the -ALL option to display the user's attributes in each project to which that user belongs. However, it will only display those projects which you administer.

The REBUILD Command in Project Administrator Mode

You use this command to rebuild your project to hold more members. You use the command in the following form:

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

You do not use the -PROJECT option unless you administer several projects.

You use the -SIZE option to specify the total number of members you want in your project. This 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.

Note that project members cannot log into your project while EDIT_PROFILE rebuilds it.

EDIT_PROFILE MESSAGES

The following list describes messages which may be displayed by EDIT_PROFILE. Following each message in brackets is a description of the type of the message. The following are used:

NOTICE: The error is of an advisory nature only; execution continues.

RETRY: Data of an invalid format has been entered. It must be entered correctly before execution may continue.

INIT: An error occurred while processing the PRIMOS command line invoking EDIT_PROFILE. The user is returned to PRIMOS command level.

COMMAND: Current command is aborted, and the user is returned to EDIT_PROFILE command level.

FATAL: EDIT_PROFILE is aborted, and the user is returned to PRIMOS command level. Fatal error messages are usually preceded by a standard PRIMOS error message.

Initialization Errors

- SAD does not exist. [FATAL]

EDIT_PROFILE has discovered that no SAD exists in the directory specified as the parent on the command line. (By default, the parent is the MFD of logical device zero.) You are given this message in case the SAD has been destroyed, in which case you would not necessarily want to create a new one. This message is always followed by the query "Create it?". If you indeed want to create a new SAD, answer YES. If you answer NO, EDIT_PROFILE returns you to PRIMOS command level.

- *** SAD is either not properly set up or has been damaged *** [FATAL]

This message is always followed by the advisory: "Restore from backup or delete and re-initialize." The message appears when the SAD can be found but the UVF cannot be accessed. Possible reasons for this include the following:

- The UVF has been inadvertently deleted.
- The disk on which the SAD resides has been damaged.
- A previous initialization of the SAD was aborted.
- An incomplete MAGRST of the SAD has been done.
- An incomplete COPY of the SAD has been done.

In order to rectify the problem, a good copy of the SAD should be restored from a backup disk or tape, or the damaged SAD should be deleted and a new one created.

- *** Protection in the SAD has been damaged *** [FATAL]

This message appears when the SAD and UVF can be accessed, but the MGF and/or MPF cannot. It generally indicates that the ACLs protecting the files and directories of the SAD have been damaged or changed. It is always followed by the query: "Do you want EDIT_PROFILE to fix it?". If you answer YES to the query, EDIT_PROFILE will reprotect the SAD in the standard manner, and will then restart the run of EDIT_PROFILE. If you answer NO, EDIT_PROFILE is aborted.

- *** Read/write locks in the SAD have been damaged *** [FATAL]

This message appears when the UVF and MPF can be accessed, but a file in use error was returned on the MGF. It indicates that 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. It is always followed by the query: "Do you want

EDIT_PROFILE to reset them?". If you answer YES to the query, EDIT_PROFILE will reset the read/write locks and reinitialize itself. If you answer NO, EDIT_PROFILE will abort and return you to PRIMOS command level.

- <primos error> Can't read user ID [FATAL]

The user-id of the user running EDIT_PROFILE could not be retrieved from PRIMOS for some reason. This error indicates a serious problem with either PRIMOS or EDIT_PROFILE, and should be reported to your system analyst.

- Directory pathname too long. [INIT]

The pathname of the SAD's parent directory which was supplied to EDIT_PROFILE was more than 80 characters long. This name is limited to 80 characters to ensure that the longest subtree name in the SAD may be appended to the parent tree within PRIMOS's 128-character limit for pathnames.

- Parent directory is not an ACL directory. [FATAL]

An attempt was made to create a SAD in a non-ACL directory. Non-ACL SADs may be created only from the system console on the MFD of the command device.

- Warning: security and project support cannot be provided without ACLs. [NOTICE]

You created a password SAD at the console, and EDIT_PROFILE is informing you that restrictions will apply.

- <primos error> When adding Priority ACL. [FATAL]

When EDIT_PROFILE is run in initialization mode at the system console, it attempts to put a Priority ACL on the disk to facilitate creation of ACLs in which SYSTEM might not be found. For some reason, the priority ACL could not be set for the command device. This error indicates a serious problem with either PRIMOS or EDIT_PROFILE, and should be reported to your field analyst.

- EDIT_PROFILE is in use. Please try again in a few minutes. [FATAL]

Another user is running EDIT_PROFILE in System Administrator mode. In order to prevent updates from being confused, 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.

- Insufficient access rights. <sad_pathname> [INIT]

You are not authorized to use EDIT_PROFILE on the specified SAD.

- System administrator = "<sa_name>". [NOTICE]

When a SAD is created anywhere but the system console, the user running EDIT_PROFILE must become the System Administrator (because of ACLs). EDIT_PROFILE is informing you that it has set the System Administrator in this SAD to be the name specified.

- <filename> created at <datetime>. [NOTICE]

When running in initialization mode, EDIT_PROFILE informs you as it creates the files directly contained in the SAD.

- *** Creating project "DEFAULT". [NOTICE]

When creating a password SAD, project DEFAULT is always created. EDIT_PROFILE informs you of this fact.

General Errors

- Unrecognizable command "<command>". [COMMAND]

The command given is unknown to EDIT_PROFILE in its current mode. This message can mean either that the command is completely unknown, or that a system-administrator-only command was given in Project Administrator mode.

- Unrecognizable option in command. [COMMAND]

The command was invoked with an option that does not exist in this mode. Either a mistyping or a System-Administrator-only option was given in Project Administrator mode. The command line option list will be repeated, and the option in error will be indicated by an uparrow (^) on the line underneath it.

- Improper data format in command. [COMMAND]

Some object of a command or command option had incorrect format. For example, the object could be a user or project-id which is more than 32 characters long, or which contains an illegal character. The erring object is indicated by an uparrow (^) on the following line.

- Duplication of options in command. [COMMAND]

An option was given to a command more than once. All EDIT_PROFILE commands allow only single instances of each option. The duplicated option is indicated by an uparrow (^).

- Too many objects specified in command. [COMMAND]

All EDIT_PROFILE commands take at most one object. More objects were specified than the command expected to see. Perhaps the "-" was left off an option name. The excess object is indicated by an uparrow on the following line.

- Incorrect format: "<option1>" and "<option2>" options are exclusive. [COMMAND]

Two options were given to a command, but these options may not be given together.

- Incorrect format: "<option_name>" option requires an argument. [COMMAND]

An option that takes an argument was given, but no argument was supplied. Re-enter the command either without the option, or supplying the required argument.

- Incorrect format: No options allowed without <object_type>. [COMMAND]

All commands that take objects require that the object be supplied if any options are given. Here, one or more options were given, but no object was. Either use the command with no options (in most cases EDIT_PROFILE will prompt for them), or supply an object.

- "<project_id>" is not a valid project. [COMMAND]

The requested project does not exist or, in Project Administrator mode, is not under the jurisdiction of the Project Administrator.

- *** EDIT_PROFILE system error: <error> when parsing command. [FATAL]

This indicates an EDIT_PROFILE programming error, and should be reported to your analyst.

- *** Group <group_name> not legal for this project. [NOTICE]

When assigning a project-based group to a user (with ADD_USER or CHANGE_USER) or to a project profile (with ADD_PROJECT or CHANGE_PROJECT), that group was not found in the MPP for that project. The group is not assigned.

- *** Project Data File overflow [COMMAND]

The Project Data File (PDF) for the project in which you were working has attempted to grow to more than 64K words. The command is aborted. First, attempt a REBUILD of the project to delete any "dead" entries from the PDF. If that does not solve the problem, break the project up into more than one project.

- *** New <object> added to <location>: "<name>". [NOTICE]

The <object> will be either PROJECT or GROUP. The <location> will be either SYSTEM or PROJECT. The <name> indicates the name of the object being added. This message indicates that a new object of the given type has been added to the data bases. It is provided primarily to warn you in case you have made a typographical error and inadvertently created a new group or project that no one can use. For instance, if you wanted to add the group .OPSYS to a user's list, and typed instead .OPSSS, you would get the message "*** New group added to system: .OPSSS". At this point you could go back and correct your mistake.

- Pathname must be fully qualified. [RETRY]

When entering an initial attach point, you have not supplied a full pathname, that is, one including a partition name. EDIT_PROFILE will continue to prompt you until a fully qualified pathname has been entered.

- Pathname must have at least one directory level. [RETRY]

When entering an initial attach point, only a partition name was supplied. EDIT_PROFILE will continue to prompt until at least one directory name is included in the pathname.

- Cannot support names of depth greater than 16. [RETRY]

The maximum pathname depth for an initial attach point is 16 levels. Supply a new initial attach point of 16 or fewer levels.

- Illegal <object_type> "<name>". [RETRY]

The <name> given is not a legal object of type <object_type>. The type may be user-id, password, group name, or project-id. EDIT_PROFILE will continue to prompt you until you enter a legal object of type <object_type>.

- *** Input truncated to 256 characters. [NOTICE]

You have attempted to enter a command line or response that is more than 256 characters long. EDIT_PROFILE ignores all characters past the 256th, and prints this warning.

- Token too long; truncated to "<token>". [NOTICE]

Tokens (individual items) in a command line may not be more than 32 characters long. You have attempted to enter a token which is longer than that. 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 be caused as a result of the truncation.

ADD_PROJECT Command Messages

- *** Project "<project_id>" already exists. Must use DELETE or CHANGE. [COMMAND]

The command was given for an existing project. Either the name was mistyped, Change_project should be used to change the attributes of the existing project, or Delete_project followed by Add_project should be used to create a new project with the existing name.

- Projects not supported in non-ACL systems. [COMMAND]

An attempt was made to create a project in a password SAD. The SAD must be converted to ACL protection with SET_DEFAULT_PROTECTION before ADD_PROJECT may be used.

- *** Can't find like reference "<project_id>". [COMMAND]

The -LIKE option was given, but the project whose attributes were to be copied does not exist.

- Project "<project_id>" created. [NOTICE]

The command was successfully executed.

ADD_USER Command Messages

- *** User "<user_id>" already on system. Must use DELETE or CHANGE. [COMMAND]

An attempt was made to use the command to create a user, but that user is already on the system. Either the name was mistyped, CHANGE_USER should be used to change the attributes of the existing user, or DELETE_USER followed by ADD_USER should be used to create a new user with the existing user-id.

- *** User "<user_id>" already in project "<project_id>". Must use DELETE or CHANGE. [COMMAND]

An attempt was made to use the command to add the user to a project, but that user is already in the project. Either the name was mistyped, CHANGE_USER should be used to change the attributes of the existing user, or DELETE_USER followed by ADD_USER should be used to create a new user with the existing user-id.

- *** Can't find like reference "<user_id>". [COMMAND]

The -LIKE option was given, 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 reference user is either not in the UVF, or is not in the PVF of the specified project.

- Verify_ns option may only be used by true SA; ignored. [NOTICE]

Because the -VERIFY_NS option involves opening SADs on remote systems, the option may not be used by anyone other than the "real" System Administrator known to PRIMOS. The option is ignored and execution continues.

- Warning: User "<user_id>" found on system(s): <list> [NOTICE]

The -VERIFY_NS option was given, and the user was found on at least one other system in the naming sphere. All systems on which the user was found are listed.

- Warning: all users must have an initial attach point. [NOTICE]

No initial attach point was specified for the user being added. All users must have an initial attach point in order 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 in order to be able to log in to that project.

- Warning: Project "<project_id>" is overloaded. [NOTICE]

The Project Validation File is more than 75% full, or the number of overflow entries in the PVF is more than 10% of the total number of entries. For maximum efficiency, the PVF should be rebuilt. If the -NO_QUERY option was not given, EDIT_PROFILE will ask at this point if the PVF should be rebuilt.

This warning is not given if the PVF is already at the maximum size.

- Warning: User validation file is overloaded. [NOTICE]

The User Validation File is more than 75% full, or the number of overflow entries in the UVF is more than 10% of the total number of entries. For maximum efficiency, the UVF should be rebuilt. If the -NO_QUERY option was not given, EDIT_PROFILE will ask at this point if the UVF should be rebuilt.

This warning is not given if the UVF is already at the maximum size.

- User "<user_id>" added to system. [NOTICE]

The user was successfully added to the UVF. If only project DEFAULT exists, indicates that the user was also successfully added to its PVF.

- User "<user_id>" added to project "<project_id>". [NOTICE]

The user was successfully added to the specified project.

CHANGE_PROJECT Command Messages

- Only one administrator allowed in non-ACL systems. [COMMAND]

An attempt was made to the use -CHANGE_PA option on a password system. On non-ACL systems, the Project Administrator for project DEFAULT must always be the System Administrator.

- Project "<project_id>" is being modified. Please try again in a few minutes. [COMMAND]

Another administrator is using EDIT_PROFILE on the specified project. If only one person should have access to this project, this message may indicate a possible breach of security.

- Project "<project_id>" updated <date/time>. [NOTICE]

The command was executed successfully.

CHANGE_SYSTEM_ADMINISTRATOR Command Messages

- Change_sa command may not be used on test SADs. [COMMAND]

The CHANGE_SYSTEM_ADMINISTRATOR command is legal only when operating on the "live" SAD, since that is the only case in which PRIMOS' copy of the SA name may actually be changed.

- *** System administrator name is not known by PRIMOS. [COMMAND]

PRIMOS normally holds the SA name in its internal data base. When the SAD is first created, however, the name is not read by PRIMOS until the system has been re-booted. Since PRIMOS will allow the SA name to be changed only by the current SA, the CSA command may only be used once that name is established. This message is followed by the advisory: "System must be re-booted before Change_sa command may be used."

- New administrator's name same as old one! [COMMAND]

The new SA name given is the name of the existing SA. The command is ignored.

- <primos error> Can't set priority ACL. [FATAL]

An error has occurred while attempting to set the priority ACL that EDIT_PROFILE uses to ensure access to the SAD during the changeover from the old to new SA. This error indicates a serious PRIMOS problem, and should be reported to your analyst.

- <primos error> Calling Chg$sa [FATAL]

The call to the PRIMOS routine CHG$SA to change the System Administrator's name has failed. This is a serious error, and should be reported to your analyst.

- *** Mandatory exit from EDIT_PROFILE *** [FATAL]

The CHANGE_SYSTEM_ADMINISTRATOR command has successfully completed. Since the old SA will no longer have access to files in the SAD, the run of EDIT_PROFILE is terminated.

CHANGE_USER Command Messages

- *** User "<user_id>" not found on system. [COMMAND]

The user whose attributes were to be changed did not exist. Check for possible misspellings.

- *** User "<user_id>" not found in project "<project_id>". [COMMAND]

The user whose project-based attributes were to be changed did not have an entry in the specified project. Check for misspellings of both the user and the project-id.

- Warning: all users must have an initial attach point. [NOTICE]

No initial attach point was specified for the user, or his existing initial attach point was removed. All users must have an initial attach point in order 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 in order to be able to log in to that project.

- User "<user_id>" updated <date/time>. [NOTICE]

The command was executed successfully.

DELETE_PROJECT Command Messages

- *** Can't delete DEFAULT unless other projects exist. [COMMAND]

If project DEFAULT is the only project on the system, it may not be deleted.

- *** Can't delete "<filename>": <primos error>. [NOTICE]

The specified file could not be deleted. Execution continues, but you should probably delete the file later with the DELETE command.

- *** Project "<project_id>" deleted <date/time>. [NOTICE]

The command was successfully executed.

- (<count> default projects reset.) [NOTICE]

<count> users had the project which was deleted as their default login project. Since 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.

DELETE_USER Command Messages

- PROJECT option not available when only DEFAULT project present. [COMMAND]

If there is only one project on the system, a user may not be deleted only from that project. If the user is to be removed from the system, give the Delete_user command without any options.

- *** Can't delete System Administrator! [COMMAND]

The System Administrator must always have an entry in the UVF. An attempt to delete this entry has been rejected.

- *** User "<user_id>" not found on system. [NOTICE]

The user who was to be deleted had no entry in the UVF. The command continues to attempt to delete the user from all projects.

- *** User "<user_id>" not found in project "<project_id>". [COMMAND]

The user whose entry was to be deleted from a specified project had no entry in the specified project. Check for misspellings of both the user and the project-id.

- User "<user_id>" deleted from system <date/time>. [NOTICE]

The user was successfully removed from the UVF.

- User "<user_id>" deleted from project "<project_id>". [NOTICE]

The user was successfully deleted from the PVF of the specified project. If the -PROJECT option was not given, this message will appear for each project from which the user was deleted.

- (Project "<project_id>" is now empty.) [NOTICE]

The user who was deleted was the last user in the specified project. That project's PVF now contains no entries.

DETACH_PROJECT Command Messages

- "<project_id>" is not the current project. [COMMAND]

The specified project-id does not match the name of the current project. Check for mistypings.

LIST_PROJECT Command Messages

- *** User "<user_id>" not found in project "<project_id>". [COMMAND]

The user specified in the -USER option did not exist in the project.

- Can't open "<filename>" for output. Please try again. [COMMAND]

The file specified in the -OUTPUT option cannot be opened for output for some reason. Give the command again, making sure you have not made any typographical errors. In Project Administrator mode, you must supply a treename with the -OUTPUT option. Attempts to open output files with simple entrynames in PA mode will always fail because of insufficient access rights on the SAD.

LIST_SYSTEM Command Messages

- GROUPS option not supported in non-ACL SADs. [COMMAND]

Since there are no ACLs and thus no ACL groups, the -GROUPS option is illegal in a password SAD.

- Can't open "<filename>" for output. Please try again. [COMMAND]

The file specified in the -OUTPUT option could not be opened for some reason. Check for typographical errors and try again.

LIST_USER Command Messages

- *** User "<user_id>" not found on system. [NOTICE]

The specified user did not have an entry in the UVF. If the -PROJECT or -ALL option was given, the command continues to search for the user in the PVF.

- *** User "<user_id>" not found in project "<project_id>". [NOTICE]

The user did not have an entry in the specified PVF.

NO_NULL_PASSWORD Command Messages

- Warning: the following users currently have null passwords: <list> [NOTICE]

The NO_NULL_PASSWORD command was given either with no options or the -ON option. Those users who are in violation of the new standard are listed here.

REBUILD Command Messages

- *** <file> backed up into file "<file>.OLD" <date/time>. [NOTICE]

As each file is backed up, EDIT_PROFILE informs you of the names of the backup files it creates when a rebuild takes place.

- Duplicate entry for user "<user_id>" (entry <number>). [COMMAND]

A serious error in the SAD data base has been found. The rebuild is aborted, and the original UVF, MPF, and MGF are replaced by their backups. This message is always preceded by the warning: "*** EDIT_PROFILE system error! ***". Contact your field analyst should this error occur.

- <primos error> when copying files. [FATAL]

The specified error occurred while copying to or from the backup files used during the REBUILD. If the message occurs before all the "File xxx backed up..." messages appear, the original files are still in a consistent state. If it occurs after all the initial copies are done, restore the files in question from their backup copies. Generally the cause of this problem is a serious physical disk or hardware error, and if so it should be reported to your analyst.

- The following project-id's have been removed from the MPF: <list> [NOTICE]

During a system rebuild, "dead" entries are removed from the MPF and MGF. Those projects that are no longer valid are listed here.

- *** Rebuild complete <date/time>! *** [NOTICE]

The rebuild completed successfully.

SET_DEFAULT_PROTECTION Command Messages

- <primos error> Converting MFD. [FATAL]

The error indicated occurred while attempting to put an ACL on the MFD. This indicates a serious PRIMOS or EDIT_PROFILE error, and should be reported to your system analyst.

- Master Group File created <date/time> [NOTICE]

When the -CONVERT option is given to the SET_DEFAULT_PROTECTION command, a Master Group File (MGF) is created to hold the names of all ACL groups that are legal on the system.

VERIFY_USER Command Messages

- Only true SA may use Verify_user command. [COMMAND]

Because the VERIFY_USER command involves opening SADs on remote systems, the command may not be used by anyone other than the "real" System Administrator known to PRIMOS.

- User ID and -ALL option are exclusive. [COMMAND]

Either a user-id or the -ALL option may be given, but not both.

- No room. Too many nodes in network. [FATAL]

Your network has more than 256 nodes configured. EDIT_PROFILE is aborted since it has most likely suffered damage to its stack during its attempt to get information on the network. It is quite likely that EDIT_PROFILE will not be able to successfully terminate its run after this message. You can only solve this problem by reducing the number of nodes configured in your network.

- <primos error> in X$stat call. [FATAL]

EDIT_PROFILE has been unable to gather information about the network. This generally indicates a serious problem with PRIMENET and, if repeatable, should be reported to your system analyst.

- Warning: User "<user_id>" found on system(s): <list> [NOTICE]

The specified user-id was found on at least one other system. All systems on which the id was found are listed.

5
Setting System Access

INTRODUCTION

Your tasks concerning system access are:

- To set everyday access for everyone to MFDs and to top-level directories.
- To grant special disk-wide access on those occasions when someone needs it.

Your purpose in setting system access is to give all users the scope to do the tasks they need to do, while minimizing the danger of interference with files used in common. (These may include users' files, as well as system files and directories.)

Everyday access may be set in one of two ways: by directory passwords, or by access control lists (ACLs). (For a comparison of the two systems, see Chapter 15, SECURITY.) Since ACLs provide both the greatest security and the most flexibility, this chapter assumes that you will be using ACLs on your system.

The first half of this chapter, therefore, will contain:

- A brief review of access control lists. (For full information, see the Prime User's Guide.)
- Brief guidelines on setting access on MFDs and on user directories.

- An explanation of the way that ACLs and ATTACH interact.
- Two tables showing the access rights to system directories required by users (and by certain system processes).

The second half will discuss priority ACLs.

System Access

The rights that can be granted in an access control list are shown in Table 5-1.

Table 5-1
ACL Access Rights

Symbol	Right	Applies To	Meaning
R	Read	Files	File may be read.
W	Write	Files	File may be modified.
U	Use	Directories	User may attach to directories.
L	List	Directories	Directory contents may be listed.
A	Add	Directories	Directory entries may be added.
D	Delete	Directories	Directory entries may be deleted.
P	Protect	Directories	Access rights may be changed.
ALL		Files and Directories	All of the above rights.
NONE		Files and Directories	No access allowed.

Rights may be granted to:

- Any user: for example,

  ```
  JOE
  MARY
  ANY_ID
  ```

- A group of users: for example,

  ```
  .OPERATIONS
  .JUST_FOLKS
  ```

- "All other users": expressed as,

  ```
  $REST
  ```

Rights may be granted by anyone who has Protect access to the object to be protected, and List access to its parent directory.

Rights may be provided in:

- A specific ACL — an ACL explicitly set on the object.
- An access category — a file system object containing an ACL that protects whatever objects (within its own directory) you choose to link it to.
- Default protection — protection provided by the parent directory (or its parent) if no specific or category ACL has been set on an object.

Protection may be overriden by a priority ACL, set by the System Administrator or by an operator at the supervisor terminal. (More details on priority ACLs are given in the second half of this chapter.)

Within an access control list, individual rights take precedence over group rights, while group rights take precedence over $REST rights. For example, assume the following ACL:

```
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 they are members of any groups. Group rights are additive: if BILL is a member of both .SOME and .OTHERS, his rights are "LURWA". $REST applies only to those users not mentioned in the ACL. (If $REST is not specified in an ACL, $REST:NONE is assumed.)

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

PROTECTING MFDS

In general, you want to restrict access to MFDs so that only people with operations or administrative tasks can work there. On the other hand, users need some rights to the MFDs of disks they need to access. Specifically:

- Users need Use rights to access the disk at all. (See especially the section ATTACH RULES, below.)
- Users need List rights if you want them to be able to list the disk's contents or to protect top-level directories.
- On an open system, you may want to grant users Read rights as well. At a minimum, you may want to grant users Read rights to the DSKRAT file, so that they can use the AVAIL command.

Note

If you give a user (or group of users) no rights to a disk — either by specifying "user:NONE" or by omitting $REST from the ACL or the MFD — the user will not be able to ATTACH to the disk, or to gain any information about its contents. You may find this useful if there is sensitive data on some disk, and you wish it to be accessible to as few people as possible.

PROTECTING USERS' TOP-LEVEL DIRECTORIES

Only users who have Add rights to the MFD can create top-level directories. This frequently means that only the operators or System Administrator can do so. Similarly, only users with Protect and List access to the MFD can set protection on the newly created directories. Once again, that usually means the operators or System Administrator. (This also means that if users accidentally "lock themselves out" of their top-level directories by destroying their ACLs, you have to create new ACLs for them, in order to restore their access.)

You may grant users whatever rights you want to top-level directories. Generally, you will want to grant ALL rights to each top-level directory to at least one person (perhaps a project leader or project administrator). This lets the project administrators (or other users) create whatever subdirectories they need, set whatever protection is desirable, etc. The following list suggests other useful combinations of rights for users.

Most Useful Combinations of Access Rights

U Essential if users are to do anything, anywhere in the tree below. If you deny U access to the MFD, you deny the right even to search the disk for attaches or for information.

LU Lets users attach and find out where they are by giving LD and other "list", "display", and "status" commands.

LUR A friendly combination. Allows user to attach, to list things, and to read files. Users can gain all the information they want; they can copy things out of the directory (assuming they have someplace to put the copies); but they can't alter anything within the directory.

Note

U, LU, and LUR are often granted as rights to $REST.

ALUR Now users can not only attach, list, and read, they can add files in a non-destructive way. If Jane grants Sarah LURA rights to her directory, Sarah can:

- Attach there.
- Check file and directory names.
- Read a note Jane left for her.
- Create a new subdirectory. (It will have the same ACL protection as its parent directory has -- Sarah cannot avoid that.)
- Copy a file into the new subdirectory.
- Use the Editor (ED) to write Jane a note, and leave that note also in the new subdirectory. (Note, however: once the note is written and filed, Sarah cannot re-edit and alter it. To do that would overwrite and destroy the old note, and Sarah does not have the Delete rights required for these operations.)

ALUR, therefore, is a useful combination for users who want to be able to trade information, but who should not be allowed to alter each other's work.

ALURW A slightly unusual combination. With these rights, users can edit and alter a file; but they have to file the new version under an altered name, because they can't destroy the old version. This combination is useful for directories where you want to allow experimentation, but want all versions of files (such as source code) preserved.

DALURW A very common combination. It grants users the right to do everything — read and write files, add and delete items — but still denies them the right to change the protection on any of the file system objects involved. Thus, it's used in situations where an administrator, project leader, supervisor, or instructor wants to give users all "working rights" to a directory, but to keep the access control firmly in the supervisor's hands.

ALL Users can do anything to the directory, or to any directories beneath it in the tree structure. Common uses are:

- For operations personnel.
- For administrators.
- For any project leader, supervisor, or instructor who needs full rights to a directory and to the disk space it commands.
- For any user who has full and sole responsibility for a directory and the disk space it commands.
- For any group of users who will be working closely together and sharing responsibility for their files, directories, and disk space.

Granting users ALL rights to their directories contains the expected bonuses and dangers. If a group shares ALL rights to a directory, they can more rapidly and flexibly meet needs. For example:

- A troubleshooter joins the group for a few days. Any one of the members can give him access to the directory immediately.
- A key file is identified. Any member of the group 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 one of the group's user-ids. They temporarily deny that id access rights, and observe the results.

In all these situations, two things stand out:

- Members of a group which shares ALL rights must keep each other informed of what they are doing.

- Users who have ALL rights, especially when they share them with others, must be trustworthy. Any of the above scenarios can be rewritten to show malicious misuse of rights. (A group member grants a spy ALL access for a few hours; a practical joker alters concurrency locks, and a data base program goes haywire.)

There are limits to the power granted by ALL rights. They occur in the area of protection and are dependent on the rights granted in the directory above the directory to which ALL rights are given.

ATTACH Rules

At Rev. 19, the presence of access control lists allows the ATTACH command and its associated subroutines to make more careful determinations of whether or not a user may be attached to a directory. The general rules are as follows:

- If a user has no rights to a directory, he cannot be attached there.

- If a user does not have Use rights to the MFD of a disk, he cannot be attached to any directory on that disk, nor will he be given any information about them.

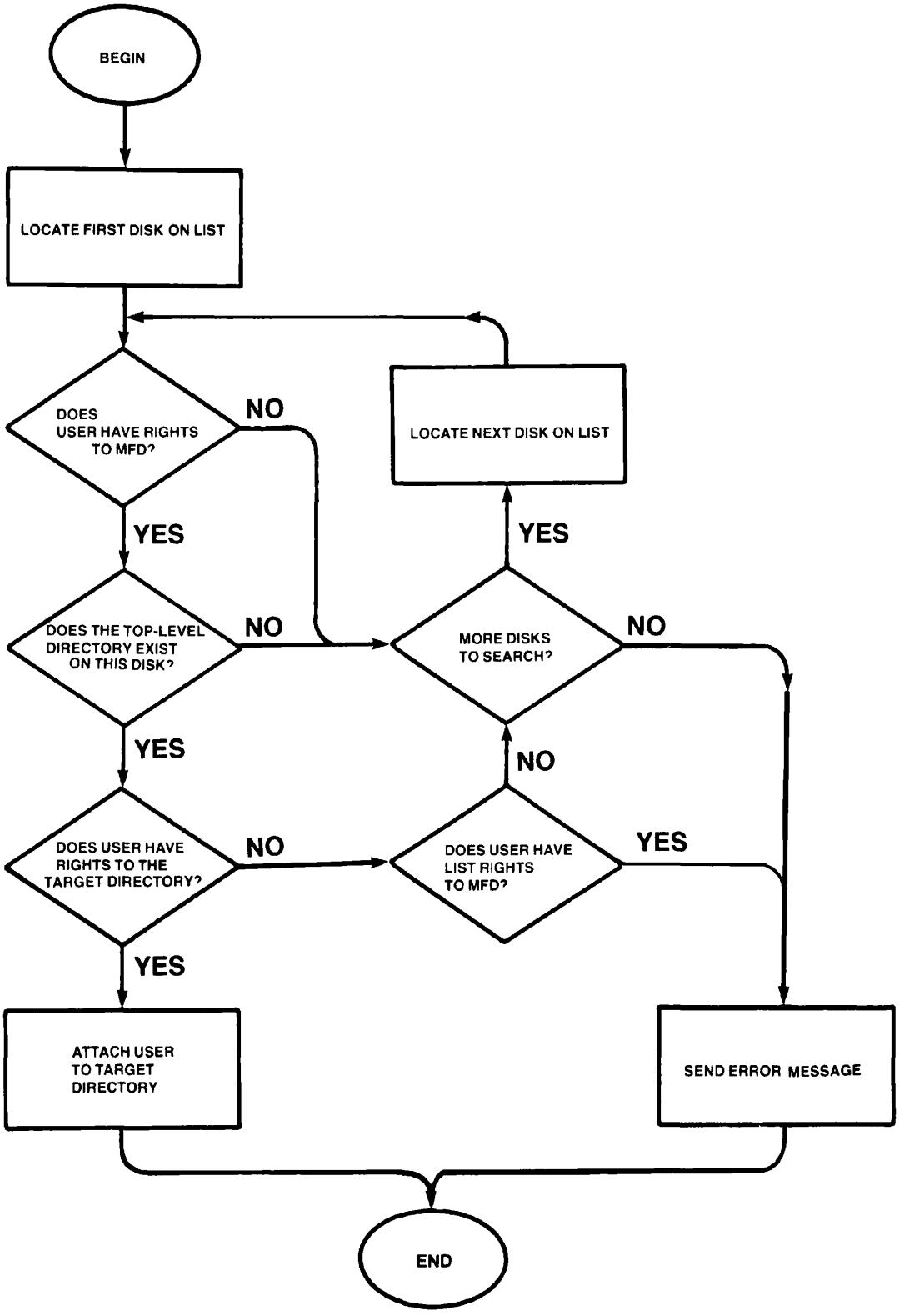
- If a user supplies a relative pathname in an ATTACH command, only his current directory tree is searched.

- If a user specifies a disk name or logical device number in an ATTACH command, only the specified disk is searched.

- If a user supplies a pathname that begins with a top-level directory, the search is carried out as shown in the flow chart in Figure 5-1. Only disks for which the user has Use (U) rights to the MFD will be searched. The order in which disks are searched is:

  — All local disks are searched first, in logical device order.

  — Remote disks (if any) are searched next, in logical device order.



How ATTACH Scans MFDs
Figure 5-1

Remote Searches: You can ensure that remote searches are done in the quickest and least costly manner by ensuring that disks from a single system are grouped together in the logical device order. Figure 5-2 shows good and bad orderings of a list.

Note

If no slaves are available to search disks on a remote system, or if a remote system is down, those disks will be skipped over, and the search will continue with the next disk on the list.

GOOD ORDER

DISK	LDEV
LOCL-1	0
LOCL-2	1
LOCL-3	2
SYSA-1	3
SYSA-2	4
SYSA-3	5
SYSB-1	6
SYSB-2	7
SYSB-3	10
SYSC-1	11
SYSC-2	12
SYSC-3	13

(3 remote calls search all disks)

POOR ORDER

DISK	LDEV
LOCL-1	0
LOCL-2	1
LOCL-3	2
SYSA-1	3
SYSB-1	4
SYSC-1	5
SYSA-2	6
SYSB-2	7
SYSC-2	10
SYSA-3	11
SYSB-3	12
SYSC-3	13

(9 remote calls needed to search all disks)

Good and Poor Ordering Of LDEV Numbers
Figure 5-2

How Searches Finish: A search finishes when one of the following conditions is met:

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

If the directory is found, and the user has Use rights to it, and to any subdirectories the user may have specified in the pathname, the user will be attached. If a top-level directory of the right name is found, but the user does not have Use rights to it, then:

- If the user has List rights to the MFD, the user will receive the error message, "Insufficient Access Rights."
- If the user does not have List rights to the MFD, the search continues with the next disk on the list.

If ATTACH finishes its scan of the MFDs without being able to attach the user anywhere, one of three things must have happened. Either the requested directory did not exist; or, it was found on a disk to which the user had no rights; or, it was on a remote disk that was temporarily unavailable. Therefore, ATTACH returns the message, "Top-level directory not found or inaccessible."

Possible Problems with Attaches

The new search rules can occasionally be confusing to users. Imagine a situation in which two users, both attached to the directory <HOME>ARMCHAIR, give the identical command, "ATTACH BALLPARK". One user, having rights to the local disk, BOSTON, is attached to <BOSTON>BALLPARK. The second, with no rights to BOSTON, finds himself attached to <ATLANTA>BALLPARK — perhaps without understanding how he got there!

Specifying Disk Names for Remote Searches: Attaching to a remote directory is much more efficient, and the messages received are more informative, if the user specifies a disk name as the first element in a pathname. When this is done, only the specified disk is searched. (Since disk names must be unique on Rev. 19 systems, there is no possibility of ambiguity.) If the directory exists, and the user has rights to it, the user is attached. If the directory does not exist on the specified disk, the user gets the message, "Not found". If the disk exists, but the remote system is down, the user gets the message, "Remote system down." If the system is up, but no slaves are available to search for the directory, the user gets the message, "No NPX slaves available."

Therefore, if the systems within your network tend to use the same directory names, you should encourage users, when attaching to remote disks, to supply disk names as part of the pathname.

PROTECTING SYSTEM DIRECTORIES

System directories contain Prime-supplied software that is used by some or all users of a system. If users (or certain system processes) have insufficient rights to these directories and to the files they contain, they may not be able to work. Table 5-2 presents the minimum protection required for standard system directories. Table 5-3 presents the minimum access required for special products. (You may have all, some, or none of these on your system.)

PRIORITY ACCESS

Operators and System Administrators occasionally need special access to all files and directories on a disk. This happens, for example, when backups are to be done.

This special access is created by priority ACLs. (Priority ACLs may be set either on password-protected or on ACL-protected disks.) This final portion of the chapter discusses priority ACLs and explains the commands used to set, remove, and list them.

SETTING PRIORITY ACCESS

The command that sets priority access is:

```
{SET_PRIORITY_ACCESS}  partition-name  access-control-list
{SPAC               }
```

The access-control-lists within priority ACLs use the same identifiers, access rights, and general formats as do the regular ACL commands. However, there are these differences between priority ACLs and regular ACLs:

- The System Administrator can set or remove priority ACLs from any terminal. Any administrator or operator can set or remove them from the supervisor terminal.
- Unlike regular ACLs, priority ACLs do not contain an implied $REST:NONE. If you want to exclude all users not mentioned in the priority ACL, you must include the $REST:NONE in the command line explicitly. (Including $REST:NONE denies $REST all access to the disk.)
- Priority ACLs may be set both on ACL-protected and on password-protected disks.
- Priority ACLs take precedence over other ACLs.

Priority ACLs may be either inclusive or exclusive. That is, priority ACLs may either add some special access to the access rights that already exist on the disk or they may entirely replace the current access rights on the disk.

You might want to use an inclusive priority ACL if a pair of analysts were doing some trouble-shooting. The command:

```
SET_PRIORITY_ACCESS HOLMES:ALL WATSON:ALL
```

would give the troubleshooters all rights to all directories on the disk. The rights of other users to the files and directories on the disk would not be disturbed.

On the other hand, if you are about to back up the disk, you might give the command:

```
SET_PRIORITY_ACCESS SYSTEM:LUR $REST:NONE
```

Only SYSTEM would then have any rights at all to the disk until SYSTEM removed the priority ACL with the REMOVE_PRIORITY_ACCESS command. No one else could access the disk in the meantime.

REMOVING PRIORITY ACLS

The command that removes a priority ACL from a disk is:

```
{REMOVE_PRIORITY_ACCESS}  partition-name
{RPAC                  }
```

This command may be given by the System Administrator from any terminal. It may be given by any administrator or operator from the supervisor terminal.

LISTING PRIORITY ACLS

When a priority ACL is in effect for a partition, its contents are displayed in a LIST_ACCESS command. However, since the priority ACL may prevent users from accessing any part of the disk, the LIST_PRIORITY_ACCESS command allows listing of the priority ACL on any partition at any time. Its format is:

```
{LIST_PRIORITY_ACCESS}  partition-name
{LPAC                }
```

Table 5-2
Access Rights Needed for System Directories

Directory	Minimum Access Needed
BATCHQ	(protection set by Batch subsystem)
CMDNC0	$REST:LUR
DOS	SYSTEM:LUR
HELP*	$REST:LUR
LIB	$REST:LUR
LOGREC*	SYSTEM:ALURW
MFD (on command disk)	$REST:U
PRIMENET*	NETMAN:ALL SYSTEM:UR
PRIRUN	SYSTEM:LUR
SAD	(protection maintained by EDIT_PROFILE)
SEGRUN*	$REST:LUR
SYSOVL	$REST:LUR
SYSTEM	SYSTEM:LUR

In addition, the following rights may be desirable:

CMDNC0	ALL rights for System Administrator
LIB	ALURW for anyone modifying the libraries
LOGREC* PRIMENET*	ALL rights for operators, so that they can control event logging files
INFO	$REST:LUR

Table 5-3
Access Rights Needed for Special Products

Product Name	Directory	Minimum Access Required
DBMS/QUERY	VISTA	System Administrator:ALURW $REST:NONE
	VISTASRC	System Administrator:ALURW $REST:NONE
	VISTA*	Should be a password protected UFD
FED	FED*	$REST:R Installer:ALL
FORMS	FORMS*	$REST:ALL
FTS	FTS	System Administrator:ALL
	FTSSRC	System Administrator:ALL
	FTSQ*	SYSTEM, YTSMAN, and FTS Servers:ALL $REST:DALURW
POWER	POWER*	$REST:ALL
	POWRCM	$REST:ALL
PRIMENET	PRIMENET*	SYSTEM:UR NETMAN:ALL OPERATORS:ALL
	FAM	SYSTEM:UR FAM:ALL

Table 5-3 (continued)
Access Rights Needed for Special Products

Product Name	Directory	Minimum Access Required
RJE	RJSPLQ*	Operator:ALL User:ALL
	RJSPLQ*>CMDHELP RJSPLQ*>ERRHELP	Operator:LUR User:LUR
	RJSPLQ*>BINARY RJSPLQ*>PUNCH RJSPLQ*>Qxxx RJSPLQ*>SAVE RJSPLQ*>SDRF RJSPLQ*>TQ_ROUTE	Operator:DALURW User:None
	RJSPLQ*>CMDNCO SYSCOM	Operator:None User:None

6
Disks

INTRODUCTION

Creating work areas for your system's users, allocating disk space to users, and monitoring the use of that space are important tasks. Before your disks can be used for reading, writing, and updating information, the disks must conform with your system's requirements and your users' needs.

Providing optimum efficiency and security for your disk space requires important decisions and responsibilities in setting up and monitoring disk space. These include:

- Knowing your disks' memory capacities and read/write capabilities.
- Deciding how to divide your total disk space into subdivisions called "partitions" or "logical disks," which function as disks themselves, and how to distribute those partitions on your system.
- Creating disks and disk partitions to conform with your system and user needs. This creation, called "formatting," sets up disks for the storage of a user's work and disks for "paging" — temporary storage of data that is used by the processor at any given time.
- Deciding how to allocate paging space -- when and how to create a partition that will be used only for paging or a "split disk" partition, which uses part of your partition for paging.

- Deciding how to allocate space within partitions -- setting limits, called "quotas," on the number of records that are allocated to each top level directory.
- Deciding what to do when disk space gets crowded.
- Monitoring the use of disk space through various PRIMOS commands and utilities.

What This Chapter Discusses

This chapter discusses the following subjects:

- The types of disks Prime supports
- Deciding how to divide disks and distribute partitions
- Formatting disk partitions
- Allocating paging space
- Allocating space (quotas) within partitions
- What to do when your disk gets crowded
- Monitoring the use of disk space

References to Other Books

In addition to other parts of the System Administrator's Guide, this chapter will refer to the System Operator's Guide, the Prime User's Guide, and the PRIMOS Commands Reference Guide for more information. You should have all of these books available for quick reference and detailed procedure.

LIST OF DISKS PRIME SUPPORTS

Prime currently supports four categories of disks at Rev. 19, each with varying storage capacities. These four categories are:

- Cartridge Module Disks (CMDs)
- Fixed Media Disks, also called "Winchester" disks
- Storage Module Disks (SMDs)
- Diskettes (Floppy Disks)

Cartridge Module Disks (CMDs)

Prime supports CMDs, or "cartridge disk drives," with three storage capacities: 32, 64, and 96 million bytes (megabytes) of memory. Each CMD is made up of one removable cartridge platter (two surfaces) and one, three, or five fixed surfaces, respectively, that are permanently attached to the cartridge drive itself.

Fixed Media "Winchester" Disks

The Winchester disks -- a technology name that is synonymous with that type of fixed media -- are permanently fixed to and enclosed in air-tight, dust-free drives. Prime supports two storage capacities for Winchesters: 160 and 600 megabytes.

Storage Module Disks (SMDs)

The Storage Module disks are all removable platters enclosed in removable diskpacks. The diskpack is inserted into and removed from its "storage module drive." Prime supports two storage capacities for Storage Module Disks: 80 and 300 megabytes, which have diskpacks with five and 19 usable surfaces respectively.

Diskettes (Floppy Disks)

Floppy disks are physically small, pliable, bendable (hence "floppy") disks that are inserted like cartridges into disk drives that usually support low-memory capacity systems. All diskettes that Prime supports have four sectors per track and a total of 304 records (448 words/record). Therefore, the memory capacity for each diskette is roughly 25 thousand bytes (25K bytes).

For more information about all the disks Prime supports, see the System Operator's Guide.

DECIDING HOW TO DIVIDE DISKS AND DISTRIBUTE PARTITIONS

Deciding how to logically divide your total disk space into partitions, how large or small the partitions are going to be, and how to distribute the partitions to your disk controllers and disk drives are the first steps in "formatting" your partitions.

You have two goals in this process:

- To allocate space as equitably as possible among your users' and the system's needs for space. (This may include reserving some space for future expansion.)

- To spread the workload as evenly as possible among your disk drives and controllers.

Dividing Total Disk Area Into Partitions

As the one most familiar with the nature of your users' work, you are most qualified to create partitions along user-group lines. At the time you create your partitions, you should know:

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

- The nature of each group's work. How much storage space will each group require, given the type of work it produces?

- The workload of each group. Will the workload in each group be light or heavy six months from now? A year from now? How much storage space will each group require in the future?

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

- The number of disk drives and their storage capacities as well as the number of controllers to handle those drives. (A controller is an interface between the CPU and the drive unit.)

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

Large vs. Small Partitions

The following paragraphs offer some guidelines for using large and small partitions.

Advantages of Using Large Partitions: Failing to grant enough disk space to a user partition at the time of the partition's creation can be a common problem. Try to plan ahead when creating new user partitions, especially if your system is new. Allocate partition space so that it reduces the number of times the partition has to be moved, enlarged, or remade. Try to partition all, or nearly all, of your disk space, thereby enlarging the size of the partitions. This approach

will place the extra storage within the partitions, not on the outside surfaces that are not being used.

Planning ahead means being prepared for any group's work to increase or become a "hot project" at your installation. That group's partition would therefore require more disk space. If your system has allotted ample space to each partition at the outset, the "hot" partition could continue its work without having to stop for reformatting or backups.

Ideally, you should know in advance, before creating the partitions, which user groups are likely to have substantial increases in their workloads. If you know that, you can allot more space to those partitions accordingly.

Other advantages of using large partitions include:

- They can hold large data bases.
- They can be more efficient in storage.
- They make it easier to reallocate space among directories.

Advantages of Using Small Partitions: Small partitions can provide a way of protecting confidential data. Creating a small partition can provide one more level of data security in addition to protect rights (in ACL systems) or owner rights (in directory-password systems). Smaller partitions also provide a convenient way of guaranteeing read protection as well as write protection.

Other advantages of using small partitions include:

- They can be safer when something awful happens to your partition. For example, if most or all of the data on your partition is deleted or ruined somehow, more data would be lost in a large partition than in a small partition. (You may not want to put all your data in one basket.)
- They give you more flexibility in deciding how many directories you want to have on-line at any given time.

Note

Previously, many systems used small partitions to control the allocation of space among users. Now that quotas can be set on individual directories (as explained later in this chapter), the use of small partitions to control space usage is no longer necessary.

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

example, an 80-megabyte drive has five surfaces, and a 300-megabyte drive has 19. If you have one 80 and one 300, the only way you can do disk-to-disk backups is to have partitions of two, three, or five surfaces — that is, partitions that could fit equally well on either drive. Larger partitions on the 300 would have to be backed up on tape. On the other hand, if you had two 300 drives, each disk pack could be one partition; the two drives could still back each other up. For more information on backups, see Chapter 13 in this guide and Chapter 7 in the System Operator's Guide.

Distributing Partitions to Drives and Controllers

Whether you are adding new partitions to your system, or adjusting the partitions that already exist, one rule of thumb to follow is this: Try to distribute the use of your partitions evenly among your disk drives, and try to distribute your drives evenly among your controllers. This even distribution makes read/write operations faster and more efficient.

For example, if you have five partitions, two drives, and two controllers, you might want to place the three smallest partitions on one drive and the two largest partitions on the other, thereby evening out the data distribution as much as possible. Then, logically, you would want to place one drive on each controller, so that read/write operations on both drives can occur simultaneously.

On the other hand, if you knew that the smaller partitions received much heavier usage than the larger ones, you might want to divide those between your two drives. Each drive would then hold one large partition and one or two smaller ones, or, one lightly used partition and one or two heavily used ones.

Monitoring the Distribution: Making sure that your partitions and drives are evenly distributed is an ongoing process. You should monitor the data distribution regularly. Watch the trends and patterns in the way your users manipulate their storage space. For example, if one partition's workload increases, more data will be added to the partition, and the read/write operations on that partition will increase substantially. Be prepared to adjust the data distribution so that the increase in read/write operations does not hamper system efficiency.

Three commands that monitor system operations and data storage information are AVAIL, STATUS, and USAGE. These commands, along with other commands and information on monitoring the system, are discussed in the PRIMOS Commands Reference Guide and in Chapter 5 in the System Operator's Guide.

FORMATTING THE DISKS AND PARTITIONS

Once you have decided how to partition your total disk space, the disks must be "formatted" or "made." Bringing a disk into conformance with your system's software addressing method and user needs is called "formatting."

MAKE is the system utility that formats and partitions your disks. MAKE will format both user disks (disks used for the actual storage of a user's work) and paging disks (disks used for temporary storage of data when paging occurs). MAKE will create and structure any form of disk storage supported by PRIMOS. When a disk partition is formatted, MAKE writes the following PRIMOS files and directories to it:

- The Master File Directory (MFD), the top level of the file system that contains all directories and files on the partition.
- The BOOT file, used in bootstrapping the disk.
- The Disk Record Availability Table (referred to as the DSKRAT), containing information about the physical structure of the partition. The DSKRAT file has the name of the partition.
- A badspot file (BADSPT), used to indicate the location of any badspots on the disk. (Badspots are records on a diskpack that cannot hold data.) If there are no known badspots on the disk, the badspot file will not exist.
- An empty directory named CMDNCO, for future storage of the run modules for PRIMOS commands.
- An empty directory DOS, for future storage of the run module for PRIMOS II.

MAKE may be run from a command file, in either PRIMOS or PRIMOS II.

What to Do Before Running MAKE

Prior to running the MAKE utility, you must determine the device number of the physical disk (or disk partition), add the physical device number to the system's Assignable Disks Table, and assign the disk to be formatted to your terminal.

Determining the Physical Device Number: Each physical disk or disk partition has a physical device number. Disk partitions are treated as if they are actual physical devices. The physical device number tells the system the type of storage device being used, the drive unit on which the disk is mounted, and, for partitions, the size of the partition and its location on the diskpack. For complete information on how to determine the physical device number, refer to the tables in Appendix A.

Adding the Number and Assigning the Disk: Before a disk can be assigned, its physical device number must be added to the system's Assignable Disks Table with the DISKS command, which may be given only from the supervisor terminal. After the physical device number has been added to this table, the users or operators can assign themselves the disk by issuing the ASSIGN command.

The Procedure to Be Followed: The exact procedure for determining the physical device number, adding the device number to the Assignable Disks Table, and assigning the disk is listed below:

1. Determine the physical disk number. (See Appendix A.)

2. If you are working in PRIMOS, log in to any UFD located on another physical device.

3. Add the physical device number to the Assignable Disks Table by issuing the DISKS command from the supervisor terminal. The format for adding disks is:

   DISKS pdev-0 [pdev-1] ... [pdev-7]

   pdev-0 ... pdev-7 are physical device numbers. No more than 10 disks may be entered into the Assignable Disks Table. A physical disk number must be specified in this table before a user can invoke the ASSIGN command to assign that disk.

4. To prevent accidental erasure of data on a disk because a physical device number was mistyped, the following procedure is recommended:

   - Under PRIMOS, only the disk to be created by MAKE should be assigned to the terminal.

   - Under PRIMOS II, all running disks should be write-protected except the disk to be created by MAKE. (Most disk drives have a switch labelled WRITE PROTECT. Push this switch.) The DISKS and ASSIGN DISKS commands are not given when MAKE is run under PRIMOS II.

5. If you are working in PRIMOS, assign the disk to be formatted to your terminal. To do this, use the ASSIGN command plus the physical device number. The format for this command is:

   ASSIGN DISK physical-device-number

   DISK is an argument that must be given when assigning a disk or disk partition.

Running MAKE

The MAKE utility is initialized with the MAKE command. After initializing MAKE, a computer-user dialog will ensue, and you will be called upon to respond to several prompts (questions) from MAKE. These prompts will ask you for information such as the physical disk number, the type of storage device, the number of records for paging, the name of the partition, the baud rate, and so on.

For a complete step-by-step procedure for using the MAKE utility, see Chapter 6, FORMATTING DISK DEVICES, in the System Operator's Guide.

What to Do After Running MAKE

After the disk partition has been formatted, it should be unassigned with the UNASSIGN command and should be removed from the Assignable Disks Table with the DISKS command. The sequence of command formats is:

UNASSIGN DISK physical-device-number

DISKS NOT physical-device-number

DISK is an argument that must be given when unassigning disks. NOT is an argument that must be given when removing a disk partition from the Assignable Disks Table. physical-device-number is the number of the partition that is being unassigned or removed.

For a more detailed procedure, see Chapter 6 in the System Operator's Guide.

Pre-Rev. 19 Partitions

You can use Rev. 18 partitions on a Rev. 19 system. However, you may want to convert your Rev. 18 partitions to Rev. 19 to take advantage of Rev. 19 features such as quotas, ACLs (Access Control Lists), and a new badspot-handling method. The FIX_DISK utility, PRIMOS's disk repairing utility, released at Rev. 19, can convert your pre-Rev. 19 partitions to Rev. 19 partitions.

For a complete explanation of FIX_DISK, see Chapter 8, REPAIRING FILE PARTITIONS, in the System Operator's Guide.

It is not expected that you will want to change Rev. 19-format partitions to Rev. 18-format partitions. However, should some circumstance arise in which you feel that such a change is essential, it can be done. See Appendix C for details.

ALLOCATING PAGING SPACE

In addition to user work space and space occupied by the MFD and other important PRIMOS files and directories, you must set aside a certain amount of disk space for "paging."

Paging is a system process that divides large programs and data files into subdivisions called "pages." PRIMOS moves pages between the paging device and main memory as they are needed. (This is known as demand paging.) Paging frees up main memory space for other users and increases the processor's speed and efficiency.

For example, if you were to run a huge program, the processor would divide the program into page blocks, load a page that was needed for processing, and put the other pages out on the disk space that you set aside for paging until the processor needed them. When the processor needs another part of the program, it would take another page out of the paging area and load it into main memory.

Paging space, therefore, can be thought of as a temporary storage area where memory contents sit while waiting to be used by the processor. You are responsible for determining how much space to allocate to your system's paging and for creating the paging partition(s).

One Partition or Two?

Paging can take place on one or two partitions. The first partition (or the only one) is made known to the system through the PAGDEV directive. The second (if used) is made known to the system through the ALTDEV directive. For further information on these directives, see Chapter 3, CONFIGURATION DIRECTIVES.

Since paging forms part of your disks' workload, the choice of where to put paging partitions forms part of the general task of trying to balance the workload across the system. If you need help in making these decisions, your system analyst can advise you.

Paging Space Requirements

Prior to Rev. 18.2, paging space was allocated in segment-sized blocks (128K bytes per block), except for operating system segments, called "kernel" segments, which were truncated to the 16K-byte boundary nearest the actual size. This allocation method wasted space because most segments are much smaller than 128K bytes.

As of Rev. 18.2, the kernel segments are allocated space in the same way — truncated to the 16K-byte boundary that is nearest the actual size. However, all other segments are also allocated space in units of 16K bytes. This means that if only the first eight pages of a segment are accessed, only 16K bytes of paging space are ever used up by the segment, and there is a savings of 112K bytes, compared with Rev. 18.1 or earlier. This change should make it possible to support many more segments within a given paging partition than under previous Revs.

Determining the Amount of Paging Space

The most commonly used and perhaps the best rule of thumb for determining the amount of space for your paging partition is:

Allocate one paging surface for every 10 to 12 users.

The number of users is the sum of all human (terminal) users plus user processes — phantom users, slave users, and remote users.

A more complicated, precise formula for determining worst-case (maximum) paging requirements at Rev 19.0 is:

pr = (1536 + (nusr)*80 + nuseg * 64)

pr is the total number of paging records required. The number 1536 is the maximum number of records required for 24 kernel segments. The number 80 is the sum of the maximum number of records for Ring 0 stacks (16) plus the maximum number of records for Ring 3 stacks (64). nusr is the total number of users. NUSEG, which is a CONFIG directive, is the number of user segments.

To calculate the total number of paging records required, you can use this six-step procedure:

1. Calculate nusr by adding the values of four CONFIG directives:

   nusr = NTUSR + NPUSR + NRUSR + NSLUSR

   These directives stand for the number of terminal users, phantom users, remote users, and slave users respectively. The octal values of these four directives are given in the CONFIG file, which is located in the CMDNCO directory. These octal values must be converted to decimal before you can do the calculation. If the value of any of these variables is not specified in the CONFIG file, use the default value, given in Chapter 3, CONFIGURATION DIRECTIVES.

2. Multiply nusr by 80. (Call the product Result A.)

3. Take the value of the CONFIG directive NUSEG (in decimal) and multiply it by 64. (Call the product Result B.) The value of

NUSEG is also given in the CONFIG file within CMDNCO, and its default value is also listed in Chapter 3.

4. Assume that PRIMOS needs a maximum of 1536 records.

5. Calculate pr (total number of paging records) by adding Result A, Result B, and 1536:

   pr = A + B + 1536

6. Calculate the number of surfaces you need to hold pr records. There are 7407 records per surface.

Split Disks

A split disk is a disk partition whose surface is "split" between paging space and user space -- paging space that takes up only part of a disk surface, and therefore, only part of a partition.

If you use storage module disks for paging, full partitions are slightly preferred over split ones. With a split disk, read/write seek time increases because the read/write head on a split disk surface has to jump back and forth from the inner surface (paging area) to the outer surface (user area). Creating a paging partition unto itself is more efficient.

With Winchester disks, on the other hand, you are more likely to use a split disk, in order to take advantage of the badspot handling split disks provide.

When to Split a Disk: There are two general circumstances where it would be appropriate to split a disk:

- If your system is very small, and the amount of paging space that is needed does not come close to taking up a full surface.

- If your disk surface has badspots. (A badspot is a defective spot on the disk surface that cannot hold data.) PRIMOS can handle 16 badspots in paging space; but it cannot write the badspot file that handles them in the paging area itself. Therefore, the disk is "split"; paging takes place on one portion and badspot handling and other storage take place on the other portion.

How to Split a Disk: When you want to split a disk surface and use part of that surface for paging, answer YES to the MAKE utility's SPLIT DISK? prompt. A step-by-step procedure for answering prompts in the MAKE utility dialog is listed in Chapter 6 of the System Operator's Guide.

ALLOCATING SPACE WITHIN PARTITIONS — QUOTAS

Ensuring equitable sharing of disk storage among users is a primary function of the System Administrator. At Rev. 19, you can provide that equity by setting limits on the amount of storage space that directories occupy on a disk. These limits are called "quotas."

The quotas, which are measured and allocated by the number of disk records, can be set by both the System Administrator and the user. As the System Administrator, you are responsible for setting and modifying the quotas on top-level directories. The users, in turn, take your quota limits and set and modify quotas on their own subdirectories.

Note

You cannot place a quota on an MFD.

The User's Perspective

Once you have allocated quotas to your system's top-level directories, the users can set or modify quotas on subdirectories only if they have Protect rights (in ACL directories) or owner rights (in directory-password systems) to the next higher directory. That is, the user must have the appropriate rights to the directory that contains the subdirectory whose quota is to be set. Instructions and guidelines for the user on setting and modifying quotas -- the user's perspective — are given in Chapter 17 under USING DISK QUOTAS in the Prime User's Guide.

Guidelines for the System Administrator

After you have determined the number of records on the disk partition that can be reserved for users -- the amount of disk space that remains after allocating space to mandatory PRIMOS files and directories and paging — there are a number of strategies you can use to distribute and manipulate your users' disk space. These strategies involve deciding how to set the quotas on your top-level user directories. (Users set quotas on their own subdirectories if they have Protect/owner rights.)

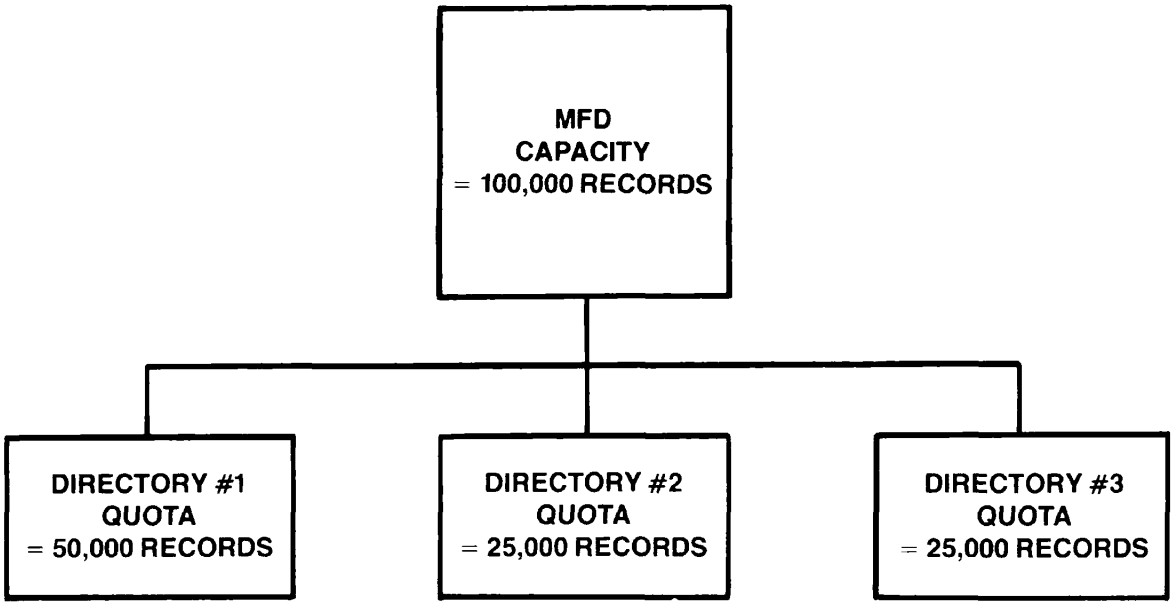
In general, you would set the quotas on top-level directories according to how loosely structured or strictly structured you want your user space to be — whether you want to set strict limits on each user group or whether you want to give them more records than they need so they can compete for the disk space.

There are four major strategies to consider:

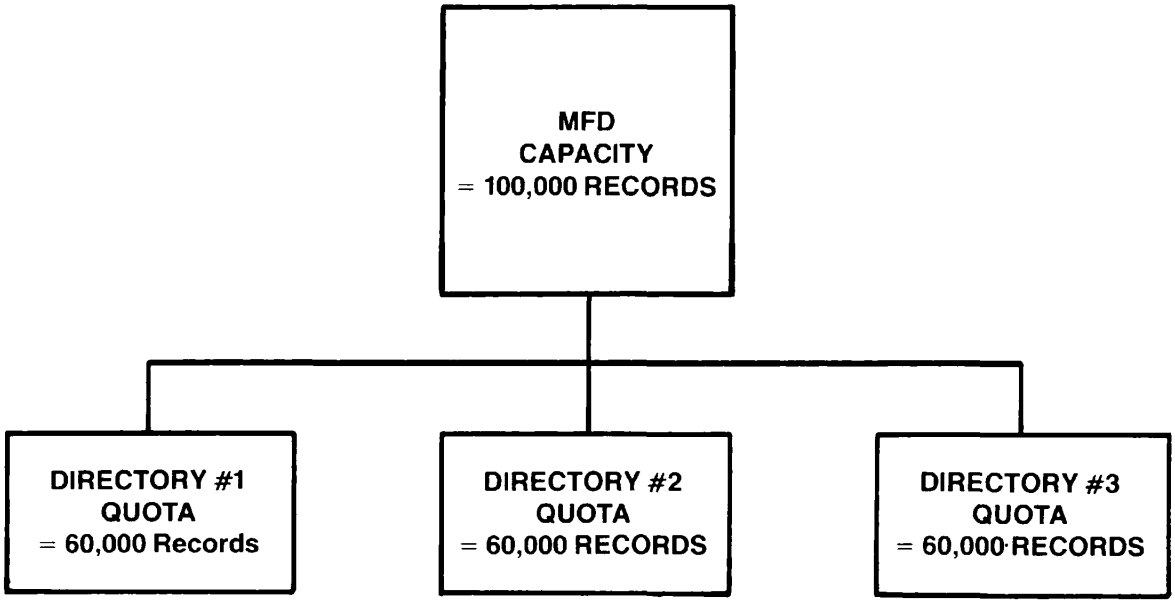
- The Conserved strategy — guaranteeing the partition's quota limit by dividing up the exact number of user records among the top-level directories.
- The Overcommitted strategy — maintaining competition among users by setting the total number of records on the directories above the capacity of the partition.
- The Undercommitted strategy — reserving space by setting the directory quotas below the record capacity of the partition.
- The Unregulated strategy — setting no quota — infinite (unlimited) quota — on one or more directories.

The Conserved Strategy: Suppose your partition (MFD) had a capacity of 100,000 records that were earmarked for user's work space. Taking a strict approach, you could "conserve" that space by guaranteeing that your users never use up more than 100,000 records. If you had three top-level directories, for example, you might want to give one directory 50,000 records and give the other two 25,000 each, according to which user group needed more space. (Figure 6-1 illustrates the Conserved strategy.) After setting the quotas you would subsequently monitor which top-level directories were using their space and modify the quotas accordingly.

The Overcommitted Strategy: Taking the same example, a partition with 100,000 records, you may want to prevent your users from underutilizing their disk space and create competition among the users. Competition for disk space can be maintained within a quota system by setting the quotas above the record capacity of the partition. Under this strategy, users would be more inclined to use as much space as they needed without feeling restrained by the limits. With a 100,000-record capacity, you might want to allocate each top-level directory 60,000 records. (Figure 6-2 illustrates the Overcommitted strategy.) This strategy is particularly useful if you know your system has more than enough space to handle all your users' needs.

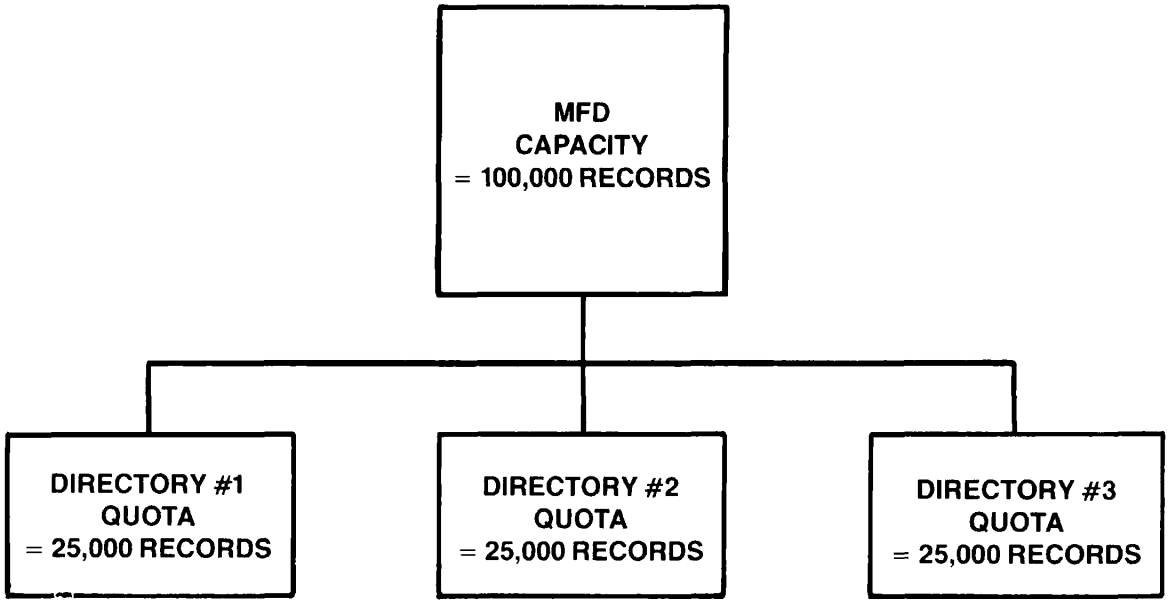


The Conserved Strategy
Figure 6-1



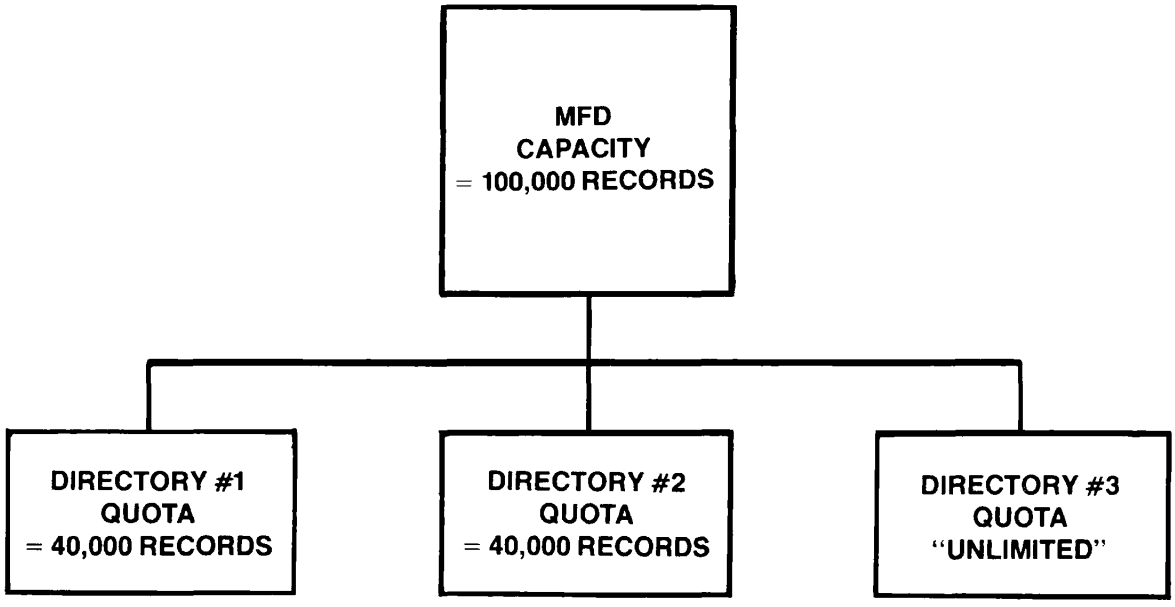
The Overcommitted Strategy
Figure 6-2

The Undercommitted Strategy: This strategy is the most strict and generally has the opposite effect of the Overcommitted strategy. When your system is tight for space and it is possible for the users to exceed the disk space capacity of the partition, you can reserve space by setting the quotas below the partition's capacity. This strategy creates an incentive for the users to be more efficient, reserving their space for essential data and deleting the "deadwood" data. It also guarantees extra space on the system that could be used for emergency storage. So with the 100,000 record capacity, for example, you could set a quota of 25,000 records on each directory. (Figure 6-3 illustrates the Undercommitted strategy.)



The Undercommitted Strategy
Figure 6-3

The Unregulated Strategy: The least rigid of these strategies is the Unregulated strategy, where no quota is set on one or more directories. When no quota limit is set on a directory, its storage capacity is limited only by the physical capacity of the partition. Setting no quota on a directory gives the impression to the users that their allotment of disk space is infinite and unlimited. This strategy would be employed, for example, if you had a special user group, which, by the nature of its work, would be trusted with an "unlimited" amount of disk space. With a 100,000 record capacity, for example, two of your directories could each be set at 40,000 records, and the third would have no quota set. (Figure 6-4 illustrates the Unregulated strategy.)



The Unregulated Strategy
Figure 6-4

How to Set, Modify, and Examine Quotas

The commands that allow a System Administrator or a user to set, modify, and examine quotas are as follows:

- The SET_QUOTA command (abbreviated SQ) sets the maximum storage quota on a directory.
- SET_QUOTA is also used to change an existing quota.
- The LIST_QUOTA (abbreviated LQ), LD, and SIZE commands are used to examine existing quotas and current storage use.

The following paragraphs explain how a System Administrator sets, modifies, and examines quotas. For more information, see Chapter 17 of the Prime User's Guide.

Measuring and Allocating Storage Space: Storage space is measured in disk records. A record can contain up to 2048 bytes (two bytes/word). Thus, the number of records in a file system object equals the total number of bytes in the object divided by 2048 and rounded up to the next whole number. However a zero-length object (such as an empty directory or file) always contains one record. All numbers are decimal.

If you create a directory or subdirectory, its quota will initially be set to zero; that is, it has no maximum quota. However, any maximum quota that may exist on a higher directory will limit the actual storage allowance on the subdirectory. If no limit exists on any higher directory, its storage capacity is limited only by the record capacity of the partition. A quota of zero, in other words, means that the storage is "unlimited" — limited only by the capacity of the higher directory or the capacity of the partition itself.

Setting Quotas on Directories: To set maximum storage quotas on directories and subdirectories, use the SET_QUOTA command. The format is:

{SET_QUOTA / SQ} pathname -MAX number

The pathname is the pathname of the directory having its quota set. If you want to set a quota on the current directory, you must use the full pathname. If you want to set a quota on a subdirectory within the current directory, you need specify only the simple name (the final element of the full pathname).

-MAX number is the maximum number of records the directory can use.

If a user attempts to use the SET_QUOTA command without having Protect or owner rights to the directory, PRIMOS returns the message:

"Insufficient access rights"

and the quota will not be set.

If you try to set a quota on a directory when the directory has no current quota (quota = 0) and when there are attached users or open files in the directory or its subtree, you will receive the error message:

File in use. directory-name (set_quota)
ER!

(See Chapter 17 in the Prime User's Guide for details.)

Note

This restriction may make it difficult for users to set quotas on their origin directories. To make things easier, you may want to set the initial quotas on users's origin directories yourself, before the users log in to them.

Modifying Quotas on Directories: Once a quota exists on a directory, you may raise, lower, or remove it with a new SET_QUOTA command. The format for raising or lowering a quota is the same as for establishing a quota on a non-quota directory:

SET_QUOTA pathname -MAX number

You may remove an existing quota from a directory by setting the quota to zero. Any of the following command formats will set a directory quota to zero:

SET_QUOTA pathname

SET_QUOTA pathname -MAX

SET_QUOTA pathname -MAX 0

Examining Quotas and Current Storage: You may wish to examine the quota on a directory and the current storage space used by directories, files, and segment directories. The LIST_QUOTA, LD, and SIZE commands provide this information.

Using LIST_QUOTA: The LIST_QUOTA command tells you the maximum quota on a directory, the total number of records used by the entire subtree beginning with and including the designated directory, and the number of records used by this particular directory. The format of the command is:

```
{ LIST_QUOTA } [pathname]  [-BRIEF]
{ LQ         }
```

pathname gives the name of the directory on which quota information is requested. If pathname is omitted, the quota information on the current directory is listed. The user must have List access to the target directory or its parent and Use access to all higher directories.

For example, to list the quota information on the current directory REPORTS:

```
OK, LQ

Maximum records allowed on "<Current directory>" = 200.
Total records used = 178.
Records used in this directory = 65.
OK,
```

For more information, see the Prime User's Guide.

Monitoring Quotas with LD and SIZE: The LD and SIZE commands can also supply information on quotas and record usage. For more information on these commands, see the Prime User's Guide and the PRIMOS Commands Reference Guide.

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

See Chapter 17 in the Prime User's Guide for explanations and illustrations on how to calculate storage availability.

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

For more information on how to recover from quota overloads and overloads that occur during an editing session, see Chapter 17 of the Prime User's Guide.

WHAT TO DO WHEN THE DISK GETS CROWDED

When your disk space gets crowded, there are several options available to a System Administrator to alleviate the problem. Depending on which or how many directories are crowded, and depending on how badly your users need space, you should consider doing one or more of the following:

- Get more disks at your installation.
- Instruct the users to tighten up their space by deleting outdated or obsolete files.
- Put some of the outdated or obsolete files on magnetic tape, or instruct your users to put those files on tape. (Build a tape archives.)
- Move user groups out of one partition and into another. This move requires changing the users' Initial Attach Points via EDIT_PROFILE.
- Compress UFD space by using the FIX_DISK utility. (The FIX_DISK utility is fully explained in Chapter 8 of the System Operator's Guide.)
- Adjust the quotas on your top-level directories.

Adjusting the Quotas

Here are some strategies for adjusting the quotas when your disk space gets crowded:

- If you can afford to do it, that is, if you have employed an undercommitted quota strategy, increase the quota limit for the directories that are most in need of extra space.
- Reset the quotas on the top-level directories across the board. This strategy takes extra space away from a directory that may have ample space and gives 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, you would set the quota below that figure -- perhaps to 15,000. This very strong measure would be used as an incentive for users to delete unneeded data and to become more efficient in their use of space. The 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. (This strategy should really be used as a last resort.)

MONITORING THE USE OF DISK SPACE

Knowing how to distribute your storage space over the system and to your users, and knowing which strategy to employ in adjusting the quotas on top-level directories, means regular monitoring of system operations and the way your users manipulate their disk space. There are several PRIMOS commands and utilities available for system monitoring, including the commands AVAIL, STATUS, and USAGE. You can use these commands along with the commands that modify and examine quotas. For complete information on all of these commands and on guidelines for monitoring the system, see the PRIMOS Commands Reference Guide and Chapter 5 of the System Operator's Guide.

7
Allocating System Resources

INTRODUCTION

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

- A list of system and network default configuration values along with the corresponding directives to alter each value.
- A list of shared segments — those to which Prime has assigned products, those reserved for Prime, and those specifically reserved for customer use.
- A description of shared libraries and a list of the shared library package numbers.

SYSTEM AND NETWORK PARAMETERS

Configuration Defaults

Default values for many parameters are established by the operating system upon start up. These values can be altered by including the appropriate directives in the configuration file. The parameters, along with their defaults and the directives to alter these defaults are given in Table 7-1. (See Chapter 3 for details.)

Table 7-1
Configuration Parameters and Directives

	Parameter	Default	CONFIG Directive
	ABBREV processor	YES	ABBREV
	AMLC line input buffer	'140	AMLBUF
	AMLC line output buffer	'300	AMLBUF
	AMLC programmable clock baudrate	9600	AMLCLK
	ASR terminal input buffer	'200	ASRBUF
	ASR terminal output buffer	'300	ASRBUF
	Carrier check operations interval	2	AMLTIM
	Configure network	NO	NET ON
19.0|	DMC Input buffer	'60	AMLIBL
	DMQ AMLC buffer	'40	AMLBUF
	File system read/write lock	1	RWLOCK
	Inactivity before forced logout (seconds)	'1750	LOUTQM
	Login while logged in allowed	YES	LOGLOG
	Logout on AMLC line disconnect	NO	DISLOG
	Max. per-user guaranteed file units	'20	FILUNT
	Maximum per-user file units	'200	FILUNT
	Min. grace time for terminal lines	0	AMLTIM
	Modem disconnect operations rate	1800	AMLTIM
	Number of prepaged pages	3	PREPAG
	Phantom users, number	0	NPUSR
	Print configuration directives	NO	TYPOUT
	Print LOGIN/LOGOUT messages	YES	LOGMSG
	Remote users, number	0	NRUSR
	Restart after power failure	NO	UPS
	Segments per user process	'40	NUSEG
19.0|	Slave users, number	0	NSLUSR
	SMLC lines	OFF	SMLC
	Supervisor terminal baud rate	'300	ASRATE
	System erase character	"	ERASE
	System kill character	?	KILL
	Total virtual address space (segments)	'1776	NSEG
	Wired memory size printout	NO	WIRMEM

SHARED SEGMENTS

Normally, shared subsystems will be incorporated into PRIMOS at system startup time. At times, experimental subsystems may need to be incorporated for test purposes. The command sequence for this, from the supervisor terminal is:

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

pathname	The file to be restored into segment-number.
segment-number	The segment to be shared. See Table 7-2 for a list of segments specifically reserved for customer-shared subsystems.
access-rights	User access to the segment. Default is '600 — read and execute rights.

See the System Operator's Guide for complete details. The System Administrator will assign and coordinate shared segment usage.

> **Caution**
>
> It is possible to overwrite the operating system and the shared utilities with this command. Do not share into segments 0 - '1777. Segments 0 to '1777 are reserved for PRIMOS. Other segments that may contain system utilities are described in Table 7-2.

Table 7-2
Contents of Shared Segments

Segment	Product
2000	Editor (0-160000)
2001-2003	DBMS
2004-2011	SPSS (note 4)
2012	DBMS
2013	BASIC/VM
2014	Shared libraries (note 1)
2015	DPTX
2016	COBOL
2017	BASIC/VM
2020	MIDAS writable shared segment
2021	FORMS library
2022-2023	Reserved for Prime
2024-2025	PRIME/POWER
2026-2027	Reserved for Prime
2030-2037	Reserved for customers
2040-2042	DBG
2043	SPSS (note 4)
2044-2045	PL/I-G
2046-2047	FORTRAN 77
2050	V-FTNLIB
2051	PL/I-G
2052	FORTRAN 77
2053-2056	Reserved for Prime
2057-2063	OAS
2064-2066	PASCAL
2067	FORTRAN 77
2070	DBMS
2071	OAS
2072	SPSS (note 4)
19.0| 2073-2077	DBMS/QUERY (VISTA)
2100	Reserved for Prime
19.0| 2101	OAS
2102-2114	Reserved for Prime
19.0| 2115	DBG
2116-2127	Reserved for Prime
19.0| 2130-2137	MEDUSA
2140	Reserved for Prime
2141-2143	EMACS
19.0 2144-2146	VRPG
2147-2150	EMACS
2151-2152	FED
2153-2166	Reserved for Prime
2167	SPOOL
2170-2177	Reserved for customers
19.0 2200-2207	Reserved for Prime
2210-2216	TAPS (note 4)
2217-2267	Reserved for Prime
2270-2276	INFORMATION

Table 7-2 (continued)
Contents of Shared Segments

Segment	Product
2277	Reserved for Prime
2300-2317	Reserved for customers
2320-2377	Reserved for Prime
6001	Per-user linkage segment (note 2)
6006	Per-user linkage segment (note 3)

19.0

Notes to Table 7-2

1. Segment 2014

Allocated	Product
100-277	COBOL library (VCOBLB)
300-377	MIDAS library (VKDALB)
1000-37777	COBOL library (VCOBLB)
40000-177777	MIDAS library (VKDALB)

2. Segment 6001

Allocated	Product
0-32777	FORMS
33000-40777	COBOL (VCOBLB)
41000-66777	MIDAS (VKDALB)
67000-67770	SPOOL
67770-67777	BATCH
70000-105777	FORMS
106000-112777	ED
113000-117777	NPX
120000-131777	ABBREV
132000-177777	V-FTNLIB

19.0

3. Segment 6006

Allocated	Product
0-163777	Reserved for Prime

19.0

4. Third-party Software

SPSS and TAPS, although third-party software, have been assigned shared segments.

SHARED LIBRARIES

There is a maximum of 16 shared libraries per system (see Table 7-3.) The FORTRAN library (I/O routines only) and the MIDAS library are supplied to all users. Other libraries (see Table 7-3) are supplied to those users who have purchased that particular software product.

19.0

Table 7-3
Shared Library Package Numbers

Package Number	Library
1	V-FTNLIB
2	VKDALB
3	VCOBLB
4	VFORMS
5	DBMSLB
6	OAS
7	OAS
10	Reserved for Prime
11	Reserved for Prime
12	SPOOL
13	Reserved for Prime
14	Reserved for Prime
15	Reserved for Prime
16	Reserved for Prime
17	INFORMATION
20	Reserved for Prime

Installation of the shared libraries is the default. Small systems with few users which have only one MIDAS user or one COBOL user or where the FORTRAN formatted I/O routines are seldom used, may see no benefit from shared libraries.

Features of Shared Libraries

Each user of shared library routines uses private segment '6001 in addition to the segments otherwise required by programs. Segment '6001 is used for the impure portion of the shared libraries and represents a reduction in the size of the user's load file but not in the size of the single user working set at run time. This additional segment may be compensated for by a corresponding reduction in the number of segments in the run file. (The MI option of SEG's Loader is used for reducing the number of segments; see LOAD and SEG Reference Guide.)

Several benefits result from using the shared libraries:

- User run files will be smaller, reducing the time required to restore the SEG runfile. User interaction with the program will begin sooner.

- System load will be reduced with respect to private segments and private memory image sizes. If properly used, paging may be reduced. This is important for users with many large V-mode and I-mode programs making extensive use of the shared library routines.

- Installation of a new revision of the library will not require program reloading. Installation of a rebuilt shared library is all that is required to make the modified library available to all users of the shared library.

There must be no active users of a library when that library is being reshared. To insure this, shut down PRIMOS and then reboot it when installing a shared library.

Installation

Shared libraries must be installed each time the system is cold started. The runfiles are resident in UFD SYSTEM of the Master disk. Copy these runfiles to UFD SYSTEM on the system disk. The commands that install the runfiles at startup time may be incorporated into the C_PRMO file (as in the example in the System Operator's Guide) or called from C_PRMO by the COMINPUT command. The commands are included in C_PRMO.TEMPLATE in UFD PRIRUN. Chapter 2 gives the command files necessary to install shared libraries.

This command file installs memory image files in the proper segments (see Table 7-2) and runs the programs required to inform the operating system that shared libraries are activated. Once the libraries are installed, users with programs loaded using the special shared library object files may run V-mode and I-mode programs accessing these shared libraries. If the shared libraries are not installed, programs expecting the shared libraries to be resident will get an error message from the operating system whenever an attempt is made to access a shared library routine.

Shared Library Usage

If one of the shared libraries is to be used, all appropriate shared libraries must also be used. If the user wishes to use the shared FORTRAN library and also requires MIDAS or COBOL, the shared MIDAS and COBOL libraries must also be used. After the new V-mode run file has been created, and the shared libraries installed, the user's programs may be run as before.

The system (or FORTRAN) libraries (in segment '2050) and the spool (VSPOO$) library (in segment '2167) should always be shared. Other libraries may be shared as desired.

Administration

The shared libraries files are in UFD LIB. UFD LIB must be on logical disk 0 if SEG is to operate properly. If it is not on logical disk 0, SEG will return a "Not found" message to commands such as LIBRARY and SPLIT.

If the shared libraries are not to be used system-wide, then those users planning to use them must modify their command files to use the non-shared library files.

Rebuilding and Reinstallation

Each of the shared libraries is represented by a set of runfiles and an installation program. If only one of the libraries must be replaced, it is necessary to rebuild that library only. These command files put all the necessary files into UFD SYSTEM so that installation is easily accomplished by running the appropriate command file.

Caution

A library should not be replaced while being used. As programs using the shared libraries execute, links are made to the appropriate shared library routines in such a way that altering the memory image in use by the program can cause random and unpredictable behavior. Changing a shared library has the effect of making such an alteration to the user's memory image. Install new shared libraries only when bringing up the system with a cold start.

It is safe to replace the memory image files in UFD SYSTEM at any time as these are loaded into memory only when the explicit SHARE commands are given.

8
AMLC Lines

INTRODUCTION

Assignable AMLC lines are used for serial devices, such as serial printers. There are four steps involved in setting up such lines:

1. Use the NAMLC directive to define the number of buffers available for assignable lines. This directive is discussed in Chapter 3.

2. It may be necessary to change the characteristics of the buffers from their default characteristics. To do this, you would use the AMLBUF directive, as explained in this chapter.

3. Use the AMLC command to define which lines are assignable, as explained in the second part of this chapter.

4. Use the ASSIGN command to assign the line. This is discussed briefly at the end of this chapter. It is also discussed in the System Operator's Guide and in PRIMOS Commands Reference Guide.

AMLBUF AND RING BUFFER ASSIGNMENTS

The AMLBUF Directive

The AMLBUF directive sets the I/O buffer sizes for terminals and assigned lines. The format is:

```
AMLBUF line [in-buff-size] [out-buff-size] [dmq-size]
```

line	The AMLC line number for which buffer sizes are to be set. For terminal users and assignable lines, this value is the physical line number.
in-buff-size	The terminal input buffer size in words (two characters per word). If 0 is specified, buffer size is unchanged. The default value is '200 (128 decimal). This buffer is usually left unchanged.
out-buffer-size	The terminal output buffer size in words. If 0 is specified, buffer size is the default value, '300 (192 decimal). The default value is usually adequate for most applications.
dmq-size	The size in words for the DMQ. If omitted or specified as 0, its value is the default value '40 (32 decimal). This buffer is usually set to '200 (128 decimal) for terminal users whose lines are running at 9600 baud. If the DMQ is set to the default, the character speed will be <= 300 characters per second. If it is set to '200, the character speed will be <= 1280 characters per second.

Ring Buffer Assignments

The AMLBUF directive can be confusing since the ring buffer assignments do not necessarily represent the ring buffer sizes for the command line on which they appear. For example:

		Ring Buffer		
	Line	Input	Output	DMQ
AMLBUF	0	0	0	200
AMLBUF	1	0	0	200
AMLBUF	2	0	0	200
AMLBUF	3	0	0	200
AMLBUF	4	2000	2000	400
AMLBUF	5	0	0	400
AMLBUF	6	0	0	10
AMLBUF	7	0	0	0
AMLBUF	10	2000	2000	0
AMLBUF	11	2000	2000	0

In the group of AMLBUF directives listed above, line number 5 is used for a serial printer. The ring buffers for that assigned line do not appear on the same line. Instead, the buffers will be allocated from a rotating buffer pool that will use the buffer values from lines 10 or 11, depending on which is available. Assigned lines and how to ensure a desired ring buffer setting are described next.

Assigned Lines: When more than one assigned line is configured, it is not possible (except at cold start) to predetermine which ring buffers a particular assigned line will receive. This is because assigned lines use a rotating buffer pool system that does not guarantee the same buffers will be used when lines are unassigned and later reassigned. If you want to configure a device that requires special ring buffer sizes, the only way to be sure the device will receive them is to configure all assigned lines identically.

In the following example, two assigned lines are needed. The first assigned line (line 5) needs a large output ring buffer and a large DMQ buffer. The second assigned line (line 6) needs a large input ring buffer and a small DMQ buffer.

Example:

Device on Line	Ring Buffer Input	Output	DMQ Buffer	Comments
5	0	2000	400	This device needs a large output ring buffer and a large DMQ buffer.
6	2000 / 2000 Common Setting	0 / 2000 Common Setting	10	This device requires a large input ring buffer and a small DMQ buffer.

Each of the input and output ring buffers must be set to the special case required.

Note

Each AMLC line has a corresponding DMQ buffer. This buffer is not affected by the rotating pool system.

At Rev. 19.0 (and Rev. 18.4) it is possible to set DMQ buffers for all AMLC assigned lines. Previously, if the AMLBUF directive corresponding to the line could have changed the size of remote login ring buffers, the directive was not implemented and an error was reported.

Given the above parameters, the two assigned lines need the following buffers:

	Ring Buffers		
Line	Input	Output	DMQ Buffer
5	2000	2000	400
6	2000	2000	10

Once the common ring buffer assignments for the assigned lines have been derived, they, along with terminal and remote login users, may be configured.

Configuring the AMLBUF Directive: Using the configuration outlined below, follow the steps to configure a sample system.

There are five terminal users, three remote login users, and two assignable lines. One of the terminal users (line 4) has a PT65 and needs large ring and DMQ buffers. The other terminal users require standard settings.

Step 1. Determine the number of users in each buffer category.

Parameter	Description	Number of Users
NTUSR	Number of terminal users including the supervisor terminal	6
NRUSR	Number of remote users	3
NAMLC	Number of assigned lines	2

Step 2. Determine the assigned line requirements.

	Ring Buffers		
Line	Input	Output	DMQ Buffer
5	2000	2000	400
6	2000	2000	10

Step 3. Using the "Buffer Matrix" shown below, list the line-by-line DMQ buffer requirements. Disregard the ring buffer portion of the matrix for this step.

Buffer Matrix

Line Number	Ring Buffers Input	Ring Buffers Output	DMQ Size	DMQ Requirements (Line-by-line)
0			200	User terminal with a DMQ setting of '200 (128 decimal)
1			200	User terminal with a DMQ setting of '200 (decimal 128)
2			200	User terminal with a DMQ setting of '200 (decimal 128)
3			200	User terminal with a DMQ setting of '200 (decimal 128)
4			400	PT65 terminal with a large DMQ buffer setting.
5			400	Assigned line with a large DMQ buffer.
6			10	Assigned line with a small DMQ buffer.
7			0	Ignore DMQ buffer settings for remote lines. They will be set by the system they are on.
10			0	
11			0	

Note

DMQ buffer sizes are set for the line to which the device will be connected.

Step 4. For now, disregard the DMQ buffer information just placed into the matrix and fill in the ring buffer requirements according to the "Buffer Arrangement" indicated.

Ring Buffers

Buffer Arrangement	Line Number	Input	Output	DMQ Size	Ring Buffer Requirements
Terminal Users	0	0	0		First terminal with default settings
	1	0	0		Second terminal with default settings
	2	0	0		Third terminal with default settings
	3	0	0		Fourth terminal with default settings
	4	2000	2000		Fifth terminal with large input and output buffers
Remote Users	5	0	0		First remote user with default settings
	6	0	0		Second remote user with default settings
	7	0	0		Third remote user with default settings
Assigned Line Pool	10	2000	2000		Assigned line pool with large input and output buffers
	11	2000	2000		

Step 5. Combine all buffer information into a matrix and read from left to right to obtain the complete AMLBUF directive.

Buffer Matrix

Ring Buffers

Line Number	Input	Output	DMQ Size	List of AMLBUF Directives
0	0	0	200	AMLBUF 0 0 0 200
1	0	0	200	AMLBUF 1 0 0 200
2	0	0	200	AMLBUF 2 0 0 200
3	0	0	200	AMLBUF 3 0 0 200
4	2000	2000	400	AMLBUF 4 2000 2000 400
5	0	0	400	AMLBUF 5 0 0 400
6	0	0	10	AMLBUF 6 0 0 10
7	0	0	0	AMLBUF 7 0 0 0
10	2000	2000	0	AMLBUF 10 2000 2000 0
11	2000	2000	0	AMLBUF 11 2000 2000 0

Notes

Users can assign line numbers below the number NTUSR (number of terminal users) by zeroing the right-hand byte of the line's lword in the AMLC command (described in the System Operator's Guide). However, if AMLC lines are assigned below that value, some terminal lines will not be available as terminals can use only the lines below NTUSR.

NTUSR + NRUSR + NPUSR + NSLUSR determines the total number of configured users. NTUSR + NRUSR + NAMLC is used to determine the arrangement and number of ring buffers.

See Chapter 3 for complete details on the syntax of the directives and their value ranges.

THE AMLC COMMAND

The AMLC command is used to start up both terminal and assigned AMLC lines. The command format is:

AMLC [protocol] line [configuration] [lword]

AMLC Protocols

The protocol argument must be one of the following: TTY, TRAN, TTYUPC, TTYNOP. The basis for selection of the proper protocol is discussed below.

Note

Protocols for older model AMLC boards (model 5054) are discussed at the end of this section.

TTY: TTY, the terminal protocol, is the default protocol assigned at cold start to lines controlling interactive terminals. With terminal protocol, all input from the terminal is echoed if the line is set for full duplex; a carriage return and a line feed is echoed following carriage return. Bit 8 (the ASCII code parity bit) of each character input from the terminal is forced on. CONTROL-P and BREAK are interpreted as a QUIT if the terminal is connected to PRIMOS as a user terminal. If the terminal is connected to PRIMOS as an assigned AMLC line, CONTROL-P and BREAK are not interpreted. A carriage return input by the terminal is transmitted as a new line (or line feed) to the program requesting input. If the line input buffer becomes full, input is no longer echoed; any further characters typed are lost.

If the System Administrator has enabled the watchdog disconnect timer (via CONFIG's AMLTIM directive), users at terminals may have a limited time to log in after their carrier becomes active. (Locally connected terminals activate the carrier when switched on; remotely connected terminals activate the carrier when the dial-up connection is complete.) If users do not log in within the specified time, their lines are disconnected.

TRAN: TRAN, the transparent protocol, is usually used by lines connected to peripheral devices or other computers. With transparent protocol, no input is echoed, no response is made to the input of a line feed or carriage return, and there is no transformation of carriage return to line feed. CONTROL-P has no special meaning under this protocol. It is simply passed through to the program.

TTYUPC: TTYUPC, the standard-speed translating protocol, is used to avoid sending lowercase output to terminals or peripheral devices that cannot print lowercase characters.

TTYNOP: TTYNOP configures a line to ignore all traffic.

If no protocol is given when using the AMLC or ASSIGN commands, the transparent protocol is assigned by the operating system.

> Caution
>
> The baud rate of the last AMLC line on the last board is the AMLC interrupt rate for the system. The standard rate is 110 (one interrupt every tenth of a second). Changing the baud rate of this line can have adverse effects on the rest of the system. For this reason, it is recommended that users not be allowed to assign this line (and perhaps change its baud rate). Instead, this last line should be assigned to SYSTEM or made non-assignable.

Line

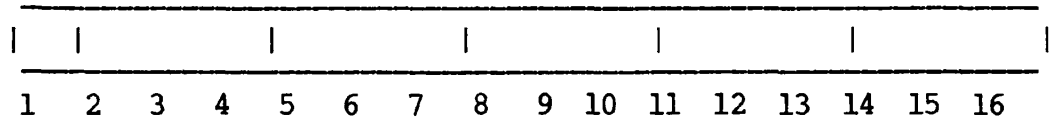
The AMLC line number is an octal number from 0 to the highest line number present in a system.

> Note
>
> Only the first NIUSR-1 lines are for interactive users.

Configuration

The configuration argument, which sets the line configuration, is an octal number that corresponds to the bit pattern illustrated in Figure 8-1. Three commonly used values are shown below. They are for data sets with parity disabled and 8-bit character length.

Config	Baud Rate
2213	300
2313	1200 (default)
2413	9600



Bits	Assignment
1-4	Line number (bit 4 is LSB)
5	Not used
6	Data Set Control Bit (1=on)
7	1=Loop line, 0=do not loop

8-10 Octal Value	Standard Line Speed (Data Rate)
0	110 Baud
1	134.5 Baud
2	300 Baud
3	1200 Baud
4	programmable clock
5	75 Baud*
6	150 Baud*
7	1800 Baud*

Note

Speeds marked with * are assignable by hardware jumpers. The speeds shown are the default values. Other choices are 75, 150, 600, 1800, 2400, 4800, 9600 or 19200 Baud.

If octal 4 is specified, the line speed is that set by the programmable clock (i.e., by the AMLCLK config directive). This directive takes values from 29 to 19200 baud. If no clock speed has been set, the default value of 9600 baud is used.

11	Reserved
12	0= 1 stop bit, 1=2 stop bits
13	0=enable parity, 1=disable parity
14	0=odd parity, 1=even parity
15-16	Character length (does not include the parity bit):

0 0 5 bits
1 0 6 bits
0 1 7 bits
1 1 8 bits

Setup of Line Configuration
Figure 8-1

Lword

Lword, an optional parameter, is a 16-bit octal integer constructed as follows:

Bit			
1	Set	(1)	Line is half-duplex
	Reset	(0)	Line is full-duplex
2	Set	(1)	Do not echo LINE FEED for RETURN
	Reset	(0)	Do echo LINE FEED for RETURN
3	Set	(1)	Recognize XOFF (CONTROL-S, '223) and XON (CONTROL-Q, '221)
	Reset	(0)	Don't recognize XOFF and XON
4	Set	(1)	XOFF seen. Output suspended
	Reset	(0)	Output permitted
5	Set	(1)	Use reverse channel protocol
6	Set	(1)	If data set sense is off(1), do XOFF If data set sense is on(0), do XON
	Reset	(0)	If data set sense is off(1), do XON If data set sense is on(0), do XOFF
7	Set	(1)	Enable error detection
9-16	—		Number of user to which AMLC is connected.

Note

The system's default formula for determining user number is:

User-number = Line-number + 2

The user number is the number printed at the terminal upon LOGIN or LOGOUT, or printed by the STATUS USERS command. (STATUS prints this number as a decimal value, but its octal equivalent must be used in the AMLC command.) If the rightmost eight bits (9-16) of lword are zero, the AMLC line is not associated with any user space and is available to be assigned if the CONFIG directive NAMLC is greater than 0.

For example, to set line 3 as an assignable line using the transparant protocol and having a baud rate to 9600, the command line would be:

```
AMLC TRAN 3 2413 0
```

To reset line 3 as one that can be used for logging in, with TTY protocol, a baud rate of 1200, and connected to user 5, the command line would be:

```
AMLC TTY 3 2313 5
```

Terminal characteristics such as FULL DUPLEX/HALF DUPLEX, XOFF, etc. may be set at the user terminal with the PRIMOS-level TERM command.

Reverse Channel Protocol: Bits 5 and 6 of the lword are used to support devices that require "reverse channel" protocol to signal busy/ready.

For the devices that can use the "reverse channel" protocol, the carrier detect line may be used for the signal. Bit 5 of the lword indicates that the carrier detect line should be interrogated for data sense before performing an output operation. If a busy is detected, an action equivalent to an XOFF will be done for that line. When the carrier signal goes to the ready state, it will be flagged as an equivalent XON and output will resume.

A device may signal busy as data-set-sense on or off. Bit 6 is interrogated only if 5 is set; 6 can be set to be interrogated either way. For example, if the device signals busy as data sense off(1), the lword bit setting would be:

Bit 5 = 1 (Use reverse channel protocol)

Bit 6 = 1 (If data set sense off, do XOFF; if on, do XON)

Error Detection: Setting bit 7 of the lword enables a limited set of error detection. When this bit is set, the system checks for overflow of the user input buffer and for parity errors. If it finds an error in an incoming character, it replaces that character with an '225 (ASCII NAK).

Assigned AMLC Lines

The AMLC command may be used to configure assigned AMLC lines as well as terminal AMLC lines. After the system is running, users may assign the AMLC lines through the following command:

```
ASSIGN AMLC [protocol] line [configuration]
```

After the system is running, users may unassign the AMLC lines through the following command:

```
UNASSIGN AMLC line
```

When assigning or unassigning AMLC lines, the parameters used in the command line are the same as those used with the AMLC command.

System I/O Clock Rate

The system I/O clock rate is that of the top line number of the actual configuration (each AMLC board has 16 lines). This is independent of the number of users for whom the system is configured. See above.

Determining Line Numbers

There may be from one to eight AMLC boards. Each board has up to four ports (C,D,E,F). Each port has four cable connectors (J1,J2,J3,J4). Line numbers may be calculated with the formulas given in Table 8-1.

Table 8-1
Determining Line and User Numbers

AMLC Board	Value of x =	Address
1	0	'54
2	1	'53
3	2	'52
4	3	'35
5	4	'15
6	5	'16
7	6	'17
8	7	'32

AMLC Port	Value of y =
C	0
D	1
E	2
F	3

Cable Connector	Value of z =
J1	1
J2	2
J3	3
J4	4

To determine:	Use the formula:
Physical line number of AMLC	16(x)+4(y)+(z-1)
User number	16(x)+4(y)+(z+1)
AMLC command line (line)	Convert physcial line number of AMLC to octal

Note

Decimal-to-octal conversions are easily done at PRIMOS command level as:

```
TYPE [TO_OCTAL decimal-string]
```

Protocols for Older Model AMLC Boards

Lines attached to older model AMLC boards (model 5054, also known as the DMT AMLC board) may use the following values, referred to as "high-speed protocols", for the protocol argument: TTYHS, TRANHS, TTYHUP.

With a line using the high-speed protocols, a drastic increase in system overhead can result depending upon the Baud rate and the number of lines in the group. The user must be careful not to assign protocols to lines that normally have their character-time-interrupt flag always set (for example, the last line on the last AMLC). The basis for selection of high-speed protocols is discussed below.

TTYHS and TRANHS: TTYHS and TRANHS, the high-speed protocols, are used by lines connected to peripheral devices that can run at greater than standard terminal speed (9600 baud). These protocols are the same as those described above with one exception: for output only, the line's character time interrupt flag is set when the output buffer contains more than 40 characters, and it remains set until the output buffer contains fewer than 40 characters. The protocols have a burst mode effect on the output device.

TTYHUP: TTYHUP, the high-speed translating protocol, is used to avoid sending lowercase output to terminals or peripheral devices that cannot print lowercase characters.

> **Caution**
>
> Due to the increase in system overhead resulting from their use, the high speed protocols (TTYHS, TRANHS, and TTYHUP) should never be used with lines attached to the 5154 AMLC board (also known as the QAMLC board, or the DMQ AMLC board).

9

Allocating Hardware Resources

MAGNETIC TAPE DRIVES

You may choose whether users have the freedom to conduct tape operations themselves, or whether the operator is the only person who is allowed to handle tapes and tape drives. (You may also decide where the tapes will be stored.) Prime's software supports your decisions with the ASSIGN and SETMOD commands, as follows:

- The default state for the system allows users to do their own tape operations. (This state can be invoked explicitly by the command SETMOD -USER.)

  When the system is in this state, the ASSIGN command gives users the option of gaining exclusive control of tapes themselves, or of requesting the operator to assign a drive, set its characteristics for them, load and unload tapes, etc. The second choice allows phantom jobs and batch jobs to be run under operator control, while interactive jobs can run under operator or user control. Either the user or the operator can UNASSIGN a tape drive that a user has assigned.

- If you do not want users in the computer room, handling tapes and tape drives, you can change the system's state with the command SETMOD -OPERATOR. When the system is in this mode, the ASSIGN command channels all requests for tape drives to the supervisor terminal. The operator must approve or disapprove each request. Either user or operator can UNASSIGN a tape drive once it has been assigned to a user.

If you want your system to function in this mode as a matter of course, you may want to add the command SETMOD -OPERATOR to your C_PRMO file, so that it will be invoked whenever the system is started up.

- There may be times when the operator is not available to handle tapes, or when you want no tape operations conducted. The command SETMOD -NOASSIGN enforces this state. When the system is in -NOASSIGN mode, any attempt to ASSIGN a tape drive produces a message stating that tape drives cannot be assigned at the present time.

  When you wish to make the tape drives available again, either you or the operator may give another SETMOD command to place the system in -USER or in -OPERATOR mode.

LINE PRINTERS

You must make the following decisions regarding your line printers:

- What types of paper will you need? The most frequently used type should be your default type, so users won't have to ask for it by name. Users, operators, and spooler phantoms will all need to know what names to use to refer to other types.

  You define the names to be used by doing the following things:

  1. Creating a file named L.FORM. You create this file with the Editor (ED) and place it in the SPOOLQ directory. Put all the form names you create, one per line, in upper case, in this file.

  2. Including the appropriate form name(s) in the phantom environment files that control the printers. These files connect the form names with aspects of printing on that type of paper: size of page, etc. (See the System Operator's Guide for details.)

  3. Informing users of names, so they can use them in the SPOOL command's -FORM option. (The SPOOL command is explained in the Prime User's Guide and the PRIMOS Commands Reference Guide.)

- If you are using several types of paper, will you schedule specific times for each form use? (For example, WIDE forms are printed at 11 A.M. each day.) Or, will your operators monitor the spool queues and change to a special form whenever they see a number of requests for one form waiting in the spool queue? (Under either arragement, users can submit jobs whenever they like. The jobs will wait in the queue until a spooler phantom recognizes and accepts the -FORM name given. Users can also

tell when their jobs have been spooled by using the SPOOL -LIST command.)

If you do decide to mount special forms on schedule, you should inform both users and operators of the schedule. This will allow users to submit jobs close to the scheduled print time, and thus will reduce the need for space in the spool queue.

- What quota (if any) will you place on the SPOOLQ directory? This directory not only stores the files that control the printers, but also holds temporary copies of every file that is in the spool queue, waiting to be printed. (If a user has requested three copies of a file, three copies are stored inside SPOOLQ). If SPOOLQ runs out of records, jobs cannot be copied into it, and users who try to spool files receive error messages. Therefore, even if you do not want to place a quota on SPOOLQ itself, you may want to place quotas on other UFDs that share the partition with SPOOLQ, to make sure that SPOOLQ gets a certain minimum number of records.

- If you have several printers, do you want to define either form names or destination names by which certain printers are known, so that users can request output from a particular printer? For example, if you have several line printers and one letter quality printer reading the same spool queue, you might want users to request letter quality jobs with "SPOOL job -AT lqp". On the other hand, if the LQP itself sometimes uses cloth ribbon (for in-house work, perhaps) and sometimes carbon ribbon (for work going out to customers), then users might request the LQP with either "-FORM CLOTH" or "-FORM CARBON". (Note that a -FORM argument does not have to specify a type of paper. It can specify anything you choose.)

  You specify form names by creating a file L.FORM, within SPOOLQ, as explained above. You specify destination names by creating a similar file named L.DEST. For details on both, see the System Operator's Guide.

- Do you want users to come to get their own printout, or do you want the operators to deliver printouts to some specified place or places outside the computer room? If printouts are to be delivered to more than one place, you may create destination names that users can specify with the -AT option. (Destination names are created and handled in much the same way as form names. See the System Operator's Guide for details.)

- If you have several computers networked, will they read each other's queues or not? For each printer, this is controlled by the UPPER and LOWER parameters in the printer's phantom environment. (See the System Operator's Guide for details.) It also requires that each system allow its disks to be read by the others, as explained in the Chapters 17 and 18, on Primenet.

- If your site has printers that use Electronic Vertical Format Control (EVFU), rather than paper tape, to regulate printing, will you use default forms, or will you (or someone else) define forms of your own?

  If you use default forms, you must make sure the default forms are properly invoked in the phantom environment (by the parameter EVFU -ON).

  If special forms are needed, you must create (or have someone else create) specific EVFU format files. These are then invoked by the parameter "EVFU filename" in the phantom environment. (For information on how to create EVFU format files, see Chapter 16, ADDING AND MODIFYING SYSTEM SOFTWARE.)

  As usual, you must also make sure that users and operators know what form name refers to this specific environment.

10
Setting Up The Batch Subsystem

INTRODUCTION

Batch is the most flexible of PRIMOS's command file utilities. It makes phantom execution of jobs easier for the user, while giving the operator and System Administrator greater control of the environment and execution of the jobs. It does this by allowing the System Administrator to define from one to sixteen Batch queues from which user jobs can run as phantoms. These phantoms can be set to run "in the background" of the system: that is, to run concurrently with interactive jobs, but at somewhat lower priorities. In this way, they use only small amounts of CPU time when interactive use is heavy, but use large amounts of CPU time when interactive use is light or absent. Furthermore, Batch jobs may be held in their queues by operators, then released to run at appropriate times. For example, extremely long jobs, such as file updates and backups, can be set up as Batch jobs during the day, then run under operator control at night.

ADMINISTERING BATCH

The System Administrator's responsibilities with a Batch subsystem are as follows:

- Deciding how many queues to define, and what the characteristics of each queue should be.

- Seeing that the Batch subsystem is properly initialized once Rev. 19.0 software is installed.
- Making sure the proper command to bring up the Batch subsystem at startup time is included in the C_PRMO file.
- Seeing that the Batch queues which have been decided upon are created and added to the subsystem in the proper order. (The operator may do this.)
- Monitoring queue activity to decide how often old jobs should be removed from queues. The administrator must make sure that the FIXBAT utility is set up to perform this cleanup.
- Making sure that the queues themselves are renewed or replaced when necessary.
- Seeing that either the FIXBAT or the INIT utility is run to repair or replace the Batch data base, if it becomes damaged.

This chapter provides guidelines and information for each of these tasks. It is supplemented by a full treatment of the daily management of the Batch subsystem given in the System Operator's Guide.

REQUIREMENTS FOR A BATCH SUBSYSTEM

PRIMOS

Rev. 19.0 Batch requires Rev. 19.0 PRIMOS. It will not run on earlier versions of PRIMOS. (Similarly, pre-Rev. 19 versions of Batch will not run on a Rev. 19 system.)

Phantoms

A Batch subsystem requires one phantom exclusively to control the Batch monitor. (This phantom runs under the name BATCH_SERVICE.) The monitor runs Batch jobs on whatever other phantoms are available.

File Units

A Batch subsystem also requires a minimum of 16 file units for each user. This means that the FILUNT directive in the CONFIG file must be set at 16 or higher. (See Chapter 3 for details on the FILUNT directive.) If FILUNT is set too low, error messages requesting that it be altered are printed by Batch on the supervisor terminal.

HOW THE BATCH SUBSYSTEM WORKS

Batch Queues

Each Batch queue is a separate entity, defined by the System Administrator to be particularly hospitable to certain types of jobs. Queues are defined by the following nine parameters:

- Name
- Default CPU time limit
- Maximum CPU time limit
- Default elapsed time limit
- Maximum elapsed time limit
- Default PRIMOS file unit for command input
- Default value for priority of job within queue
- Relative run-time priority
- Timeslice

The System Administrator creates queues and defines their characteristics using the BATGEN command (explained later in this chapter). Strategy for defining queues is explained in the section of this chapter on PLANNING FOR A BATCH SUBSYSTEM.

Queues and Users

Users submitting jobs may specify the following parameters: queue, CPU limit, elapsed time limit, file unit (for COMINPUT files only), and job priority within queue. If users do not specify these parameters, the Batch monitor places their jobs in the first available queue and assigns the jobs the queue's default values. The System Administrator must either make that first available queue a reasonable default queue or ensure that users know which queues they should use and what the time limits and default values of those queues are.

By using the BATGEN -STATUS and BATGEN -DISPLAY commands, users can see what queues are available and what their characteristics are. They can then submit their jobs to the appropriate queues.

How Batch Allocates Phantoms

A Batch subsystem can consist of a single queue with no limits (except for user-defined ones) placed on jobs running within it. The system

then simply runs jobs sequentially. Under this system, all jobs will have the same runtime priority. Users may still request priorities within queues (from 9 down to 0) for their jobs.

Alternatively, the system can contain from two to sixteen queues. In this case, the Batch monitor checks each queue in turn, beginning with queue number one. If it 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 begins checking each queue again, to see if any jobs have finished or aborted. If a job has finished, the monitor does the necessary finishing-off (marking the job as completed or aborted, deleting its temporary files, etc.) and then checks the queue for another waiting job.

A slightly different situation arises if there are fewer available phantoms than queues. For example, if there are three queues but only one phantom available to run jobs, the monitor will run all jobs that are ready to run (that is, all jobs not marked "held") from queue one before running a job from queue two; and it won't run jobs from queue three until queues one and two are both empty, or until they contain only "held" jobs.

Control of the Batch Subsystem

Once a Batch subsystem is established, the operator and Administrator can control it by:

- Pausing the monitor, temporarily halting the subsystem. (Currently executing jobs will finish before the subsystem halts.)
- Blocking individual queues, keeping those queues from executing new jobs while letting the rest of the subsystem continue running.
- Adding new queues.
- Deleting queues. They can do this gracefully, allowing all jobs in the queue to finish before the queue vanishes; or, they can do it abruptly, destroying all jobs in the queue as they do so. (Obviously, the latter is an emergency measure.)
- Aborting, canceling, or holding individual jobs. (A held job remains in the queue but cannot be executed until the operator or administrator releases it.

Details on how to do these things are given in the System Operator's Guide.

PLANNING FOR A BATCH SUBSYSTEM

The basic decisions in planning a Batch subsystem are:

- How many queues do you want to define?
- What characteristics do you want each queue to have?

Some guidelines for making these decisions follow.

How Many Phantoms

The number of Batch jobs that can run simultaneously is limited by two factors: the number of queues and the number of available phantoms. Only one job per queue can execute at one time; therefore, the number of queues provides a theoretical upper limit. On the other hand, no job can run without a phantom to run it. Therefore, the number of phantoms provides an actual limit, which may be lower than the theoretical one.

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

You need not limit the numbers of queues to the number of phantoms. In fact, it is often wise to have more queues than phantoms. Having extra queues allows the queues to segregate jobs by priority. For example, imagine a subsystem containing three queues. The first queue could have stringent CPU time limits on its jobs. This queue would be available only to very short jobs, and it would give these jobs top priority. The second queue could have moderate (or no) CPU time limits, and a moderate elapsed time limit. It would accept "average" jobs, and give them medium priority. The third queue could have no limits. Large, slow jobs would go into this queue. They would not run unless the other queues either were empty or had phantoms taking care of them.

Borrowing Phantoms: There are two ways that either the operator or the Administrator can keep the Batch subsystem from using phantoms: blocking individual queues (which frees only the phantoms that would be used by those queues) or pausing the Batch Monitor (which frees all the phantoms that service the queues). The operator or Administrator can use this ability in the following ways:

- If a special job needs to be run as a phantom, and the Batch subsystem is using all the phantoms, queues can be blocked or the subsystem paused. When a job finishes, the phantom that was running it can be used to run the special job. (You would use the PHANTOM command to run the special job.)

- If there are fewer phantoms than queues, and one job has either been waiting for an unduly long time or simply must be run, the operator or Administrator can block the higher queues. When the current jobs in those queues finish, jobs from the neglected queue will begin to run. If this queue contains jobs ahead of the job for which you're blocking queues, the operator or Adminsitrator can HOLD those jobs.

Timeslices and Scheduler Priorities

The fact that timeslices and scheduler priorities are set individually for each queue allows queues to be tailored for quick, average, or slow jobs. The general guidelines are these:

- If a queue is intended for short jobs, it should offer limited CPU time but run at a relatively high priority. Its timeslice can be short or normal.

> **Note**
>
> You can force users to set CPTIME limits on their own jobs by setting a queue's default CPU time higher than its maximum CPU time. A job which is not submitted with the -CPTIME option cannot use this queue.

- If a queue is intended for average jobs, it should have default timeslice and priority.

- If a queue is intended for large, slow jobs, it should have no CPU time limit and no elapsed time limit. It should have a relatively low priority, but a large timeslice.

If queues for short jobs are given a fairly high scheduler priority, they can operate even when interactive use of the system is fairly heavy. The low priority, high timeslice queues will not be able to run jobs when interactive use is heavy; but they will be able to take advantage of time when interactive use is light.

Search Order

When the Batch monitor is searching its list of queues, either to find a queue in which to put a job, or to find which queues have jobs that need running, it searches the queues in the order in which they were added to the system. (Queues are added by the BATGEN utility's ADD command.)

You will want to take advantage of this fact in the following ways:

- Queues for extra short jobs (if they exist) should come first in the search order but should not accept jobs without the -CPTIME option.

- Your "default queue" — the one that catches jobs submitted without options — should be the first queue a job can fall into. This means either that it must be the first queue in the search order or that the queues that precede it must require some option such as the -CPTIME option to be supplied by the user.

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

Changing Queues

The Batch subsystem numbers jobs as it puts them into the queues. Removing old jobs from queues does not change the numbering scheme; the Batch monitor continues to allocate numbers as though all previously-defined jobs were still present.

This means that you will have to delete and re-create queues as they fill up. It also gives you a means to check on queue usage. If one queue is very heavily used, you may want to define another with the same characteristics to share the load. If another queue is rarely used, you may want to remove it and replace it with a queue having different characteristics.

INSTALLING THE BATCH SUBSYSTEM

The Batch system directly involves three top-level UFDs on each system. These are CMDNC0, BATCH, and BATCHQ. BATCH uses CMDNC0 only to hold the commands $$, BATCH, BATGEN, and JOB themselves. BATCH contains the source code for the Batch subsystem, including the command files necessary to build and install itself. BATCHQ contains the command files used to initialize the Batch data base (which will also reside in BATCHQ).

Note

The contents of these directories are different at Rev. 19 than they were at Rev. 18. For details, see the Rev. 19.0 Software Release Document.

The build file for Batch, which is run when Rev. 19.0 is installed, installs the proper software for Batch in the UFDs BATCHQ and CMDNCO. The one necessary file it does not install is the file START_BATCH_MONITOR.COMI. This file is present in the BATCHQ UFD on the U1 partition of the Master Disk.

Suppressing Execution Messages

The system administrator can prevent the Batch monitor from sending messages to the supervisor terminal each time it begins or finishes a job by changing the file BATCHQ>START_BATCH_MONITOR.COMI. To do this, edit the file and change the line "RESUME MONITOR" to read "RESUME MONITOR -HUSH". The change takes effect the next time BATCH -START is invoked.

Initializing the Data Base

When the files and directories necessary for the Batch subsystem have been installed, the System Administrator should:

1. Make sure system time and date are set.

2. Initialize the data base by running the INIT program. The command format that does this is:

   R BATCHQ>INIT -RSTQ

   More details on INIT are given below.

3. Run BATGEN to create the queues for the new BATCH subsytem. (For details, see the section called DEFINING AND MODIFYING THE BATCH ENVIRONMENT, later in this chapter.)

Batch and the C_PRMO File

The Batch subsystem is usually started at system startup. The command for starting it in the C_PRMO file should be:

BATCH -START -RLV rlv -TS ts

In this command, rlv and ts are decimal integers.

Note

At Rev. 18, the corresponding commands were:

```
PH BATCHQ>PH_GO
CHAP -nn rlv ts
```

If you are converting your Batch subsystem from Rev. 18 to Rev. 19, please make sure you change the C_PRMO file accordingly.

INVOKING INIT

To invoke INIT, type:

```
R BATCHQ>INIT [-RESET_QUEUES] [-ADMINISTRATOR user] . . .
```

The -RESET_QUEUES option may be abbreviated to -RSTQ. The -ADMINISTRATOR option may be abbreviated to -ADMIN.

The -ADMINISTRATOR Option

The -ADMINISTRATOR option specifies the Batch administrator(s) for this system. user should be the name of a user who can log into the system. If the -ADMINISTRATOR option is not present, the user who invokes the INIT program is assumed to be the administrator.

More than one administrator may be specified by repeating the -ADMINISTRATOR option. In fact, System Administrators should always specify themselves as well as whatever users are to serve as Batch administrators. For example:

```
R BATCHQ>INIT -ADMIN me -ADMIN joe -ADMIN mary
```

The reason for this is that the -ADMINISTRATOR option is not cumulative. Using the option removes old administrators as well as adding new ones.

The Batch administrator is given ALL access to the BATCHQ UFD and to all its sub-UFDs and files. (Users SYSTEM and BATCH_SERVICE are also considered to be Batch administrators, and hence, also have ALL access.)

The RESET_QUEUES Option

If -RESET_QUEUES is specified, a new BATDEF file, with no defined queues, will be created in BATCHQ. If it is not specified, the old BATDEF file will be left as is. Since the Rev 19.0 BATDEF and previous versions of BATDEF are incompatible, this option must be used when the Rev 19.0 Batch subsystem is first initialized.

Note

> The INIT program does not actively prevent any user from running it. But since only Batch administrators will have access to the program, and since only they will have sufficient access to BATCHQ to initialize anything, there should be no problems with misuse of INIT.

ACCESS ISSUES

Creating a Batch Administrator

The new INIT program allows the specification of one or more Batch administrators. It uses this information to set up the access on the Batch data base. A Batch administrator is defined as any user with ALL access to BATCHQ. Therefore, users SYSTEM and BATCH_SERVICE are automatically Batch administrators.

Batch administrators can run BATCHQ>FIXBAT and BATCHQ>INIT. They can also use the Batch commands -STOP, -PAUSE, and -CONTINUE. Moreover, Batch administrators have full access to BATDEF, so they can change queue configurations at any time. All other users are given Read access only. Read access is necessary for all users to submit Batch jobs, as the JOB program must have access to the BATDEF file.

Monitor and Operators as Privileged Users

At Rev. 19, the Batch monitor runs under the name of BATCH_SERVICE. (At Rev. 18, it ran as SYSTEM.) The Batch subsystem recognizes both BATCH_SERVICE and SYSTEM as privileged users. These users can display and modify all users' jobs in the Batch subsystem. In fact, they are the only users who can manipulate other users' jobs; the other Batch administrators cannot do this. Also, only SYSTEM and BATCH_SERVCE can use the -HOLD and -RELEASE options on the JOB command.

Finally, only user SYSTEM, at the supervisor terminal, can give the BATCH -START command or abort other users' running jobs.

At Rev. 19.0, it does not matter whether or not the name BATCH_SERVICE is listed as a username in the User Validation File (UVF).

ACL vs. Password MFD

The Rev. 19 Batch subsystem is designed to take advantage of ACL protection. It grants particular rights to privileged users — BATCH_SERVICE (the Batch monitor), SYSTEM, and Batch administrators — and grants varying, lesser rights to other users. This allows them to use the system, but not to access each other's jobs or to access Batch files and directories.

If the MFD in which BATCHQ resides is a password directory, BATCHQ itself must be a password directory. However, neither BATCHQ nor its subdirectory Q.CTRL may have passwords. If these directories are given passwords, the Batch subsystem becomes inoperable. Therefore, no security exists on a system on which Batch is running and in which BATCHQ is a password directory.

How INIT Responds to a Password MFD

The INIT program tries to set access rights on BATCHQ. If it finds that it cannot do so because BATCHQ resides in a password MFD, INIT sends the following warning message:

Warning: MFD containing BATCHQ is a password MFD (INIT)

The Batch subsystem then ignores all further ACL-related error messages — such as "Parent not an ACL directory" (E$PNAC) and "Not an ACL directory" (E$NACL) — generated by its attempts to set protection or otherwise deal with ACLs.

CIFILE and OTHER are Password Directories

The directory BATCHQ>CIFILE must be a password directory. The directory BATCHQ>OTHER should be a password directory. (If it is an ACL directory, all Batch users must have ALL access to it. Since this is likely to result in an ACL of "$REST:ALL" on the directory OTHER, it is probably more secure as a password directory.)

Changing the Password: INIT creates BATCHQ>CIFILE and BATCHQ>OTHER as password directories using the current Batch password as the owner password and six blanks as the non-owner password.

As released on the Master Disk, the Batch password is OSIRAS. If you want to change the password, you can do so as follows:

```
OK, ATTACH BATCH      /* Attach to the source ufd.
OK, ED B$LIBF.FTN     /* Edit the Fortran library for Batch.
EDIT
LOCATE --- BATCH PASSWORD ---     /* Find the password.
      CALL MOVE('OSIRAS', SUBPAS(1,2),3)  /* --- BATCH PASSWORD ---.
MODIFY /OSIRAS/newpas/   /*Change the password.
      CALL MOVE('newpas',SUBPAS(1,2),3)  /* ---BATCH PASSWORD ---.
FILE

OK, R BATCH.BUILD         /* Rebuild Batch.
```

After changing the password, it is wise to set the access on the Batch UFD so that no unauthorized users can read its files.

The password ("newpas" in the example) must be six characters or less in length.

User Access for Jobs

Batch jobs submitted by users take on the profiles of the submitting users. Thus, the Batch jobs are assigned the same group and project names that the users had at the time they invoked the jobs.

DEFINING AND MODIFYING THE BATCH ENVIRONMENT

Once Batch has been installed, you must define its environment. You do this by defining from one to sixteen queues via the BATGEN command. (System date and time must have been set before you use BATGEN.) This command has the form:

```
BATGEN pathname
```

Usually, pathname will be BATCHQ>BATDEF, as the BATDEF file is the only file the Batch monitor reads in its search for queues in which to place jobs. It is possible, however, to create queues in other files and then transfer them into the BATDEF file. You do this by:

1. Typing "BATGEN pathname", where pathname is something other than BATCHQ>BATDEF.

2. Doing whatever work you want within BATGEN.

3. Exiting BATGEN with the command FILE BATCHQ>BATDEF.

For an example of this, see the section on CLEANING UP QUEUES, later in this chapter.

> **Caution**
>
> Only BATGEN can copy new queue configurations correctly into BATCHQ>BATDEF. If you try to copy in new configurations with COPY or with FUTIL, you will disturb the ACLs set on BATDEF by INIT. If this happens, users will be unable to use the Batch subsystem. (They will probably get the error message, "Insufficient access rights. BATDEF missing.") In order to remedy this situation, you must use BATGEN to recopy the desired queues into BATCHQ>BATDEF.

Once pathname has been read and validated, BATGEN types a prompt character and waits for a BATGEN command. Available commands are:

ADD queuename

MODIFY queuename

DELETE { queuename | ALL }

BLOCK { queuename | ALL }

UNBLOCK { queuename | ALL }

DISPLAY { queuename | ALL }

STATUS

FILE pathname

QUIT

A queuename is an alphanumeric name of up to 32 characters. It is created by the ADD command and is the only name by which the queue may be referenced. Queuenames must conform to standard PRIMOS filename rules. The name ALL is illegal, as it would cause ambiguity in commands such as BLOCK ALL or DELETE ALL, where ALL means "all queues".

Note

The name of the queue has nothing to do with the queue's number or with the order in which queues are searched for jobs. The id number (which becomes the first digit after the "#" of the job number for jobs executing from that queue) is assigned by the Batch system and reflects the order in which queues are used when they are first defined. The search order reflects the order in which the queues are created, or added, to the Batch subsystem. To establish a queue as the number-one queue for searching, ADD it first; ADD the number-two queue second, and so on.

The BATGEN commands, and their subcommands, are defined on the following pages.

Command	Function
ADD queuename	Instructs BATGEN to create a new queue. If queuename is acceptable, ADD returns the message, "Enter queue characteristics:", prints a prompt ($), and waits for a subcommand (described below). If queuename is already in use, it returns a fatal error message, "Queue queuename already exists." ADD subcommands are discussed immediately following this list of commands.
MODIFY queuename	Instructs BATGEN to modify an existing queue. If queue queuename exists, MODIFY responds, "Enter queue characteristics:", prints a prompt ($), and waits for subcommands (described below). If queue queuename does not exist, or if it is flagged for deletion, MODIFY sends a fatal error message. MODIFY subcommands are discussed immediately following this list.
DELETE {queuename / ALL}	Flags an existing queue (or all queues) for deletion. Queue(s) will accept no more jobs and will be deleted when all currently waiting jobs have been run.
BLOCK {queuename / ALL}	Sets flag in status control block of an existing queue (or of all queues) to disallow submission of further jobs to the queue.

UNBLOCK {queuename / ALL}	Resets flag to allow submission of jobs to a previously blocked queue (or to all queues). Default status for queues is "unblocked".
DISPLAY {queuename / ALL}	Displays name, status, and characteristics of the named queue (or of all queues). Omitting the optional argument displays information for all queues.
STATUS	Shows name and status of all queues in tabular form.
FILE pathname	Modifies file named pathname to include commands given during this session. If pathname is not given, current file is modified (the usual situation).
QUIT	Terminates session without changing file. If anything was modified during the session, BATGEN will ask, "Environment modified, OK to quit?" A "yes" answer (or a carriage return) is then needed to execute QUIT. (BATGEN may be restarted with the PRIMOS START command after a QUIT, with no loss of information.)

ADD and MODIFY Subcommands

Subcommands for the ADD and MODIFY commands are identical. Six of them — CPTIME, ETIME, FUNIT, PRIORITY, RLEVEL, and TIMESLICE — define queue characteristics. Two others — RETURN and QUIT — tell BATGEN to save or ignore the preceding subcommands. The ADD/MODIFY subcommands function as follows. (All numeric values must be decimal integers.)

▶ CPTIME default maximum

Sets CPU time limits for jobs run in this queue. The default limit will be placed on any job whose user does not specify a CPTIME limit. The maximum is an absolute limit: jobs asking for greater CPTIME than the maximum will not be allowed into the queue.

The values for CPTIME are given in decimal seconds. The word NONE may also be used, to signify that no time limit is to be set. Thus, the subcommand CPTIME 30 NONE would cause jobs submitted without CPU limits to be limited to 30 seconds of CPU time but would allow unlimited time to those requesting it.

The default value may exceed the maximum. For example, CPTIME NONE 60 is a legal command. Its effect is to close the queue to jobs that do not specify CPTIME limits of 60 seconds or less, since these jobs would

be given the queue's default limit of NONE and then denied admission to the queue because their CPTIME limit was greater than the queue's maximum. If you wish to demand that users define their own time limits, this is the way to do it.

As delivered, the system has default values of "NONE" for default and maximum CPTIME. Unless both CPTIME limits are explicitly given, they will be set to "NONE" when the queue is created.

(When modifying existing queues, one or both limits may be changed. In this case, the command "CPTIME default maximum" would change both values, while the command "CPTIME default" would change only the default value.)

▶ ETIME default maximum

Sets elapsed time limits. It acts exactly as CPTIME does, except that its values are given in minutes rather than seconds. Its system defaults are both NONE.

▶ FUNIT number

Sets a default file unit for command input for any non-CPL job in the queue which has not specified its own file unit number. Numbers range from 1 to 126. The maximum is dependent on the number of file units set by the System Administrator. System default is 6.

▶ PRIORITY value

Sets the default value for a job's priority within the queue itself — that is, its priority vis-a-vis other jobs in the same queue. Any job not specifying its own priority will be given this default value. Permissible values are from 0 to 9 with 9 being the highest priority and 0 the lowest. System default is 5. (Note that this priority affects only the order in which jobs within a single queue are initiated. It does not determine how fast they run. Use RLEVEL and TIMESLICE to determine run-time priority.)

▶ RLEVEL delta-value

This subcommand does not set the runtime priority for jobs in the queue. Rather, it determines the amount their priority will be lowered from the priority of the Batch monitor. (The monitor's priority is set with the -RLEVEL option of the BATCH -START command.) Delta-values may range from 0 to 7, with 0 meaning that the queue's jobs will run on the same priority as the monitor does, and 7 representing the maximum

lowering. (Note that this is one value users may not specify for themselves.) System default is 0.

PRIMOS currently allows a process to have a priority from 0 to 3. Therefore, if the Batch monitor is running at priority 3, RLEVEL values from 3 to 7 are identical. If the monitor is running at priority 1, RLEVEL values from 1 to 7 are identical.

▶ TIMESLICE value

Sets the timeslice value for jobs in the queue. A queue's timeslice may be smaller than the monitor's timeslice and be effective; but if it is larger, it will be ignored, and the monitor's timeslice will be used for each job in the queue. (Again, the user has no control over this value.) Timeslice values represent tenths of a second. They may range from 1 to 99, but probably should not go below 20 unless job priority is unusually high. System default is 20, equaling 2 seconds.

The next two commands terminate subcommand sessions and return you to BATGEN command level.

▶ RETURN

Saves the new characteristics for future display and/or filing.

▶ QUIT

Throws away the work done at subcommand level. If you were modifying an old queue, QUIT leaves that queue unchanged. If you were adding a new one, QUIT throws away the new queue's name as well as its characteristics. If you modified anything before quitting, BATGEN asks, "Queue definition modified, ok to quit?" If it does not receive an answer of "yes" (or a carriage return), it prompts you to save work with "Please return."

USING FIXBAT

FIXBAT (FIX BATch) is an off-line utility designed to:

- Handle the start-up protocol for the Batch monitor, making sure that the data base is valid before starting the monitor.
- Fix any broken pointers within the queue files.
- Reclaim disk space by deleting from the Batch queues all inactive jobs of a given age, or older.

FIXBAT is run automatically every time the Batch monitor is started up by the BATCH -START command. The System Administrator can choose whether FIXBAT merely checks for a valid data base during this procedure (cleaning up the data base, if necessary), or whether it also reclaims disk space by removing old files from the queues.

FIXBAT may also be run interactively. (If the Batch data base becomes invalid, for instance, you would run FIXBAT interactively to repair it.)

Running FIXBAT at Startup Time

FIXBAT is run automatically by the Batch monitor whenever it is started up by the BATCH -START command. The command that runs FIXBAT is found in the command file BATCHQ>START_BATCH_MONITOR.COMI. As released, the command is:

```
RESUME FIXBAT -STARTUP SAVE
```

This command checks to see that the data base is valid before beginning the monitor, but it does not clean old jobs out of the data base. Since most administrators want this cleanup done on a frequent basis to conserve disk space, you probably want to add the -DAYS option to the command line. -DAYS takes a numeric argument. You probably want to use either 0, 1, or 2. The argument 2 cleans out jobs that have been run two days ago or more. The argument 1 cleans out jobs that have been run one day ago or more. The argument 0 cleans out all finished jobs. (For further details on the -DAYS option, see the discussion below.)

To add the -DAYS option, edit the RESUME FIXBAT command in the file START_BATCH_MONITOR.COMI.

When BATCH -START Runs FIXBAT

Here is a brief explanation of what happens when the BATCH -START command is run. Details are provided in the fuller explanations that follow.

1. The BATCH -START command starts up a phantom (logged in as BATCH_SERVICE) that runs the command file BATCHQ>START_BATCH_MONITOR.COMI.

2. The command file runs FIXBAT.

3. FIXBAT deletes any temporary files from BATCHQ.

4. FIXBAT checks to be sure the Batch data base is valid.

5. If the data base is not valid, or if the -DAYS option is specified, FIXBAT attempts to fix the data base. If FIXBAT cannot repair the data base, it aborts with an error message.

6. If FIXBAT fixes the data base, it also checks to make sure that CIFILE is still a password directory. If it finds that CIFILE has been changed to an ACL directory, FIXBAT aborts with the error message:

   BATCHQ>CIFILE cannot be an ACL ufd. Do R BATCHQ>INIT. (FIXBAT)

7. When FIXBAT finishes its tasks, the command file executes the command RESUME MONITOR. This starts the Batch monitor and sends the message:

   Monitor started

   to the supervisor terminal.

Note

The length of time it takes for the "Monitor started" message to appear after the BATCH -START command is given depends on the amount of work FIXBAT has to do. Remember, too, that the monitor cannot begin to work until the system time and date have been set.

Invoking FIXBAT Interactively

FIXBAT resides as a program, FIXBAT.SAVE, in the BATCHQ UFD. To run FIXBAT:

1. Log out the Batch monitor (if it is running), using the command BATCH -STOP.

2. Log in as SYSTEM or as a Batch administrator.

3. Attach to the BATCHQ UFD.

4. Resume FIXBAT, with the desired options (explained below).

If you try to start FIXBAT while the Batch monitor is running, FIXBAT returns with the error message, "Batch monitor is running, do BATCH -STOP."

The FIXBAT Command and Its Options

The format for the FIXBAT command is:

RESUME FIXBAT [options]

There are three options:

Option	Meaning
-DAYS n	Removes all cancelled, completed, or aborted jobs which are n or more days old from the Batch queues. (n must be an integer between 0 and 60.) If n is 0, all non-active jobs are removed from the queues.
-QUIET	Does not send a message to the terminal when FIXBAT removes a job from the queue. (This option is useful only if the -DAYS option is also given.)
-STARTUP argument	Tells FIXBAT to start the BATCH monitor. When this argument is given, FIXBAT assumes that it is being run by the BATCH -START command. That is, it assumes it is being run as a phantom from the supervisor terminal. When the -STARTUP argument is used, the phantom that runs FIXBAT becomes the Batch monitor when FIXBAT is done. The -STARTUP option takes one of four arguments: SAVE, SPOOL, DELETE, or NOLOG. These arguments tell FIXBAT what to do with the Batch comoutput file. SAVE: Renames the current comoutput log "OLDLOG" (deleting any existing "OLDLOG"). Creates a new comoutput file named O_LOG. SPOOL: Spools the current comoutput file, calling it BATCH.LOG. Creates and opens a new O_LOG file. DELETE: Opens O_LOG as a comoutput file. (The file is truncated when it is opened, destroying the existing contents.) NOLOG: Takes no action with regard to comoutput files.

The O_LOG file, when generated by -STARTUP, contains an identifying first line, suitable as a header line for a spool file. It contains the time of day, the date (and the day), and the FIXBAT revision number. After that come two blank lines, and the log trail of what FIXBAT did.

This may include such comments as "Fixing database", "Deleted T$0001", "Deleted C00041", which are not errors, simply notifications that certain files deemed useless by FIXBAT were deleted.

It may also include the information on any jobs that were deleted from the queues (unless -QUIET was specified). In addition, any strange file formats (such as partial queue entries) are noted here.

One thing to note is that most of FIXBAT's output when -STARTUP is specified is preceded by the time of day on the line. This is useful for determining the timing of operations related to a user complaint that "something went wrong". For example, running FIXBAT interactively might result in the line:

```
Deleted T$0000
```

being typed out, whereas if FIXBAT had been running as part of BATCH -START, the line might have looked like:

```
06:43:52 Deleted T$0000
```

Note

The DELETE and NOLOG arguments to the -STARTUP option are not recommended, as they make it difficult for the System Administrator to analyze any Batch problems that may occur. However, when problems do occur, the Batch subsystem will try to create a file named "ERROR." in the BATCHQ UFD giving some information on the error.

Changing the argument to -STARTUP may result in old comoutput files (OLDLOG or O_LOG), being left around, or recent log files being deleted. It is recommended that whenever a change like this is made, the System Administrator make sure that any existing files are taken care of before the Batch monitor is started up with the new START_BATCH_MONITOR.COMI file.

If FIXBAT aborts, the cause can generally be found by looking at the log file. Usually, deleting the offending file and restarting the Batch monitor (and therefore FIXBAT) is the fastest way to fix any problems.

If FIXBAT has been run by the BATCH -START command, it has been running as the Batch monitor. In this case, when FIXBAT has finished, the BATCH -START command will RESUME MONITOR and the monitor rev number will be typed out, followed by a log trail of its activities.

Cleanup Operations

When FIXBAT is run interactively (without the -STARTUP option), it automatically fixes the data base. When FIXBAT is run with the -STARTUP option (as with BATCH -START), however, it fixes the data base only if one (or more) of three conditions is met:

- It fixes the data base if -DAYS n has been specified, in order to remove old jobs from the queue.
- It fixes the data base if it cannot find the file "BATCHQ>OTHER>VALID." (The absence of this file indicates an invalid database.)
- It fixes the data base if it cannot find the "MON.ST" file in the BATCHQ UFD. (The absence of this file indicates that the monitor was not logged out gracefully — i.e., that it aborted, was forcibly logged out, or was halted by a system shutdown or crash.)

How FIXBAT Works

The course of FIXBAT's activities is as follows:

1. FIXBAT begins by deleting temporary files out of BATCHQ. Temporary files are defined as files with six-character names beginning with "T$" and with the last four characters successfully converting to decimal numbers using CNVA$A. (See the Subroutine Reference Guide for a description of the APPLIB CNVA$A routine.) These files are generated by the Batch monitor to "bootstrap" Batch jobs, including attaching to the home UFD. If that attach fails, the temporary file stays around. Usually, the Batch monitor deletes these files itself. When a temporary file is deleted, the message "Deleted <name>" is output.

If FIXBAT decides to fix the data base, it prints the message "Fixing database", and takes two further actions:

2. It protects all command files in CIFILE to "D NIL" protection. A command file is defined here as a file with a six-character name beginning with "C", where the second character is between "0" and "9" or "A" and "V" (inclusive), and with the last four characters successfully converting to a decimal number using CNVA$A. Between this step and step 3, FIXBAT will go through the Q.CTRL queue files, and whenever it finds an active job, it will attach back to BATCHQ>CIFILE and protect the command file of the job to "R NIL" protection.

3. It deletes all command files in CIFILE, ignoring "Insufficient access rights" errors, and printing a "Deleted <name>" message if successful. This results in all command files not belonging to active jobs being deleted.

Whenever FIXBAT fixes the data base, it checks CIFILE to make sure that it is still a password directory. If it finds that CIFILE has been changed to an ACL directory, FIXBAT prints the error message:

BATCHQ>CIFILE cannot be an ACL ufd. Do R BATCHQ>INIT (FIXBAT)

Deleting the Old Batch Job Entries: When FIXBAT deletes old Batch job entries from the queue files, it physically removes the job entry from the queue and writes the next job entry over the deleted one, repeating this procedure (in effect filling up the hole made) until the end of the queue file is reached.

It will only perform this operation, however, if a -DAYS argument was specified on the command line.

The mechanism for determining whether or not a job should be deleted is as follows:

1. The job must not be an active job, i.e., it must be in a cancelled, aborted, or completed state.

2. Unless "-DAYS 0" was specified, the job must have completed, aborted, or been cancelled in the same or previous year as the current year.

3. Unless "-DAYS 0" was specified, the job must have completed on a date such that there are at least <n> full days between that date and the current date, non-inclusive. This means that if a job was completed on April 10, 1982, and the current date is April 12, 1982, the only way that job can be deleted is if <n> is 1. If <n> is 2, the job will not be deleted until the next day. (<n> is the argument supplied to the -DAYS option.)

When FIXBAT deletes a job, it outputs the final information on that job in a similar format to the information returned by a "JOB -DISPLAY" command, unless the -QUIET option was specified on the command line.

Note

If a deleted job is displayed, the queue name may be blank. This occurs if the user did not explicitly specify a queue. Also, the queue name may not resemble the queue name as defined in BATGEN with regard to uppercase/lowercase mapping. For example, for queue "COBOL", a deleted job may have "(queue COBOL)", "(queue cobol)", or "(queue)" output.

FIXBAT Error Messages and Responses

While FIXBAT is running, it may output certain messages describing what it is doing, or it may abort with a particular error message.

In general, if FIXBAT aborts, it means that certain parts of the data base are irretrievably lost. It is expected that this will usually be Batch job data. While deleting the offending file and rerunning FIXBAT may help, it does not guarantee that FIXBAT won't abort on a different file.

If FIXBAT does not seem to be able to fix the data base, the INIT program should be invoked.

CLEANING UP QUEUES

Each Batch queue numbers its jobs from 0000 to 9999. When number 9999 is reached, the queue is considered "full", whether it still contains jobs or not.

When full queues exist, the following things happen:

- When users submit jobs to the full queue (using the JOB command's -QUEUE option), they receive the error message, "queue full".

- When users submit jobs without specifying queues, the monitor conducts its usual search for queues. However, it ignores the full queue, treating it as if it were blocked. If the full queue was the only queue that met a user's requirements, that user receives the error message, "No queue available for job." (If some other queue is acceptable, the monitor simply submits the jobs to that queue.)

Therefore, when a queue becomes full, the System Administrator must first delete the queue, then define it, so that new jobs may be submitted to it.

There are three ways to remove jobs from queues:

- Run the INIT program as explained in INSTALLING THE BATCH SUBSYSTEM. This is the fastest way to clean out queues, as it will empty all queues. (If you run INIT with the -RSTQ option, it will also wipe out BATDEF.) Be warned, however, that this process destroys active jobs along with inactive ones. To prevent wiping out jobs that have not yet run, block all queues with the BATGEN "BLOCK ALL" command and let all waiting jobs finish before running INIT.

- Delete one queue at a time, using the BATGEN "DELETE queue-name" command. This method assures greatest continuity of Batch service since it leaves some queues available at all times and destroys no information on active jobs.

  The monitor ignores the queue when handling job submissions. Users attempting to submit jobs to the queue are given a fatal "Queue does not exist" message. Active jobs in the queue, however, are not disturbed, but run as they would in any other queue. Only when the last job has completed or aborted is the queue actually deleted. Then its data base is deleted, the queue is removed from the BATDEF file, and a message, "Queue queue-name deleted," is sent to the supervisor terminal.

  Note

  If a queue has never had a job submitted to it, the deletion message "Removed queue-name from BATDEF", indicates that no job information was wiped out during the deletion.

  Once the queue has been deleted, a new queue -- either different from or identical to the old one -- may be placed in the BATDEF file. (If BATDEF originally contained less than sixteen queues, the new queue could be added before the old one was deleted or flagged for deletion; but the name of the new queue could not be identical to the name of the old one. To conserve name, DELETE, wait for the monitor to remove the queue from BATDEF, then ADD.)

- Forcibly remove one or more queues from BATDEF by the following method:

  1. Create an empty file with the command:

     BATGEN new-pathname

  2. ADD queues identical to those you wish to retain in the BATDEF file. (Names and all parameters must be identical.) ADD also any new queues you wish to create.

  3. FILE your new BATDEF file with the BATGEN command:

     FILE BATCHQ>BATDEF

     The existing BATDEF file will then be replaced by the new one, and the new configuration will take effect immediately.

> **Caution**
>
> This method will abort all jobs running in the removed queues and delete all job information on those jobs and on any waiting or held jobs. It is not recommended as standard practice.

PART II
Maintaining Your System

11

Equipment And Environment

KEEPING YOUR SYSTEM UP AND RUNNING

Once your system is up and running, you, as System Administrator, are responsible for ensuring that it stays up and running. Unless you are also the system operator, this does not mean that you are responsible for the minute-to-minute operations. It means that you are responsible for:

- Making the decisions that control the day-to-day and minute-to-minute operations.
- Setting up rules for the maintenance of the system and its environment.
- Deciding when to call your Prime support personnel, such as analysts and field engineers, for assistance.

Day-to-Day Operations

The day-to-day operations of your system can be considered under two headings:

- Environmental and hardware maintenance.
- System maintenance.

Environmental and hardware maintenance includes such things as determining maintenance schedules for your machines, establishing rules of behavior for the machine room, controlling the machine room environment, and dealing with hardware problems. These subjects are discussed in this chapter.

System maintenance includes setting schedules for backups, monitoring system usage, controlling interactions between the users and the machines, and dealing with software problems. These subjects are discussed in Chapters 12, 13, 14, and 16.

MAINTAINING THE HARDWARE AND ENVIRONMENT

In the following discussion, the term "hardware" is used to mean any physical part of a computer system, such as the CPU, disk or tape drives, printers, terminals, and other peripherals. The term "environment" is used to mean the habitat in which, and the conditions under which, the hardware is set up and functioning. This includes such things as the machine room, temperature and humidity controls, and air filtration equipment.

Some of the common hardware-oriented tasks for which you are responsible are:

- Defining user and operator procedures for handling hardware and environmental processes.
- Establishing a set of machine room rules.
- Deciding how maintenance and cleaning tasks will be done.
- Defining safety rules, including what to do and whom to contact in an emergency.

User and Operator Procedures

You can decide whether or not users are allowed in the machine room. In general, it is a good idea to restrict access to only those people who are essential to the operation of the machines. If you do allow users into the machine room, you may need to make sure that they are trained to use the machines correctly.

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.

In order for the interface between users and machines to be a smooth one, you may want to set up:

- Rules about who may use the machine room and such machines as printers and plotters.
- Training sessions for everyone 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 your operator(s) of any problems with the hardware and software. (Users should always inform some responsible person of problems with terminals or other equipment. They should never attempt to do repairs themselves.)

Procedures will vary from installation to installation. You, as the person who knows your situation best, can decide on the procedures for your installation. Your system analyst will be glad to help you.

DISK AND TAPE HANDLING AND STORAGE

Handling

Disks and tapes should be handled with care. Disks in particular are fragile. Improper handling of disks may result in an injury to the disk or its surface that will cause a head crash when that disk is used.

Even when inside their protective covers, disks should be handled with care. They should not be carried in large piles. If they are, they are likely to be dropped, and dropping a disk will probably result in the disk being injured by distortion of the platters or cracking of the magnetic surfaces. Such injuries will probably result in a head crash if the injured disk is used. If a disk has been dropped, it should not be used again until it has been professionally inspected.

Disk drives should not be banged or kicked when a disk is mounted. Such actions are likely to result in a head crash.

Tapes are rather more robust than disks. The main problems in handling are that the tape can become unwound or scratched. If a tape is scratched, data may be lost. If the loss occurs at the beginning of the tape, the entire tape may be unusable. Fingerprints are also a problem on tapes.

Storage

The storage requirements for disks and tapes are approximately the same. (The temperature and humidity ranges for tapes are a little larger than those for disks.) If you are unsure whether your disk and tape storage meets the requirements, consult your field engineer.

A log book should be kept in the storage area. It should contain information about every disk or tape in the archive. All disks and tapes should be marked with their contents and date of creation, and this information should be kept in the log book. Whenever a tape or disk is taken from storage, a note should be made in the log showing the name and date of creation of the tape or disk, the date of withdrawal, and the name of the person taking it. This will enable a check to be kept on the whereabouts of all storage media.

If you have any storage media that contain confidential information, you may want to have a special secure area in which to keep them. This may be a locked strong box, cupboard, or room, depending on the number of disks and/or tapes that are to be stored there. If you have such an area, you may want to have special rules for access.

MACHINE ROOM RULES

Your machine room contains various devices that are designed to keep your computer system at the right temperature and humidity, and to exclude most environmental contaminants. These devices include such things as heating systems, air conditioners, air filters, sealed windows, and anti-static mats. Environmental problems will occur if operators or users circumvent these devices. Therefore, it is important to establish rules that will maintain the necessary environment.

General Rules

There are two rules to which there should be no exceptions:

Rule 1. No smoking in the machine room.

Rule 2. No food or beverages in the machine room.

There are three additional rules that should be enforced as closely as possible:

Rule 3. Keep the machine room free of dust and other contaminants.

Rule 4. Maintain the machine room environment within the temperature and humidity limits specified by your field engineer.

Rule 5. Keep the machine room closed to unauthorized personnel.

You will probably have other rules that are essential to your installation, but these five are essential to the well-being of your machines. The rules, and the reasons behind them, are discussed in the remaining sections of this chapter.

Installation-Specific Rules

You know the special requirements of your installation and should 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, and determine such things as:

- Who performs what functions.
- How magnetic media (tapes and disk packs) are moved and stored.

Smoking, Food, and Beverages

Many people do not think of smoking, food, or beverages as contaminants. To a computer system, and particularly to disk and tape storage media and their attendant drives, they definitely are. Even a smoke particle or a fingerprint is larger than the distance between a disk's surface and the moving read/write head above it. It is easy to see, therefore, how a head crash can be caused by careless handling of a disk or by the intake of smoke through the drive.

In addition to not eating or drinking in the machine room, all personnel should wash their hands before handling magnetic media. A donut eaten at coffee break can leave a residue on the fingers that may cause major problems. This is particularly true with regard to reel-type tapes, where the recording surfaces are always handled to some degree during a load. If the surface of the tape becomes sticky, the contaminant may be transferred to the read/write heads during normal operation.

Dust and Dirt

Dust can be a major problem. A speck of dust on one of your disks can cause a catastrophic head crash and the loss of many days' work. While you cannot completely eradicate the possibility of a head crash, you can make it much less likely to occur, and make sure that, if it does happen, you can recover from it. Recovery from head crashes and other data loss situations is discussed in Chapter 13, BACKUPS.

Paper dust from a printer can be a major source of airborne dust. Make sure that the printer is vacuumed regularly. This operation can be carried out by your servicing agency or by your own personnel.

If you have a machine room with filters on the air intakes to trap airborne dust, do not leave the doors and/or windows open. If you do, the air filtering system will not work. If your machine room does not have filtered air, you can reduce the amount of airborne dust entering the area by keeping the windows sealed and the doors closed as much as possible.

Cleaning

All machine rooms should be cleaned regularly. Use vacuum cleaners, not brooms or dry mops, which throw dust into the air and thus increase the chances of dust getting into places where it is unwelcome.

The air filters on machines such as disk drives and the read/write heads on tape machines should be cleaned regularly. There are several cleaning operations that can be done by your staff. Others should be done by your field service person. Consult your field engineer to discuss which cleaning operations may be carried out by your staff, which should be left to field service personnel, and how often the cleaning should be performed. Once this is done, you should be able to:

- Draw up in-house maintenance schedules and define methods and rules for those jobs that can be done by your own personnel.
- Draw up maintenance schedules with the servicing agency for those jobs your staff are not going to do.
- Draw up schedules and define methods and rules for machine room cleaning.

Environmental Controls

Environmental controls are important. Your machines are designed to give optimum performance only within the range of operating environments specified by your system installer. Moreover, if you do not conform to the environmental specifications, you may invalidate your sales or support contracts. Prime computer systems are designed to operate at temperatures between 68 and 78 degrees Fahrenheit (20 and 26 degrees Celsius) and at humidities between 40% and 60%.

If your machine room is regularly exceeding the maximum temperature or humidity requirements, do not try to solve the problem by opening the doors or windows. If you do so, you will probably let in dust and cause even more problems. The best way to resolve this type of problem is to consult with your manager and your Prime analyst.

If you have a major 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 result in a series of head crashes and other problems that are troublesome and expensive to resolve.

Often, cables and boxes of supplies such as printer forms creep into the machine room and occupy spaces that are supposed to be clear. Do not allow anything to encroach on the clear space around your machines. If you do, you may have problems with accidents and obstructed exit routes. The obstructions may also impede the airflow around your machine. This can cause overheating, even if you have reliable air conditioning.

Unauthorized Personnel

Unauthorized personnel in the machine room can cause problems. The problems are generally of two kinds: misuse of the supervisor terminal, and mishandling of equipment. People trying to do such things as load a tape on a tape drive or a new box of paper onto a printer can do a great deal of harm to your equipment if they do not know the correct method. Mishandled machines can develop problems; your computer system is no exception.

You are responsible for deciding who is allowed into the machine room. Your operators must have access to it. There may be some users who need to be in there too.

Keeping the machine room doors closed will help to keep unauthorized people out of the machine room. If you can't or don't want to lock the machine room, you should ensure that every person who is allowed in is adequately prepared to use the machines it contains.

EMERGENCIES

There are many kinds and degrees of emergencies, covering a range from the system suddenly going down, perhaps from a electrical problem, to the illness of a key operator, to a disaster such as a fire that destroys the entire machine room. This section is concerned with the less extensive kinds of emergencies. Disasters are discussed in Chapter 15, SECURITY.

Accidental Data Loss

The most common non-disastrous problems are those involving the accidental erasure of data from a storage medium. These problems can be caused by such things as system crashes, disk crashes, loss of power, voltage spikes, and human mistakes.

> **Caution**
>
> It is imperative that a disk which has been involved in a head crash is never again mounted on any drive. If it is, the read/write heads of the drive it is mounted on will be ruined by the loose magnetic oxide from the damaged disk surface. Similarly, the drive involved in the original crash must be serviced before another disk is mounted and used, or the new disk will be ruined.

In most cases, accidental data loss should not prove disastrous provided you have a set of recent backup tapes or disks. At worst, your users may lose any data entered since the last backup. More often, they will lose nothing.

Probably the most common effect of the system going down is that any data not yet saved will be lost. While this is a definite annoyance, the data should be easily recoverable if there has not been a large amount of data entry since the last backup. (If you have repeated crashes, you should take tape dumps for your analyst's use.)

Data loss can often be avoided or its impact reduced by good habits on the part of users and operators. If the people using the machine room follow the rules for excluding contaminants, and users are encouraged to save their data often, data loss will be lower, even when a problem does occur. Chapter 15, SECURITY, discusses some methods of reducing the likelihood of system problems.

System Halts

There are many things that can cause a system crash. Fortunately, they are generally rarely encountered. Some of these causes are easily discovered; others are not.

If your system is crashing often, the first thing to do is to see if the crashes share any common conditions. For instance, have the crashes always occurred during a thunder storm, or is the system situated near a potential source of electromagnetic radiation such as an arc-welding shop or a physics laboratory? Looking through the system log book and through any COMOUTPUT files generated during system monitoring sessions around the time the trouble started can often disclose such coincidental occurrences.

If the system has previously been stable, you should check to see if any change has occurred in the area around your installation. Has a new company moved into the area? If so, does it have machinery that produces electromagnetic radiation?

When you have surveyed the possibilities, you may have the answer to your problems. If you have, call your system analyst and discuss methods of alleviating the problem. If you haven't, call your system analyst for help.

Accidents

In many cases, you are in a position to prevent accidents happening. A first step is to look over your machine room and list the possible danger points. When you have found any hazards, you can take whatever actions are necessary to remove or, at least, reduce them.

An instance of a potential danger is a cable snaking across the floor. This is a potential accident-causer for at least two reasons:

- A cable can be tripped over, perhaps causing a broken bone.
- It is a potential source of an electrocution hazard.

Electric Shock

Electric shock (while extremely rare) is potentially the greatest hazard in your computer room. Computer systems use high voltage electric currents, and such currents are dangerous unless they are handled correctly. In many cases, the effects of an electric shock can be mitigated by the swift application of cardio-pulmonary resuscitation techniques. If possible, have at least one person trained in CPR in or near the machine room at all times. (This person may also be useful if an employee has a heart attack.)

All personnel should be made aware of the possiblity of electrical hazard. They should under no circumstances touch any internal components of your system.

12
System Monitoring

INTRODUCTION

Whether your system is healthy or having problems, it is useful to know how it is running. There are two ways to keep track of system events:

- Through formal event logging mechanisms.
- Through the use (either scheduled or occasional) of system monitoring commands.

This chapter discusses both methods.

EVENT LOGGING

There are two mechanisms for system logging:

- Logbooks, the contents and format of which are defined by the System Administrator and entered by the operators.
- Software event loggers, which are kept by the system, with some parameters supplied by the invoker through a command line.

Site-specific operator entries to logbooks are handwritten entries to an actual book. Prime does not define what information goes into such a logbook; the System Administrator does. The next section of this chapter discusses the purposes of logbooks and suggests some of the items that should be entered into them.

Prime supplies automatic event loggers for the system and the network. Most events that are logged automatically are concerned with the status of the hardware, network operation, or the operating system. The section called EVENT LOGGING discusses Prime's event loggers.

THE SYSTEM LOGBOOK

Every system should have a logbook for recording information about system status and operation. The System Administrator is responsible for deciding what goes into the logbook. The system operators are responsible for entering the required information. You must ensure that all operators know what to enter into the logbook, and how to enter the information.

The Purpose of the System Logbook

A system logbook is used primarily 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, your system support personnel may have some clues as to the reason for the problem. They will then be able to find (and cure) the problem faster than they would otherwise be able to do.

Logbook Formats

The following list contains standards and procedures that have been found useful by Prime's system operators.

- Logbooks should be numbered and dated with the dates of the first and final entries.
- Logbooks should be bound. Loose-leaf pages are easily detached and lost, particularly if they are often referred to.
- Logbooks should always stay flat when open. This makes them a lot easier to write in.
- The page size should allow printouts and listings to be pasted in. However, the exact page size is not important.
- Each entry should be labelled with its date and time. This gives an historical record, which is useful in reconstructing how a system crash or other unexpected event occurred. It also enables such events to be correlated with external events, such as power failures.

- Each entry should be signed or initialed by the person making the entry. Your support personnel (field engineer or system analyst) will then know whom to ask for further information about a specific event.

- All entries should be made in ink, not in pencil or erasable ink. Any incorrect entry should be neatly crossed out and initialed by the person deleting it.

Logbook Contents

The exact contents of your logbook are up to you, since you are the only person who knows the exact needs of your system. However, the following lists show some types of information and events which should be recorded in any system logbook.

Hardware Information:

- The physical system configuration, including the model number and serial number of every piece of equipment. It may be helpful to list each type of machine together 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 — that is, any addition, deletion, alteration, or substitution of any piece of equipment.

- Any change in the operating status of any component — that is, component failure, unexpected occurrences (even if not fatal), etc.

Environmental Information:

- Any abnormal temperature or humidity conditions. The date, time, and duration of the conditions should be included, if possible.

- Any other unusual conditions, such as smoke, dust, or chemical spillage. Again, note the date, time, and duration of the conditions, if possible.

- Any unauthorized access to the computer room, together with the date and time that the unauthorized access was discovered, and the name of the person by whom it was discovered.

- Any loss of or damage to equipment, together with the date, time, and cause, if known.

- Any unauthorized use of the computer, including attempts at remote login.

- Any other unusual or unexpected events or results.
- Any action taken to correct any environmental problem.

Software Information:

- The listing of the system startup file C_PRMO. If you have several alternate configurations, listings of all the alternate startup command files.
- A listing of the CONFIG data file.
- A list showing 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 CMDNCO and the library directory LIB.
- A listing of the memory loadmaps RING0.MAP and RING3.MAP for the version of PRIMOS used by the system.
- Any addition, deletion, alteration, or replacement to any of the above.

Note

All terms used in this list are discussed in Part I of this book.

Operations Information:

- Every system startup. Any special conditions (such as the omission of the BATCH or FAM I system startup) should also be noted.
- Any use of the FIX_DISK utility, together with the name and physical device number of the partition being processed and the result of the operation.
- Any disk formatting performed, together with the name of the partition(s) created.
- Information about any backups performed, including the name(s) of the partition(s) 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), and the media statistics (such as tape speed and density).
- The name of any file or directory restored to the system, together with the date, time, and reason for the restoral.

- The name of any file or directory that is archived (removed from the active disks to storage for possible later use), together 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 to be kept.

- The date, time, and place of storage of any Event Logger printout. (The Event Loggers are described below.)

- The addition, deletion, alteration, or replacement of any commands in CMDNCO or libraries in LIB, together with the date, time, and reason for the action.

- Any shutdowns that occur, together with information about their extent (partial or complete), date and time, and cause (such as environmental factors, plant shutdown, configuration change, or system update).

- Any directories that are added to or deleted from the system.

Information on Halts:

- The status of the system when it halted. This 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 the Prime 750 and 850, DSWPARITY).

- If the system halted on an uncorrected parity error, the contents of the X, A, and B registers should be recorded.

- The type of start required after the HALT — that is, was a WARM or a COLD start required?

- After the restart, the behavior of the machine should be noted at various times; for instance, did the system function correctly immediately after the restart? Did it continue to function correctly after a half-hour?

In addition, if more than one halt has occurred recently, you may want to do a "crash dump" onto tape. (Instructions on dumps, as on other procedures for handling halts, are explained in Chapter 13 of the System Operator's Guide.)

Note

System halts are described in detail in the System Operator's Guide. The information listed above is the minimum that should be recorded in the system logbook during or after a halt.

EVENT LOGGERS

An event logger is a mechanism that automatically records information about significant system or network events. Events which get logged include cold starts, machine checks, disk errors, network link problems, etc. The output from these loggers can be useful in tracking any problems that may arise, especially those that develop or worsen over a period of time.

PRIMOS includes two event loggers: one which controls system event logging and one which controls network event logging.

Event logging, and the contents of the events logs, are discussed in the System Operator's Guide. The following discussion provides a brief overview and some suggestions on the use of the event logs.

System Event Logging

System event logging may be enabled or disabled in two ways:

- The LOGREC configuration directive enables or disables logging when the system is cold started. To enable event logging, give a 0 or positive value to the directive in the CONFIG file: for example, LOGREC 0. To disable event logging, give the directive a negative value: for example, LOGREC -1. You would want to do the latter to prevent logging messages onto a write-protected disk.

- Once the system is running, the EVENT_LOG command can enable or disable event logging. "EVENT_LOG" or "EVENT_LOG -ON" enables logging. "EVENT_LOG -OFF" disables it.

System logs are kept in the directory LOGREC*. The files are kept in binary form. They are named LOG.MM/DD/YY, with MM/DD/YY representing the date on which event logging was last enabled. Thus, if you start your system on Monday, 7/12/82, with event logging enabled, the log file bears Monday's date. If you shut the system down and then restart on Wednesday, 7/14/82, a new file is begun, with Wednesday's date as part of its filename. However, if you do a second cold start on Wednesday — or if you turn event logging off and then on again with EVENT_LOG — then LOG.07/14/82 will be reopened, and new entries appended to it. In other words, you may have up to one log file per day, but not more than one.

Network Event Logging

The procedures for network event logging parallel those for system event logging. Network event logging may be enabled and disabled by:

- The NETREC configuration directive.
- The commands EVENT_LOG -NET -ON and EVENT_LOG -NET -OFF.

The log files are kept as binary files in the directory PRIMENET*. They are named NET_LOG.MM/DD/YY, where MM/DD/YY represents the date of either the last cold start or the last use of the command EVENT_LOG -NET -ON.

Access Issues

System event logging is performed by User 1, (SYSTEM). Network event logging is performed by NETMAN. Therefore, SYSTEM needs ALURW rights to LOGREC*, and NETMAN needs ALURW rights to PRIMENET*. System administrators and operators generally need to be able to purge and write to the logging files, via LOGPRT. Therefore, they need LURWD rights. $REST usually should be given LUR rights.

Note

The system event logging file can be opened or closed only by the EVENT_LOG command. An attempt to close the file with the CLOSE command will produce an "Insufficient Access Rights" message.

Error Handling

Any errors that arise during event logging are reported every five minutes to the supervisor terminal. The errors fall into three categories:

- Quota exceeded
- Disk full
- Disk shutdown

Quota Exceeded: In order to prevent log files from consuming an undue amount of disk space, you may want to set a quota on LOGREC* and/or PRIMENET*. If you do so, and if the quota on either directory is exceeded, one of the following messages will be sent to the supervisor terminal every five minutes, until more space is created for the files.

- Exceeding quota on LOGREC*. System event logging not taking place. (LOGEV2)

- Exceeding quota on PRIMENET*. Network event logging not taking place. (NETEV2)

For information on how to create more space, see the section called Controlling the Size of Event Logging Files, below. Alternatively, you can halt the messages by using the EVENT_LOG command to disable event logging.

Disk Full: If the disk on which logging is taking place becomes full (either because of uncontrolled growth of the log files or because other directories are using up all the space), event logging can no longer take place. In this case, one (or both) of the following error messages will be printed every five minutes at the supervisor terminal, until either event logging is disabled or space is made available on the disk:

- Disk full. System event logging not taking place. (LOGEV2)

- Disk full. Network event logging not taking place. (NETEV2)

Disk Shutdown: Shutting down disk 0 while event logging is enabled closes the event logging files. If the disk is then added back to the system, event logging must be reenabled with the EVENT_LOG command. If you do not reenable event logging, one of the following sets of error messages will be sent, once only, to the supervisor terminal:

- Disk has been shut down. System event logging not taking place. (LOGEV2)
  Disk has been shut down... Please reenable system event logging. (LOGEV2)

- Disk has been shut down. Network event logging not taking place. (NETEV2)
  Disk has been shut down. Please reenable network event logging. (NETEV2)

Both types of event logging will then be disabled until you issue the EVENT_LOG commands to reenable them.

Other Events: All other errors will cause one of the following messages to be printed at the supervisor terminal every five minutes until you correct the error or disable event logging:

- error message. System event logging not taking place.

- error message. Network event logging not taking place.

Controlling the Size of Event Logging Files

If allowed to grow and multiply indefinitely, event logging files can consume large amounts of disk space. You can do two things to control the amount of space consumed. You can:

- Put a quota on the directories LOGREC* and/or PRIMENET*.
- Print out and delete old log files at regular intervals.

Using Quotas

You set the quota on an event logging directory with the SET_QUOTA command, just as you would for any other directory. (For information on SET_QUOTA, see either Chapter 5, DISKS, or Chapter *** of the Prime User's Guide.

Once the quota is filled, event logging stops. To warn you of this fact, an "Exceeding quota" error message (as shown in the preceding pages) appears every five minutes at the supervisor terminal, until you either:

- Increase the directory's quota.
- Use the LOGPRT command, with its -SPOOL and -DELETE options to print and then delete old log files (as explained below).
- Delete empty input log files with the DELETE command.

In normal use, exceeding the quota and losing some events from the file will not be disastrous. However, troublesome situations, when things are beginning to go wrong, cause more events to be logged than are logged when everything goes well. Thus, the times when you most need your log file are also the times most likely to cause quota problems. In general, then, it is best that you keep plenty of space in your directory by printing and deleting log files at regular intervals.

Printing Log Files

To print log files, use the LOGPRT and LOGPRT -NET commands. These commands are explained in full in the System Operator's Guide. The LOGPRT command converts the files from their binary format to a legible format. It then displays the file on your terminal or prints it as a spool file.

Once the file has been printed, it can be deleted from the directory. If you use both the -SPOOL and -DELETE options with LOGPRT, the file will be deleted when it has been spooled. In general, files should be printed at least once a week to keep your hard copy records up to date and to keep the directories clean.

Storing Log Files

General rules of thumb for storing log files are:

- Keep the most recent files with the system logbook, for quick reference.
- Store older files in a safe archive (such as a file cabinet). The amount of time files are kept varies from installation to installation. You can decide what makes the most sense for your system.

To sum up: when you are deciding how often log files should be printed and purged, and how long the printed output should be kept, you will want to consider:

- How much disk space you can allow the files to occupy.
- How much storage space you have for the printed files.
- How your system has been behaving over the last few weeks or months.

These things are highly variable, and only a person with first-hand knowledge of the installation can make such decisions. If you have any problems, talk to your systems analyst or field engineer.

MONITORING THE SYSTEM

System monitoring provides status information when requested. Monitoring samples and the event logs can disclose such things as the status of the CPU and various other parts of the system hardware, the status of the network, or which user is logged in on what line. Logbooks should contain information about external events which may cause problems, such as power failures, etc.

If you have a series of logs and reports from regular system monitoring samples, it may be possible to foresee trouble and forestall it. If the system does develop a problem, backtracking through the logs and COMOUTPUT files of monitoring sessions may disclose a use or event pattern, which may, in turn, disclose a cause. The logs and monitor output files are particularly useful for finding the cause of intermittent, unpredictable problems.

Prime systems contain several tools for system monitoring. Among them are the STATUS, USAGE, SET_QUOTA and LD commands. STATUS and USAGE are used to monitor various system status and use statistics, such as information about the state of the system hardware, the network, or users. LD provides information on the use of the file system. LD and LIST_QUOTA can be used to check the number of records used at the various levels in a directory tree.

Note

> LD (and LIST_QUOTA) are designed for Rev. 19-format disks. They cannot provide information reliably for Rev. 18-format disks.

Two additional tools are the SIZE command and the AVAIL command. The SIZE command can be used to find the size (in records) of any file. AVAIL is used to monitor disk space usage and availability.

All these commands output their information to the screen, not as hard copy. If you want a hard copy of the output, you should open a COMOUTPUT file before beginning the monitoring sequences. This file can then be spooled and printed.

This chapter is concerned with the reasons for and value of event logging and system monitoring, and, in particular, with how these tools can show you what is happening in your system. The commands and their options are fully documented in the System Operator's Guide.

Monitoring System Status: The STATUS and USAGE Commands

The STATUS and USAGE commands both monitor system usage. They are complementary. This section discusses their use by the System Administrator. (Both commands can be issued by any legal user from any terminal. However, there are differences in the output and defaults of the commands when run from a user terminal or the supervisor terminal. These differences are discussed below.) For a full discussion of how to use the commands and all their options and arguments, refer to the System Operator's Guide.

USAGE monitors events internal to the system at the hardware level, such as the total CPU time used since the system was started up, the number of input/output operations occurring per second through the sampling time, and per user CPU and I/O use statistics.

STATUS monitors higher level system events such as information about users, the status of devices and the network, the current version of PRIMOS, and the amount of physical memory.

The STATUS Command

The STATUS command displays the following information:

- The network node name of this system.
- All configured network nodes and their status (UP or DOWN).
- The size of main memory.
- The number of file units open.
- All currently assigned magnetic tape units by the physical and logical device numbers of the tape drive and by the user-id and number of the assignee.
- All currently started disks by physical and logical device numbers, partition name, and node name.
- A single logged-in user or all logged-in users by user number and AMLC line number.
- The disk partition in use by and any devices assigned to every logged-in user.

The STATUS command allows you to do such things as:

- Find out if anyone is using the system, when the system is about to be shut down.
- Find out if anyone is using a partition that is about to be backed up or reformatted.
- Identify the currently started disks.
- Identify which tape drives are in use and by whom.
- Find out which user is using which terminal.
- Find out what remote users, phantoms, and slave processes are using the system.

These types of information allow the operator to be aware of the state of the system before starting any system operation. The operator can then make sure that any users who may be affected by the operation are warned before the process is started, so that they can take whatever action is necessary to ensure that their work is not harmed. It is, therefore, useful if it is operating policy to run STATUS before every operation that may alter the user/system interactions; that is, such operations as shutting down the system for preventive maintenance, formatting a disk partition (using MAKE), performing a backup, etc.

There are slight differences in the way STATUS works at the supervisor terminal and at a user terminal. At the supervisor terminal, the

STATUS default is ALL. In other words, if you type "STATUS", the resulting output will be the same as it you type "STATUS ALL". At the user terminal, typing "STATUS" alone omits information about users. If you are monitoring system status from a user terminal, you must bear this difference in mind.

The USAGE Command

USAGE is a system metering tool. It can be used by any user at any terminal, unless you specify otherwise. The information it generates describes the status and performance of the CPU and other system internals. A sequence of one or more USAGE samples can be generated automatically or manually.

USAGE is a useful tool for the System Administrator. The USAGE command, its options, and its output are documented in the System Operator's Guide.

Using Quotas to Meter Disk Usage

The LIST_QUOTA command can be used to show how many records have been used by a directory (including its subdirectories, if any). In order to use LIST_QUOTA, you must normally have List access to the target and parent directories, and Use access to any higher level directories. However, this restriction can be overridden through the use of a priority ACL.

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 = 500
Total records used = 425
Records used in this directory = 28
```

The output shows that the maximum number of records that can be used by the directory TEST and all the subdirectories contained within it is 500. 425 of these records have already been used, leaving the directory (and all the subdirectories) 75 records before the directory tree runs out of space. The subdirectory STATS has used 28 records out of the total 425 records used.

If no quota has been set on the directory, a message to that effect is given in place of the maximum number of records. However, the total number of records used by the top-level directory and any by the subdirectories is still given.

LIST_QUOTA has an option that prints the quota data in tabular form. This option (-BRIEF) does not display a message if the directory is a non-quota directory. Instead the maximum number of records is given as 0.

The LIST_QUOTA command is fully explained in the PRIMOS Commands Reference Guide and in the Prime User's Guide.

Monitoring Disk Space Utilization: The AVAIL Command

For any specified disk or partition, the AVAIL command displays:

- The number of records used.
- The number of records still available for use.
- The percentage of records used.

Records used can be shown in two formats: physical records or normalized records.

The default gives the number in terms of physical records. A "physical record" contains 2048 bytes. The term "physical record" comes from the fact that this is the size of each "slot" for a user-data record on the disk. In fact, each record on the disk requires some identification data as well, so the total size of each disk record is actually 2080 bytes.

You have the option of displaying the information in the format of "normalized records". "Normalized" records contain 880 bytes.

Data from AVAIL can be shown for all partitions that are currently started (including remote disks), for a single named disk or partition, or for a disk or partition identified by a logical device number.

The AVAIL * Command: A particularly handy feature of AVAIL is the "AVAIL *" format. This command can be used to display data for all disks on the system, in tabular form.

To make the AVAIL * command work, however, the System Administrator (or the operator) must take the following steps:

- Use the Editor to create a file named DISCS within the directory SYSTEM.
- Place information on each of the system's disks within the DISCS file, so that the AVAIL command can use it. If the system is networked, you may also place information on remote disks in the DISCS file.

- Information needed is as follows:

  — The DISCS file must contain at least one column of text. This column must contain the names of all disks to be listed, one per line.

  — For fuller information to be printed by AVAIL *, the DISCS file may contain two or more columns of text. The first column, again, contains the disk's name. The other columns may contain (in any order) the disk's logical device number; the disks's physical device number; and any other information you wish to add: for example, the name of the system to which a remote disk is physically connected, or the fact that a disk is write protected.

- 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, giving the AVAIL * command lists the file's contents. For each disk that is actually running, it also gives information on space usage. For other disks, a message appears indicating that the disk is not running.

Here is an example of a DISCS file, and of the output from an AVAIL * command which uses that 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
```

Access Rights: If you want any form of the AVAIL command to be accessible to users, you must grant users Read (R) rights to the DSKRAT file on each disk, and LU rights to the MFD. In addition, if your disks are password protected, one of the passwords on the MFD must be XXXXXX. (If you don't want users using AVAIL, it's simplest to deny them rights to the AVAIL command itself, in CMDNCO.)

If you frequently run out of space on your disks, you may need more or larger disks. If you have several partitions, and only one or two of them regularly exceed this percentage, you should consider increasing the size of these partitions. However, if you do this, make sure than you will not be making any other partitions too small. You must also take care not to reformat a "live" disk, or the data on it will be lost. (For further information, see Chapter 5, DISKS.)

The LOOK Command

One further monitoring command exists. However, it is rarely used by administrators, being intended as a tool for system analysts. Its format and use are as follows:

LOOK [-userno] [segno] [access] [mapseg]

LOOK is an internal operator command that provides access to any segment in the system.

userno Number of the user owning the segment (default is user 1).

segno Number of the segment to be examined. The default is '6000 (the Ring 0 stack segment for the user).

access Access rights to be granted (as in the SHARE command). Default is '200 (read-only).

mapseg Segment of user 1's address space into which the specified segment is to be mapped. The default is '4001.

> **Caution**
>
> This command is intended mainly for the use of systems engineers and field analysts as a debugging tool. The operator and administrator will normally have no use for it. Misuse of the LOOK command can destroy system data.

If the LOOK command involves an attempt to examine a segment that does not exist, an attempt to write to a segment that does exist, or attempts to map either shared or stack segments with write permission, the command is considered risky or dangerous to system integrity. The REALLY? prompt is issued for any LOOK command whose request is considered to be risky or dangerous to system integrity. A YES response allows the operation to proceed.

13
Backups

WHY DO BACKUPS?

A backup operation is a procedure for making copies of the current contents of the system's on-line storage devices (that is, disks). These copies are available if any of the data is lost, or if users want copies of files as they were at the backup date. This chapter discusses the things that you should consider when deciding how, when, and what your system should back up. The commands used to perform backups are discussed and documented in the System Operator's Guide.

Data loss may be caused by a number of things. Major losses may be caused by:

- Environmental problems, such as overheating in the machine room.
- Operator error, such as running MAKE on a disk containing in-use data.
- A disk crash.
- Fires, floods, and other catastrophes.

Minor losses may be caused by:

- Power failure during a write operation.
- Accidental truncation or deletion of a file by a user or an operator. (This is the most common cause of data loss.)

If your system suffers a major loss of on-line data, your only hope of recovery is to have a recent copy of the lost data. Such a copy will have been created by your most recent backup operation. Using this copy, you can restore your entire data base as it was on the date that the backup copy was made.

If the loss is minor, you will need to restore only a small part of the system. On a system with good backup procedures, either major or minor restoration can be performed easily whenever necessary, and data entry operations kept to the minimum.

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. Each of these generations is kept in a different location. When a new backup is made, the generations are rotated, so that the oldest is deleted.

At least one of these generations should be kept off-site, preferably in a completely different building. At the very least, one copy should be kept in a fire-proof vault. These precautions provide the greatest likelihood that most of the system's data can be recovered, even if the entire computer center is destroyed.

The latest backup disk or tape should not be kept in exactly the same place as the originating data, but it should be easily and quickly obtainable in the event of a disk crash. This copy is the one which will restore data to the system in the form which requires least up-dating.

The intermediate copy should be kept in a secure (and preferably fireproof) location, different from the location of the latest copy, but possibly in the same building. This will allow it to be quickly accessed if both the current disk and the first level backup are destroyed, but the machine room and the disk and tape drives are not.

The oldest copy should be kept off-site, if possible. (The oldest copy is usually used as the off-site copy because the off-site copy is the one least likely to be needed.)

Data Archives

Backups can also be used to create data archives. Data archives contain copies of inactive files that may be required at some future time. Once an inactive file is archived, it may be removed from the disk, thus freeing disk space for active use.

You may want to have your normal backups serve for archiving as well as for security against data loss. In this case, you would be essentially archiving your whole data base, and would plan to keep copies for a

relatively long time. Alternatively, you could keep archived material separate from "backed-up" material. Under this scheme, archived material would be viewed as long-term storage, perhaps for indefinite lengths of time, while backup copies would be short-term storage, with the oldest disk or tape being re-used as soon as two or three newer copies existed.

GUIDELINES FOR BACKUPS

Each site has different needs for backups. You are the one who decides:

- What data is to be backed up.
- What types of backups to use.
- When to do backups (how often, and at what hours).
- How long to store backup copies.

The rest of this chapter provides some guidelines for you to use in making these decisions. While reading it, you will want to consider the following questions:

- How important is your data?
- How much does your system data change from day-to-day?
- How often can you perform backups, given that you must restrict access to the disk while performing the backup, and that the task requires some operator time?
- How long can you take to restore the system to its precrash state?
- What type of backups are going to be performed? Are all backups going to be full backups, or are some of them going to be incremental backups?
- What media are you going to use? Are you going to do disk-to-disk backups, disk-to-tape backups, or a mixture of the two?
- Where are you going to keep the backup media?
- Do you have any users with special backup needs?

TYPES OF BACKUPS

A backup copy can be made to a disk or to a tape. There are advantages and disadvantages to both types of media.

Disk-to-disk Backups

Disk-to-disk copies are fast. Typically, a fully used 300 megabyte disk pack can be copied in about one hour. Smaller packs take less time.

More information can be held on a single disk than on a single tape. A full 300 megabyte disk would require nine reels of tape when recorded at a density of 1600 bpi.

A disk backup can be used directly without requiring a restore operation. The disk is immediately and rapidly accessible in the normal way, using the directory tree structures. (You can have both a current disk and one of its backups running simultaneously, if you change the name of one of the two disks — using ADDISK's -RENAME option — when you add it.)

However, disks are expensive, and they require special handling and storage because they have lower tolerances for mechanical and environmental changes. Disks are also less easily transportable from site to site than tapes.

Disk-to-tape Backups

Tapes have their own advantages. They are much less expensive than disks, even though several tapes are required to hold the same amount of data as a single disk. They are easier to store and transport.

There are two methods of disk-to-tape transfers. MAGSAV copies the data file by file. PHYSAV makes an exact copy of the disk contents as it appears on the disk. A PHYSAV tape cannot be used to supply a copy of 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 random access operation is possible. A PHYSAV operation that transfers a full disk takes less time and uses fewer tapes than an equivalent MAGSAV. However, MAGSAV allows the data to be restored on a file by file basis, and thus makes it possible to restore a single file quickly and efficiently.

Disk-to-tape or tape-to-disk copies are slower than disk-to-disk copies. The fastest way of copying a 300 megabyte disk to tape would be using PHYSAV. This transfer would take about 45 minutes at 6250 bpi; however, it would have to be restored to disk before use, which would take as much time again, for a total time of 1 1/2 hours.

Full or Incremental Backups?

There are two types of backup — full backups and incremental backups.

An incremental backup copies only those files that have changed since the last backup copy was made.

A full backup copies the entire contents of a disk to another disk or tape. Everything that is on the source disk is written to the target media, regardless of whether or not it has been altered in any way since the last backup.

Incremental backups may be used to supplement full backups. For example, if the volume of activity on the system as a whole is low, and does not require a full backup to be performed very often, but the volume of activity on a few files is high, incremental backups might be useful. In this case, the backup schedule would be set for full backups on the basis of the overall system activity, while incremental backups would keep the high-activity files up to date. Incremental backups are faster to make, because fewer records are copied. However, it is slower to restore a complete data base from them, as each increment must be reloaded in order.

PRIMOS or PRIMOS II Backups?

If you are backing up to tape, you must perform backups under PRIMOS. Otherwise, you will not be able to save access control or quota information.

If you are backing up onto disk, you can choose between doing backups under PRIMOS and under PRIMOS II. It is recommended that you do all backups under PRIMOS, since this is most efficient. Moreover, doing backups under PRIMOS allows you to run FIX_DISK as part of your normal backup procedure. Running FIX_DISK on a disk before you back it up is highly recommended. See the chapters on FIX_DISK and on backups in the System Operator's Guide for details. It is possible (though not recommended) to do disk-to-disk copies under PRIMOS II.

WHEN TO PERFORM BACKUPS

How often you perform backups depends on several factors, including how often the data in your system changes, how important it is that the data is up-to-date, and other installation specific details.

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 use time and 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, in the event of a disk crash, all the data entered since the last backup will have to be reentered to restore the system to its pre-crash state. The closer to the time of the crash that the backup copy was made, the less data will have to be reentered.

With a highly changeable system, you may find the incremental backup facility useful. Incremental backups can reduce the number of full backups required, and thus 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. A system that gets data from off-site (from cards or tapes, over a half-duplex network, etc.), processes it, and then sends the results off again may need few backups. Data is rarely resident on such a system 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, since it sees much data entry and many changes to data files.

The third factor to be considered is the system resource factor. System resources may be the physical plant (disk and tape drives), or they may be personnel.

There are three system resource considerations that will influence the timing of backups:

- First, the type of media used affects the time that a backup takes. Disk-to-disk backups are faster than disk-to-tape backups.

- Second, backups require that any disk used be accessible only to the person performing the backup during the time that a backup is taking place.

- Third, the amount, type, and timing of the use that your system gets must be considered. If your system is very busy all through the normal office "day", you will need to schedule backup times out of the normal work day. If some of the disks are busy at certain times but idle at others, then you should take this into account when scheduling backups.

Operator time can be conserved by running backups from a CPL program or a COMINPUT file.

The following example shows how a typical development system might be handled.

- The system consists of a Prime 750, with 2x300 megabyte storage modules and 1x80 megabyte storage module.
- All backups are full backups because all the data on the system must be absolutely up-to-the-minute and quickly restorable.
- Most of the system activity is concentrated on the 300 megabyte drives.
- One of the 300 megabyte disks is backed up to another disk on Monday, Wednesday, and Friday mornings before normal working hours.
- The other 300 megabyte disk is backed up to disk on Tuesday and Thursday mornings.
- The 80 megabyte drive is not as active. It is not backed up during the week.
- 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 are kept for two years.

The degree of data protection given by this regimen is probably well in excess of that required by most installations. It must be emphasized that each installation has its own special requirements, which can only be decided on the basis of knowledge of that installation, in the area of backups as in all others.

14
Looking After Users

INTRODUCTION

Users call on the System Administrator for many things, including:

- Access to the system.
- Help in a difficulty or an emergency.
- Information about the system, in particular about site-specific details.

Operators and Project Administrators may assist you with many of these tasks, but some duties remain yours. These duties are concerned with setting up and administering the User Profile data base, as explained in Chapters 1 and 4.

This chapter will review briefly the tasks involved in adding a new user to the system. It will then deal with some common problems that users may bring to you, and suggest ways for you to deal with them.

ADDING USERS TO THE SYSTEM

Before a user can log in and use the system, you must supply him or her with two things:

- A set of user attributes.
- One or more sets of project attributes.

The user attributes must include a user-id and a password (possibly null). They may also include a default project membership, and membership in one or more system-wide ACL groups.

The minimal project attributes are the user-id, placed in the project data base, and an Initial Attach Point. The Initial Attach Point may be specified for the user, or it may be the project's default IAP. In addition, project attributes may include membership in one or more project-based ACL groups.

You define user attributes and project attributes with the EDIT_PROFILE utility, as explained in Chapter 4.

Origin Directories

It is not enough to define the user's Initial Attach Point (or origin directory) with EDIT_PROFILE. You (or someone else) must also make sure that the directory really exists! Users do not have to log in to top-level directories at Rev. 19. Their Initial Attach Points may be anywhere in the directory structure. Often, therefore, if a new directory is to be created for a new user, the user's project leader or supervisor can create it. Still, there may be times when new top-level directories need to be created for users to log in to; and you may be the only person who has enough rights to the MFD to create those top-level directories.

HELPING USERS WITH PROBLEMS

Helping a User Who Can't Log In

The action you take in this case depends on what message the user receives when trying to log in.

If the message is "Incorrect user ID or password", try the following:

- 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 check to see that the user-id given by the user is actually registered in that system's SAD. (Use EDIT_PROFILE to check this out.)

- If the user-id is correct, find out whether the terminal the user was using is connected to the proper system. If not, the user must either use a different terminal or do a remote login.

- If the user-id is correct, and the user is trying to log into the right system, then the password is probably incorrect. Since passwords are not stored in readable form, and since users can change their passwords, you will not be able to check the password. In this case, you should probably assign the user a new password with EDIT_PROFILE. The user should then try to log in again, using the new password. (The user can retain the new password, or change it later, as he prefers.)

If the error message indicates that the user couldn't be attached to the system, then either the user's Initial Attach Point has not been entered into the SAD correctly, or the directory itself either has not been created or has been deleted. Check the SAD with EDIT_PROFILE to find out what the Initial Attach Point is; then check the relevant disk to make sure the directory is there.

If the message is "Invalid Project ID", then 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.) You can check the user's project affiliation with EDIT_PROFILE's LIST_USER command. (The format you want is "LIST_USER user-id -ALL".)

A user who is not accustomed to specifying a project-id may come to you because the system is suddenly demanding a project-id at login time. This happens when the user's default login project has been deleted. Either you must give the user a new default login project, or the user must specify a project-id at login.

Helping Users With Access Problems

Users may come to you because their programs are failing because of access problems. These problems are signalled by messages such as "Insufficient access rights" or "Top-level directory not found or inaccesible".

You have several choices of how to handle this situation. Probably the main factor involved is time. Situations in which time is at a premium — for example, when an "end-of-the-month" accounting package won't run -- call for different handling than do less critical situations. This section suggests strategies for both cases.

In Ordinary Situations:

- Your first step may be to find out exactly where the access violations are occurring, and what protection is causing them.

- Next, you may want to check whether this user really should have the right to run these particular programs, or to access the data he's being denied.

- Finally, when all the facts are in, you can decide how to remedy the situation, so that the programs in question work correctly for those users who need them.

In Time-critical Situations:

- Impose a priority ACL that allows the program to run. You may want the ACL to allow the original user to run the program; or, you may choose to run the program yourself.

Helping Users With Disk Space Problems

On a system where directory quotas are in use, users may have problems with quotas that they cannot solve. These will require your intervention.

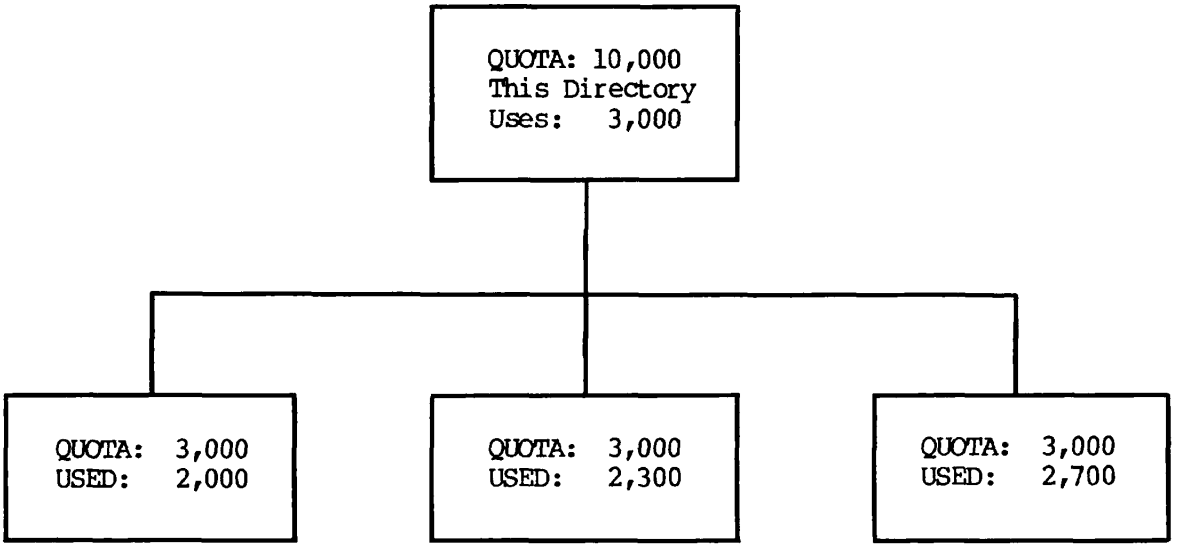
The most common case is this: A user gets a message, "Maximum quota exceeded". He gives the LIST_QUOTA command, and discovers that he has unused records left. He comes to you to find out why he's getting the two conflicting messages.

What has probably happened is that the quota has been exceeded, not in the user's own directory, but in some other directory. Figure 13-1 shows examples of ways that this could happen. If the user has List rights to the parent directory, and to its parent, he can trace the quota problem himself. If the user does not have the necessary access rights, you will have to do the checking for him. When you find out which directory is causing the problem, you can do one of two things:

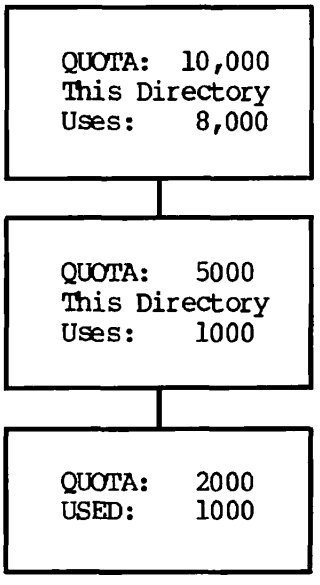
- Grant more space to that directory.

- Request the users of the parent directory, or other users of subordinate directories within that tree, to clean out those directories. You may have to archive some files for them to make the cleanup possible.

If users get a "Disk full" message, you have no choice but to see that old files are archived and/or removed from the disk. Careful monitoring of the system should allow you to warn users when disks begin to get full, so that the users can delete old material before the "Disk full" errors occur. However, if disk usage keeps creeping up, and users are unable to clean out enough files to keep usage below 80% or 90%, you may need to add more disks to the system.



No subdirectory has used its quota; but none can add records, because the top-level directory's quota is full.



This time the blockage occurs even higher in the tree. Users in the lowest directory must search up two levels to find it.

Some Cases of Unexpected Quota Errors
Figure 13-1

15
Security

SECURITY FOR YOUR SYSTEM

Computer security has three aspects: security for the physical plant, security against illegal access to the system itself (login control), and security against illegal access to data after login. No two computer installations have exactly the same security requirements.

The System Administrator is usually the person who makes security decisions. For hardware security, this means making the sort of decisions discussed in Chapter 11, EQUIPMENT AND ENVIRONMENT. These include:

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

However, hardware security also involves keeping track of equipment used outside the computer room: for example, terminals and modems. To ensure security for this more portable equipment, you may want to:

- Keep an up-to-date inventory of all equipment: what it is, what its serial number is, and where it is currently kept. (You may want to keep a separate inventory of tapes and disks.)
- Label all portable equipment in some indelible manner.

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

Software security, too, involves two main groups of decisions:

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

- How to control the authorized user's access to files and directories after login (often called "data access" or "data security").

Prime software allows you to coordinate both types of control through the User Profile system. This chapter will discuss how it does so, and how you may take advantage of it.

LOGIN AND DATA SECURITY

Login security and data security are both important. Only those people entitled to use the system should be able to do so, and only those people with a valid need should have access to data, whether program data or file data. Improper use of the system could cost your company, both in money and in loss of sensitive data.

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

- There was no default protection mechanism.

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

- Login and data security were rudimentary because of the requirement that passwords be communicated — either by word of mouth, in writing, or in source code — to anyone who might need to use them.

At Rev. 19, system access is controlled by the User Profile system. Data access is provided by Access Control Lists (ACLs).

There are a number of advantages to this system:

- 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 can have a set of specially tailored access rights. No two users need have the same rights.
- Access to the system can be 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.
- Once the access controls are set on a file or directory, no password or other transferable information is required for a "guest" user to use the data.
- A password can be a requirement for login. This password can be different for every user.
- Passwords are recorded (in the machine) in an encrypted (scrambled) form, so that they cannot be read by humans or be unencrypted.
- If required, passwords can still be set on directories.

LOGIN SECURITY

Rev. 19 of PRIMOS supplies a new login security system to help you set up and maintain system access controls — the User Profile system. This system builds a data base which must have an entry for every authorized user. The data base is consulted by the computer whenever a user attempts to log in. It contains a set of tables mapping the user-id, the login password (if any), and the project (or projects) to which this user-id is allowed access.

In addition to this powerful tool, you can write an external login program to control where, when, and how a user can login. If you have written external login programs for your pre-Rev. 19 system, you may wish to alter these programs to take advantage of Rev. 19's more powerful security features. A sample external login program may be found in Appendix B.

LOGIN PROCEDURES

The Rev. 19 login procedure requires that all users supply, at login, a "user-id" under which they will operate for the duration of this login. In addition to the user-id, users are required to supply a (possibly null) login password and a project-id.

User-ids

A user-id must be registered in the User Profile data base. Ideally, every user will have a unique ID, but you may decide that a group of people may use the same ID. People in such a group will share the same system restrictions and privileges. Allowing several people to share an ID produces a possible decrease in security, but this may be offset by the simplicity of providing an identical operating environment for many people in a single operation. The User Profile data base is discussed below.

A valid user-id consists of an alphanumeric string of up to 32 characters. It must begin with a letter. Letters, numerals, and the special characters ., _, and $ are allowed in a user-id. Examples of legal and illegal user-ids are shown below:

Legal IDs	Illegal IDs
ALICE	%ALICE
MARC_ANTONY	CAESAR&CLEOPATRA
AS.P6	6A4B
AS_$6	AS-6
A.S_6$P	$A.S

If possible, allocate user-ids that are not the given names or initials of your users. IDs using given names or initials are more easily guessed than IDs that have personal significance but are not as easily attachable to a specific user.

Your system will be more secure if every user has a unique ID. 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 to a ringnet, which can almost be considered to be your system, but also any other system that regularly accesses your system through PRIMENET or a PDN. The EDIT_PROFILE command VERIFY_USER allows you to determine if and where within your system pool a user-id is duplicated. (Networks are discussed more fully in Chapters 17 and 18.)

Login Passwords

A login password is an alphanumeric character-string of up to 16 characters in length. The password may contain any characters except PRIMOS reserved characters.

A password may be "null". In this case, no password is entered as part of the login sequence. You must decide how login passwords are to be used on your system. There are two basic questions to consider:

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

Systems where users are required to log in with a non-null password, and where echoing is suppressed, are the most secure.

Requiring Non-null Passwords: In the interests of security, you can insist that all users have valid, non-null login passwords. This is enabled by the NO_NULL_PASSWORD command within EDIT_PROFILE. NO_NULL_PASSWORD is discussed in more detail in Chapter 4.

If null passwords are allowed, a user logging in must supply a user-id and, possibly, a project-id. This decreases the degree of security by one level. If there is no project-id required, an unauthorized user may be able to gain fast access to your system, 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 get all the parts of the login entry correct by guesswork.

Requiring "Quiet" Entry of Passwords: If a password is required, it may be entered in one of two ways. In the first way, the user logs in by typing "LOGIN user-id password". Typing the password on the same line as the LOGIN command allows the password to appear on the screen. This is called "echoing". In the second method, the user types only "LOGIN user-id". The password is omitted from the login line. In this case, PRIMOS prompts for the password. The password is not echoed, and is thus not displayed on the screen.

You may require that passwords be entered by this second method, to prevent echoing. This is recommended as the safest form of password entry because a user's login password cannot be accidentally discovered by someone else during login. It is enabled by the FORCE_PASSWORD command within EDIT_PROFILE, discussed in Chapter 4. When FORCE_PASSWORD is in effect, any user entering a password as part of the LOGIN command line will receive an error message and be refused entry to the system.

Note

When users log in at half duplex terminals, passwords will be printed, whether FORCE_PASSWORD has been specified or not.

Suggesting That Users Change Passwords: The CHANGE_PASSWORD command allows users to change their login passwords at any time. Your system can gain 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 will ensure that only one person (the person who performed the change) knows the password.

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.

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

Project-IDs

In order to log in, every user must be registered in the User Profile data base as a member of at least one project. A default system project may be provided for those users who have no true project affiliation. If you do not normally use projects, you must set up a default project, as explained in Chapters 1 and 4. All users will then be members of the default project.

If projects are in use to provide special operating environments, a project-id may be required at login. A user-id may be associated with several projects, each of which gives a different set of access rights and restrictions. When this is the case, the user signals the project she or he wishes to log in to by including a project-id on the login line. For example:

```
LOGIN JOE -PROJECT ALPHA
```

will log in a person using the user-id JOE to a project named ALPHA and:

```
LOGIN JOE -PROJECT ZONKER
```

will log in the same user (or another user using the same user-id) to project ZONKER.

A valid project-id consists of an alphanumeric string of up to 32 characters in length. It may contain the special characters . , _ and $. The project-id must begin with a letter.

Degrees of Security

As the preceding discussion suggests, the loosest security is provided by requiring only a user-id for login. The tightest security is provided by requiring a user-id, a non-null password, and a project-id.

HOW THE SYSTEM HANDLES LOGIN

The system responds to a LOGIN command line as follows:

1. It runs its own, internal, login program. This program checks the supplied user-id, password, and project-id against the records in the system data base to verify that the person attempting to log in is in fact an authorized user of the system.

   At this time, the system also establishes the user's membership in any ACL groups which will be active during this session.

   If the user fails validation, the login procedure terminates at step 1. If not, it continues.

2. The system now looks for a site-supplied external login program. (If present, this program must be named LOGIN — no suffixes are allowed — and must reside in CMDNC0.) If it finds such an external LOGIN program, it runs it. If not, it proceeds to step 3.

   The external LOGIN program may do accounting. It may also perform further validation tests. If the user fails these tests, login terminates. Otherwise, the system proceeds to step 3.

3. The system attaches the user to his or her origin directory. It then looks in that directory for a user-supplied login program (named LOGIN.CPL, LOGIN.COMI, or LOGIN.SAVE), and runs the program if it exists. These programs construct user environments by enabling global variable files and/or ABBREV files, setting terminal characteristics, and so forth.

4. The user is now logged in and ready to work on the system.

The only one of these steps which concerns us at this point is step 1, the validation of the user through the internal login program. Therefore, we will now look at that in greater detail. (For more information on external login programs, see Appendix B. For more information on users' login files, see Chapter 17 of the Prime User's Guide.)

User Validation at Login

The precise steps taken for validation depend on what information the user supplies in the command line. To illustrate this, this section will present several examples, showing how the system responds to different combinations of information. All examples will refer to the data base illustrated in Figure 15-1 and will assume that null passwords and the entry of passwords on the command line are allowed.

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

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

1. The system locates the user-id FROG in the system data base.
2. It verifies that GREEN is the password associated with FROG.
3. It checks the project data base for project SWAMP. It finds FROG listed as a member.
4. It checks project SWAMP's data base for FROG's origin directory. The directory is <SWAMP>LILYPAD. The system attaches FROG to that directory.
5. It checks the user data base for ACL groups for FROG. It finds one -- .AMPHIB — and marks FROG as a member of that group.
6. It checks project SWAMP's data base for project-specific ACL groups for FROG. It finds two — .FLYCATCHERS and .MUSICIANS. It adds those to .AMPHIB to create the list of ACL groups for FROG for this session.

Example 2: Only the word LOGIN given on the command line; use of an incorrect ID.

1. The system prompts for a user-id. The user responds: ORK.
2. The system checks the user data base. ORK is not there.
3. The system prompts for a password. ORK responds: FABLE.
4. The system terminates login and prints the error message, "Incorrect user-id or password."

   (Note that the system always prompts for a password if an incorrect user-id is given without one. This practice greatly increases login security.)

SYSTEM DATA BASE

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

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

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

PROJECT SWAMP DATA BASE

```
Default IAP:
Default ACL groups:
```

```
id: FROG
IAP:  <SWAMP>LILYPAD
ACL groups:
    .FLYCATCHERS
    .MUSICIANS
```

```
id:  POSSUM
IAP:  <SWAMP>TREE
ACL groups:  .POSSUMS
```

PROJECT HOLLYWOOD DATA BASE

```
Default IAP:  <HOLLYWOOD>MOVIES
Default ACL groups:  .STARS
```

```
id: FROG
IAP:
ACL groups:
```

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

Sample Portion of User Profile Data Base
Figure 15-1

Example 3: User-id alone given on the command line. The command is:

LOGIN POSSUM

1. The system finds the ID POSSUM in the system data base.

2. It checks for POSSUM's password, and finds that the password is null. Because the password is null, and because POSSUM is a valid user-id, the system does not prompt for the password.

3. The system checks to see whether POSSUM has a default project. He does — SWAMP. Therefore, the system does not prompt for project ID.

4. The system checks project SWAMP's data base. It finds POSSUM listed as a member.

5. Project SWAMP's data base reveals that POSSUM's origin directory is <SWAMP>TREE. The system attaches POSSUM there.

6. The system data base reveals no ACL groups for POSSUM.

7. Project SWAMP's data base shows that POSSUM is a member of the ACL group .POSSUMS. .POSSUMS becomes POSSUM's only ACL group for this session.

DATA SECURITY

Passwords

Rev. 19 systems allow the use of two kinds of passwords:

- File system object passwords.
- Login passwords.

The file system object password was previously the only type of password supported by PRIMOS. It can still be used to provide security for files and directories. However, the ACL system provides more efficient security more easily. Refer to the Prime User's Guide for information about protecting file system objects with ACLs.

Login passwords are new with Rev. 19 of PRIMOS. They are used only at login time. They do not relate to any file system object. Every user must have one.

Passwords are alphanumeric strings, up to 16 characters in length. They may contain any characters except PRIMOS reserved characters. The string may be null (consist of a blank entry). Each user-id should have a unique password associated with it. The login password should be known only to the user and the System Administrator. It should be changed at irregular intervals.

Access Control Lists

Access Control Lists (ACLs) are one of the cornerstones of the file system access control mechanism. Any users with Protect access can protect their files and directories with Access Control Lists.

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

An ACL can provide access control both on a per user basis and on a group basis. Both types of control can be combined in one ACL. An ACL can also control access rights for all users who do not appear in it by name or as group members. This control is specified via the $REST ACL entry.

When a user is included in an ACL both as a member of an ACL group and as a named user, the named user controls override the group rights. This can be used to either increase or decrease the rights of the named user.

Once you have defined a group in the system data base, any user can use that group name in an ACL.

Note

> Users may use non-defined group names in ACLs. The access control mechanism does not prohibit it. Adding the non-defined group does not achieve anything, since no members have been defined for the group and therefore noone belonging to the group can ever access the object protected by the ACL. The rest of ACL, however, remains valid and useful.

Since 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.

For a full discussion of ACLs in general, see Chapter 16 of the Prime User's Guide. For guidelines to setting system-level ACLs, see Chapter 5 of this book.

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 disk on the system. Priority ACLs override all other data security mechanisms.

They are generally intended for temporary use: for example, when you need to back up a disk. For a full discussion of priority ACLs, see Chapter 5, SYSTEM ACCESS.

COORDINATION OF SYSTEM AND DATA SECURITY

Since login security and data security can both be handled through the User Profile data base, the two can be coordinated easily. In particular, the use of projects, and of project-based ACL groups, often correlates with the degree of security desired on the system.

Essentially, there are three main types of system:

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

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

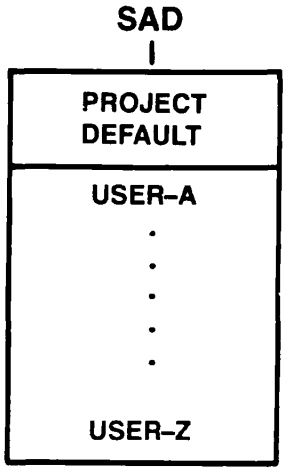
- A mixed system, which combines tight security on some projects (and for some users) with a friendly environment for other users. An example might be a college, where it would be desirable to give one set of users (the faculty) greater access and privilege than would be given to another set of users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

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

A system of this sort is diagrammed in Figure 15-2.

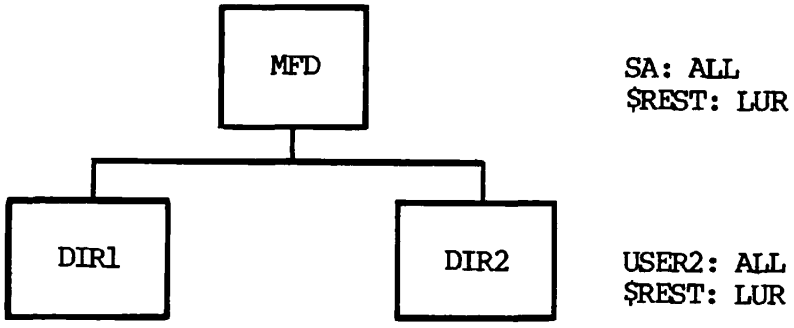
A Tightly Controlled System

A more tightly controlled system can be set up by using projects and project attributes in addition to the system-based attributes used in the friendly system shown above. When projects are used, every user can be required to log in, not only with a user-id and password, but with a project-id as well. 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.



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

Figure 15-2a



ACLs on Friendly System
Figure 15-2b

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

Figure 15-3 diagrams a sample project-based system. In this example, project DEFAULT is not in use.

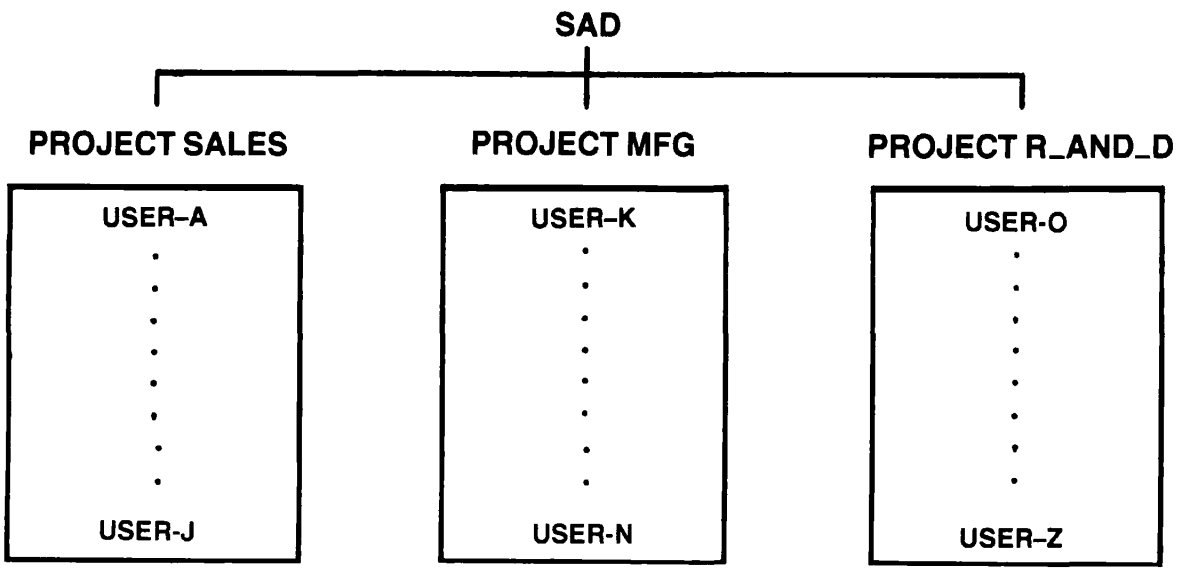
A Mixed System

A mixed system can also be constructed. This sort of system provides differing degrees of privilege for different users. It does this by setting up one level of security as the system default, then using specific projects to grant greater or lesser privileges to project members. This type of system can be used in two ways:

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

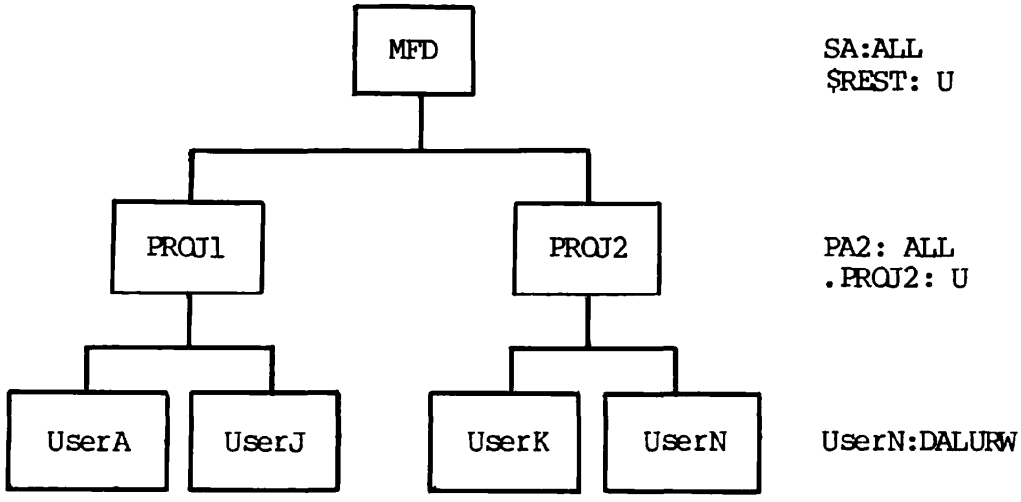
- All users belong to project DEFAULT. Some users also belong to other projects. ACLs give members of these other projects sole access to project-specific directories. Under this scheme, membership in project DEFAULT confers "standard rights". Other projects provide extra rights for their members.

Figure 15-4 diagrams a mixed system.

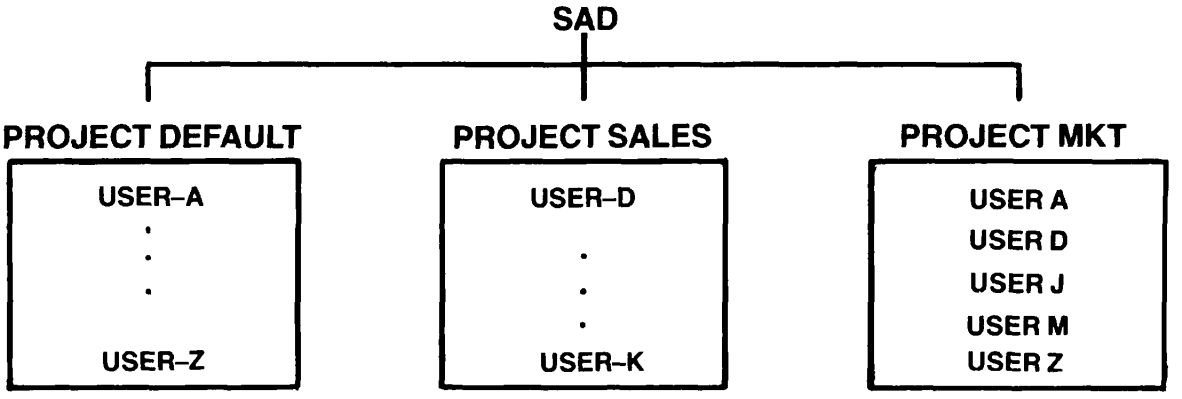


TIGHTLY CONTROLLED OR PROJECT-ORIENTED SYSTEM: Project DEFAULT does not exist. Every user must log in to a specific project. Project-based ACL groups can be used to deny non-project members access to project files and directories. Users may belong to multiple projects. Their access rights during any session will depend on the project-id they use at login time. Project Administrators may perform limited security functions.

Figure 15-3a



ACLs for Tightly Controlled System
Figure 15-3b



MIXED SYSTEM: In this example, all users belong to project DEFAULT. Users with special privileges belong to other projects, as well. System-based ACL groups tend to offer minimal access (such as LUR rights). Project-based ACL groups tend to offer wider rights.

Figure 15-4

16
Adding And Modifying System Software

INTRODUCTION

This chapter contains the following information on adding and modifying system software:

- Adding commands to the command directory (CMDNC0).
- Using customer-defined file suffixes.
- Changing defaults for translators.
- Listing of loader defaults.
- Adding new subroutines to the BASIC/VM compiler.
- Adding files to the HELP data base.
- Creating EVFU Files.
- Modifying the System and Network event loggers.

ADDING PROGRAMS TO CMDNC0

Run-time programs and CPL programs in the command UFD (CMDNC0) can be invoked by keying in the program name alone. This feature of PRIMOS is useful if a number of users invoke this program. Only one copy of the program need reside on the disk in UFD CMDNC0. Shared code is more space-efficient if more than one user will be executing the program.

In order to install new programs, you must have at least Add and Use rights to CMDNC0 (and, for V-mode programs, to UFD SEGRUN* or to the directory in which the segmented runfile resides).

> Caution
>
> When installing a new version of a command or a program, it is recommended that the operator save a copy of the old version in a convenient directory until such time as the new version is thoroughly checked out and it is determined that the old version is no longer needed.

Faster Program Startup

For best performance, the names of all runtime programs should end with the suffix .SAVE. All names of CPL programs must end with the suffix .CPL.

Users do not have to type these suffixes when invoking the commands. If a user types a command-name, the command processor looks in CMDNC0 for files in the order listed below.

```
command-name.RUN
command-name.SAVE
command-name.CPL
command-name
```

The R-mode loader, LOAD, adds the .SAVE suffix automatically when it creates a file. If you have existing run-time programs with names without the .SAVE suffix, you may want to change the names to include it.

> Caution
>
> The .RUN suffix is reserved for Prime. Do not use it for your programs. Do not change the .RUN suffix of any Prime-supplied file.

CPL Programs

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

```
COPY ANYTHING.CPL CMDNC0>ANYTHING.CPL
```

Any user may now invoke ANYTHING.CPL by typing ANYTHING.

R-mode Programs

Installation in the command UFD is extremely simple. The compiled and loaded runtime file is copied into UFD CMDNCO using the COPY command.

For example, assume you have written a utility program called FARLEY.FTN. This utility acts as a "tickler" for dates. Using FARLEY, each user builds a file with important dates. The FARLEY utility program, upon request, prints out upcoming events or occasions of interest to the user.

Notes

This utility does not exist; it is used as a plausible example.

R-mode programs can only be written for the FTN and RPG compilers and in assembly language, PMA.

First, compile and load the program:

```
OK, FTN FARLEY -64R                 Compile in 64R mode
0000 ERRORS [<.MAIN.>FTN-REV19.0]
OK, LOAD                            Invoke R-mode loader
[LOAD rev 19.0]
$ LO FARLEY                         Load object file
$ .                                 Load other modules, if needed
$ .                                 Load other libraries, if needed
$ LI                                Load system library
LOAD COMPLETE
$ SA                                Save memory image - FARLEY.SAVE
$ QU                                Return to PRIMOS
```

Then, copy the program into CMDNCO:

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

Any user can now invoke this program by typing FARLEY.

V-mode Segmented Runfiles

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

The procedure for creating an interlude is:

1. Create the desired SEG runfile.

2. Copy the SEG runfile into the directory in which it will actually reside. The directory SEGRUN* is provided for system software; however, other directories may be used if so desired.

3. Run the CPL program SEG>CMDSEG using RESUME; it will ask for the SEG runfile pathname. This CPL program will create the interlude program with the runfile base name and the .SAVE suffix. You must have at least Add, Delete, and Use rights (or be an owner) for the directory to which you are attached while executing SEG>CMDSEG.

Note

If you are going to work in the SEG directory, it would be advisable to delete protect the Prime-supplied files — CMDLIB.BIN, CMDSEG.CPL, and COMMAND_MAKE.SAVE — using the -PROTECT option of the SET_DELETE command. See the Prime User's Guide for details.

4. Copy the interlude program into CMDNCO.

Example

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

```
OK, FTN FARLEY -64V                 Compile in 64V mode
0000 ERRORS [<.MAIN.>FTN-REV19.0]
OK, SEG -LOAD                       Invoke SEG's loader
[SEG rev 19.0]
$ LO FARLEY                         Load object file
$ .                                 Load other modules, if needed
$ .                                 Load other libraries, if needed
$ LI                                Load system library
LOAD COMPLETE
$ SA                                Save runfile - FARLEY.SEG
$ QU                                Return to PRIMOS
```

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

```
OK, COPY FARLEY.SEG SEGRUN*>FARLEY.SEG
```

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

```
OK, R SEG>CMDSEG                          Run CPL program
[CMDSEG version 19.0]
Seg pathname: SEGRUN*>FARLEY.SEG          Enter runfile pathname
0000 ERRORS [<.DATA.>FIN-REV19.0]
[SEG rev 19.0]
#   vload ENK$RGWXFKVZBCKT.SEG.T
$   co abs 4000
$   mi
$   sz 4000
$   s/li share4 130000 4000 4000
$   xp 1 2
$   sy map 4000 126000
$   s/lo b_ENK$RGWXFKVZBCKR.FIN.T 0 4000 4000
$   au 3
$   d/lo seg>cmdlib.bin
$   au 0
$   d/li vapplb
$   au 1
$   d/li
LOAD COMPLETE
$   re
#   sh
TWO CHARACTER FILE ID: KT
CREATING KT4000
#   delete
#   q
```

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

```
OK, COPY FARLEY.SAVE CMDNC0>FARLEY.SAVE
```

When FARLEY is entered at the user terminal, the FARLEY interlude program in CMDNC0 is executed. This program executes the segmented runfile SEGRUN*>FARLEY.SEG.

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

Command Line Considerations

At Rev. 19.0, external commands (that is, commands in CMDNCO) are processed in three ways:

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

Note

CPL commands behave like NX$ commands. If wildcards and name generation are desired, they must be processed explicitly by the CPL program itself.

For a full explanation of command processing features, see the Prime User's Guide.

These considerations affect Systems Administrators who have added commands to CMDNCO:

1. If the added commands are not prefixed with NW$ or NX$, the commands will take advantage of full command processing functionality. The command processor will interpret the following command line options as wildcard options. (Abbreviations are shown following the long form of the option.)

```
-BEFORE date, -BF date
-AFTER date
-FILE
-DIRECTORY, -DIR
-SEGMENT_DIRECTORY, -SEGDIR
-ACCESS_CATEGORY, -ACAT
-VERIFY, -VFY
-NO_VERIFY, -NVFY
```

   It will also interpret the following options as treewalking options (abbreviations are shown):

```
-WALK_FROM n, -WLKFM n
-WALK_TO n, -WLKTO n
-BOTTOM_UP, -BOTUP
```

It will interpret all special characters as shown below:

```
Syntax suppressor   ~
Command separator   ;
Global variables    % %
Functions           [ ]
Iteration           ( )
Treewalking         @  @@  +  ^
Wildcarding         @  @@  +  ^
Name generation     =  ==  ^=  ^==  +
```

2. If you do not want your commands to utilize the above new features, or if a previously defined option conflicts with a wildcard or a treewalk option, you can rename the commands according to the rules given above. That is, the command name can begin with either NX$ or NW$. You can then write a short CPL interlude to accept the original invocation name and run the new command.

   Suppose such a command is FARLEY.SAVE. The command would be renamed NX$FARLEY.SAVE, and a simple CPL command FARLEY.CPL would be added to CMDNC0. FARLEY.CPL would be:

```
&ARGS ARGS: REST
NX$FARLEY.SAVE %ARGS%
```

   Then users of the FARLEY command can continue to invoke FARLEY, which now invokes FARLEY.CPL instead. FARLEY.CPL simply passes the unexpanded arguments on to NX$FARLEY.SAVE. Since the invocation of such commands will appear, to the user, to be the same as Prime-supplied commands, you must inform the users if processing of the command line is non-standard.

USER FILE SUFFIXES

All file suffixes beginning with the letter "U" are reserved for customer use. These suffixes may be used to create classes of user-defined files that can be processed by user-written programs and commands. All filenames must conform to Prime standards; see the Prime User's Guide for complete details on filenames. Some examples might be:

```
SALES.UDATA
UPDATE.UTRANS
PROBLEMS.UXB
```

CHANGING DEFAULTS FOR TRANSLATORS

Translators process source programs into object code, which can be loaded into a runtime memory image by one of Prime's loaders. The translators also perform other related operations such as error message printing and concordance generation. These operations are governed by command line keywords.

The procedure for changing compiler option defaults varies according to which translator is involved.

In Prime's newer translators — FORTRAN 77, Pascal, PL/I Subset G, and RPG II V-Mode — defaults are changed within a data file that is accessed by the execution of a "driver program".

In Prime's older translators -- FORTRAN, COBOL, RPG II, and PMA — defaults are changed by supplying new octal values for the A register and B register. These octal values set the default bits to on (1) or off (0) in the registers.

Note

When invoking the compilers of the older translators — FORTRAN, COBOL, RPG II, or PMA — the user can either give the desired options on the command line following the name of the source file, or supply new octal values for the A and B registers on the command line following the name of the source file. Both are allowed, but the option method is greatly preferred.

Defaults Set by Driver Programs

The FORTRAN 77, Pascal, PL/I Subset G, and RPG II V-Mode compilers have their option defaults set with driver programs. These programs are supplied in the UFDs F77>TOOLS, PASCAL>TOOLS, PL1G>TOOLS, and VRPG>TOOLS on the Master Disk. The default information is stored in data files supplied in UFD SYSOVL. The System Administrator can move these programs to other directories, if desired. The data files must remain in SYSOVL. In all cases:

- The directory in which the driver programs are stored should be password-protected and the driver programs set to allow non-owners NO rights. This prevents unauthorized execution to change defaults.
- The data file protection should be set to allow non-owners both Read and Write access. These files are in a non-ASCII format for security reasons. They are modified by the driver programs.

Changing the Defaults: To change the defaults, use the following sequence of commands:

ATTACH directory owner-password

RESUME translatorDF driver-pathname -options

directory	UFD in which the driver programs are resident. Supplied as TOOLS.
owner-password	Owner password of directory.
translator	Name of the compiler — F77 for FORTRAN 77, PASCAL for Pascal, PL1G for PL/I Subset G. (Exception — although the name of the RPG II V-Mode compiler is VRPG, the compiler name used here with DF is RPG.)
driver-pathname	Pathname of the data file, supplied in UFD SYSOVL. The files are F77DATA for FORTRAN 77, PASCALDATA for Pascal, PL1GDATA for PL/I Subset G, and RPGDATA for RPG II V-Mode.
options	New default options for the compiler.

Defaults Set by Driver Programs

The following lists the Prime-supplied compiler option defaults that are set by driver programs for FORTRAN 77, Pascal, PL/I Subset G, and RPG II V-Mode. Option names appear in uppercase and are preceded by hyphens. Options that cannot be explicitly invoked on the command line (defaults that have no official option names) appear in lowercase and are not preceded by hyphens. For a complete list of compiler options and related information for FORTRAN 77, Pascal, PL/I Subset G, and RPG II V-Mode, see the FORTRAN 77 Reference Guide, the Pascal Reference Guide, the PL/I Subset G Reference Guide, or the RPG II V-Mode Compiler Reference Guide, and their updates.

FORTRAN 77 (F77) Defaults

The Prime-supplied F77 compiler defaults are:

```
-64V
-BINARY YES
-DYNM
-ERRITY
-INIL
-LISTING NO
-LOGL
-NOBIG
-NODCLVAR
-NODEBUG
-NODOl
-NOERRLIST
-NOEXPLIST
-NOFRN
-NOOFFSET
-NOPRODUCTION
-NORANGE
-NOSILENT
-NOSTATISTICS
-OPTIMIZE
-OPT2
-UPCASE
```

Pascal (PASCAL) Defaults

The Prime-supplied PASCAL compiler defaults are:

```
-64V
-BINARY YES
-ERRITY
-LISTING NO
-MAP
nobig
nodebug
noexplist
noexternal
-NOFRN
nooffset
noproduction
-NORANGE
nosilent
nostandard
nostatistics
noxref
-OPTIMIZE
-UPCASE
```

PL/I Subset G (PL1G) Defaults

The Prime-supplied PL1G compiler defaults are:

```
-64V
-BINARY YES
-COPY
-ERRTTY
-LISTING NO
-MAP
nobig
nodebug
noexplist
nonesting
nooffset
noproduction
-NO_QUICK
norange
nosilent
nostatistics
noxref
-OPTIMIZE
-UPCASE
```

RPG II V-Mode (VRPG) Defaults

The Prime-supplied RPG II V-Mode compiler defaults are:

```
-BINARY YES
-ERRTTY
-LISTING NO
-NOBANNER
-NODEBUG
-NOERRLIST
-NOEXPLIST
-NOOBDATA
-NOOPTIMIZE
-NOSEQCHK
-NOXREF
-STATUS
```

Defaults Set as Register Values

In FORTRAN, COBOL, RPG II, and PMA, the defaults are changed by setting new values for the A and B registers.

Changing the Defaults: To change the defaults, use the following sequence of commands:

```
ATTACH CMDNCO owner-password

RESTOR translator

SAVE translator [3/A-register] [4/B-register]
```

translator	The translator utility: FTN, COBOL, RPG, or PMA.
A-register	New octal value of the A register. If omitted, the current value is unchanged.
B-register	New octal value of the B register. If omitted, the current value is unchanged.

FORTRAN (FTN) Defaults

The Prime-supplied FORTRAN compiler defaults are:

A register: '1707

```
Input file on disk (-INPUT pathname)
No listing file (-LISTING NO)
Binary file on disk (-BINARY YES)
Print error messages at user terminal (-ERRTTY)
Generate 32R mode code (-32R)
No global trace (-NOTRACE)
```

B register: 0

```
Static allocation of local variables (-SAVE)
Short integers (-INTS)
No concordances (-NOXREF)
Generate floating point skip instructions (-FP)
Do not generate code for segment-spanning arrays (-NOBIG)
Do not optimize DO loops (-STDOPT)
Do not generate code for debugger (-NODEBUG)
```

See the FORTRAN Reference Guide for values of the A and B registers.

COBOL (COBOL) Defaults

The Prime-supplied COBOL compiler defaults are:

A register: '2777

Input file on disk (-INPUT pathname)
Listing file on disk (-LISTING YES)
Binary file on disk (-BINARY YES)
Generate 64V mode code (-64V)
Suppress expanded listing (-NOEXPLIST)

See the COBOL Reference Guide for values of the A register; the B register is not used.

RPG II (RPG) Defaults

The Prime-supplied RPG II compiler defaults are:

A register: '3777

Input file on disk (-INPUT pathname)
Listing file on disk (-LISTING YES)
Binary file on disk (-BINARY YES)
Print error messages at terminal (-ERRTTY)
Print concordances (-XREF)
Print current status of compilation (-STATUS)
Suppress column index banner (-NOBANNER)
Suppress source program sequence checking (-NOSEQCHK)
Suppress octal listing of object data (-NOOBDATA)

See the RPG II Programmer's Guide for values of the A register; the B register is not used.

Note

The RPG II compiler interprets an A-register setting of 0 as '3777 (the default setting).

Assembler (PMA) Defaults

The Prime-supplied Assembler defaults are:

A register: '0777

Input file on disk (-INPUT pathname)
Listing file on disk (-LISTING YES)
Binary file on disk (-BINARY YES)
Print errors at terminal (-ERRLIST)
Suppress expanded listing (-NOEXPLIST)
Generate complete concordance (-XREFL)

See the Assembly Language Programmer's Guide for values of the A register; the B register is not used.

LOADERS

See the LOAD and SEG Reference Guide for complete details on Prime's loaders.

V-Mode (SEG)

Stack size: '6000 words
Default library: PFTNLB and IFTNLB (FORTRAN libraries)

R-Mode (LOAD)

Memory location: '122770 to '144000
Default library: FTNLIB (FORTRAN library)
Mode: D32R
Sector Zero Base Area:
  Base start at location '200
  Base range '600 words
COMMON: Top at location '077777

ADDING FORTRAN MODULES TO THE BASIC/VM COMPILER

FORTRAN routines written by programmers at your installation can be added to the BASIC/VM compiler so that anyone can use them. The following six-step procedure adds a routine to the necessary files.

1. Copy the routine's source file (for example, XYZ.FTN) to sub-directory BASICVSRC>SOURCE.

2. Edit BASICVSRC>SOURCE>FTNINIT.USER.FTN to include the routine name as follows:

   a. Insert the statement EXTERNAL XYZ after the statement EXTERNAL SAMPLE.

   b. After the statement:

      ```
      IF (NAMEQ$(NAME,LEN,'SAMPLE',6)) FTNINT = LOC(SAMPLE)
      ```

      insert the statement:

      ```
      IF (NAMEQ$(NAME,LEN,'XYZ',3)) FTNINT = LOC(XYZ)
      ```

      The numerical argument (here '3') is the length of the subroutine name (here 'XYZ') in characters.

3. In the file BASICVSRC>BASICV.BUILD.CPL, after the line:

   ```
   FTN SAMPLE -64V -XREFS -SPO -DCLVAR -PBECB -LIST NO
   ```

   insert the line:

   ```
   FTN XYZ -64V -XREFS -SPO -DCLVAR -PBECB -LIST NO
   ```

4. Run BASICVSRC>BASICV.BUILD.CPL with the RESUME command to create a new version of BASICV.

5. Run BASICV>BASICV.INSTALL.COMI with the COMINPUT command to install the new version of BASICV.

6. Run BASICV>BASICV.SHARE.COMI with the COMINPUT command to share the new version of BASICV.

Note

If you are loading subroutines from the system library VAPPLB, steps 1 and 3 are not necessary.

It is usually better not to perform steps 1 through 5 at the supervisor terminal, as they are very time consuming. Step 6 must be performed at the supervisor terminal. Regardless of the terminal being used, the user must have Write access to all the files.

For further information on calling subroutines from BASIC/VM programs, see the BASIC/VM Programmer's Guide and its change sheets for Revs. 18.1 and 19.0.

ADDING HELP FILES TO YOUR SYSTEM

The following information is provided for the System Administrator who desires to add files to the HELP data base supplied by Prime. (Please note that this information describes the files as they are at Rev. 19.0. HELP, and its file formats, may change in the future.)

The HELP Data Base

The HELP data base contains a collection of files called HELP files. HELP files are text files that contain information about a system facility, a command, or a subsystem. These files provide on-line information about these various system subjects.

Each HELP file is named after the facility, command, or subsystem with the addition of the suffix .HELP. For instance, the HELP file "SEG.HELP" contains information about the SEG command and subsystem.

Note

Prior to Rev. 19, HELP files did not use the .HELP suffix. Therefore, please delete all pre-Rev. 19 files from the HELP* directory before installing the Rev. 19 files. The old files without the suffix will not be read by the new HELP command, but they will occupy disk space uselessly.

The HELP* Directory

HELP files are kept in the HELP* directory. HELP* also contains two files with different functions. These files are:

- HELP_INDEX.HELP
- HELP_SEARCH_LIST

HELP_INDEX.HELP, as you receive it, contains a list of all the files in the HELP* directory at the time it was shipped to you. If you add files to the directory, you should update this list to include the

names of the new files. This file is displayed if the user types "HELP" or answers "YES" to the system prompt, "Can't find x, do you want a list?"

HELP_SEARCH_LIST contains command abbreviations. Its presence allows a user to type a standard (that is, system-defined) command abbreviation and view the HELP file for that command. For example, typing either "HELP ABBREV" or "HELP AB" displays the file "ABBREV.HELP".

Creating HELP Files

HELP files are standard ASCII files. They are created with the PRIMOS editor (ED).

There are two rules that must be observed when creating HELP files:

1. The first three lines of the file will not be displayed. You may want to make these comment lines, indicating who wrote the file and when it was written. Alternately, the first three lines may be left blank.

2. Any file to be read and displayed by the HELP command must have the .HELP suffix. (That is, the file created through the editor must be saved as "command.HELP".)

Adding Files to the HELP* Directory

You can add files to the HELP* directory at any time. The process is:

1. Create the new file with the editor

2. File it to HELP*>command.HELP

3. If you want the file to be shown in HELP_INDEX.HELP, you must edit that file to include the new HELP file name

Protecting the HELP Data Base

When your system is first installed, HELP* is accessible to anyone. We suggest that you limit Write access to this directory so that only authorized persons can alter the data base. This can be done by creating an ACL in which users authorized to alter the data base are given ALL access (either by name or as a group) and $REST is restricted to LUR access. Refer to the Prime User's Guide for more information about creating and using ACLs.

HOW TO CREATE EVFU FILES

EVFU files are text files created with the text editor, ED. They are graphic representations of the forms they are intended to format. Therefore, the number of lines in the EVFU file must equal the number of lines on the form.

The file assigns "channel" numbers to designated physical lines on the form. For example, if you wanted channel 5 to be on line 20, you would put the number 5 on the 20th line of the form.

EVFU files are constructed according to the following rules:

- Each line of the form may have only one channel associated with it.
- The channel number must be the first non-blank character on the line.
- The first line of the form is always represented by channel 1. Thus, a 1 must appear on line 1 of the EVFU file.
- A maximum of 12 channels may be designated.
- Only channel 12 may be assigned to more than one line.
- Lines without channels assigned to them may contain a 0, or may be left blank.
- The maximum form length for a 300 lpm printer/plotter is 132 lines.
- The maximum form length for a band printer is 143 lines.
- Comments may be entered on any line with a channel number.
- The EVFU file must reside in the SPOOLQ UFD.

Users making use of this EVFU format insert lines in their files containing "skip to channel" instructions. When the printer reaches that instruction, it skips to the line designated as that channel in the EVFU file.

When a printer environment has been set up to use an EVFU file, it will contain a copy of that file. Nonetheless, the original file should not be deleted. It contains the most convenient description of the contents of the EVFU. Moreover, should you ever modify the printer environment, the file must be reinterpreted by the phantom. For this reason, you may want to set specific ACLs, or one category ACL, on your EVFU files, to protect them from accidental deletion.

EVFU File Example

The following sample session creates an EVFU file. This file defines a format for a form titled PAYROLL. Note that as the form itself is 40 lines in length, the EVFU file has an identical length.

```
OK, ed payroll
INPUT
1


2


4


10


EDIT
fil
PAYROLL
```

SYSTEM EVENT LOGGING

First Level Event Logger (LOGEV1)

Information about an event is entered into the event buffer, LOGBUF, by LOGEV1, an internal PRIMOS subroutine. Each entry in the buffer contains the type and length of the entry and a number of data words passed to LOGEV1 by the routine programmed to record the event. (The exact format of event entries is described below.) When LOGBUF fills up, LOGEV1 discards subsequent entries and increments LOGOVF, a counter of the number of events lost.

LOGEV1 is called from the check handlers in SEG4 and DISKIO and from DOSSUB and TRWRAT.

Second Level Event Logger (LOGEV2)

The internal subroutine, LOGEV2, examines LOGBUF once a minute. LOGEV2 dumps any contents of LOGBUF to a disk file named LOGREC*>LOG.mm/dd/yy. LOGEV2 does not dump LOGBUF until the time has been set by the system operator. LOGEV2 cannot be called by the user, but users can monitor the output of LOGEV2 via the LOGPRT command.

19.0

LOGEV2 does not dump LOGBUF if the configuration directive LOGREC was set to a negative value (see Chapter 3). This allows operation with a write-protected disk.

MODIFYING THE SYSTEM EVENT LOGGING MECHANISM

The following paragraphs describe how to make modifications to the event logging mechanism. The only modules that must be changed are the file LOGPRT>LOGPRT.FTN and the PRIMOS module that will log the event.

Adding Event Types

To log a new event type, four actions are necessary:

19.0

1. An event message must be built that contains the event type, length of the message, and (optional) data words.

2. LOGEV1 must be called to enter the message into LOGBUF.

3. LOGPRT must be modified to recognize the new event type and appropriately format the data associated with the event.

4. LOGPRT must be rebuilt.

Building an Event Message: An event message must be built that contains the event type, length of the message, and (optional) data words.

19.0

An event message consists of a header word followed by data words. The header word consists of the event type in bits 1-8 and the total message length in bits 9-16. In PMA, a message could be defined by:

```
MSG   DATA   (5.LS.8)+3,DATA1,DATA2
```

The system event types currently defined for use with LOGPRT's -TYPE option are:

Type	Meaning
COLD	Cold starts
WARM	Warm starts
TIMDAT	Time/date entries (see Note)
CHECKS	Machine checks (including memory parity)
MCHECK	Machine checks (excluding memory parity)
DISKER	Disk errors
OVERFL	Record event logger overflow entries
SHUTDN	Operator shutdowns
CHK300	Prime 300 machine checks
PAR300	Prime 300 memory parity checks
MOD300	Prime 300 missing memory module checks
TYPE10...TYPE15	Entries for user-defined types 10 to 15
DSKNAM	ADDISK entries
POWERF	Power fail checks
SETTIM	SETIME command issued
QUIET	Quiet machine check mode
REMARK	Operator message
BADENT	Bad entries (not of the above types)

19.0

Note

The time/date stamps associated with the selected entries will not be processed unless TIMDAT is explicitly selected. For example: -TYPE DISKER TIMDAT will process all disk errors and their associated time/date stamps. If TIMDAT alone is specified, all time/date stamps will be processed. If TIMDAT is specified in conjunction with one or more other types, only the time/dates of the selected types will be processed. If the -TYPE option is not specified, all entries will be processed.

Calling LOGEV1 to enter the message into LOGBUF: Use the appropriate call procedure listed below.

In PMA, if the code is inside SEG4, the procedure is:

```
JSXB LOGEVL    /* Note the use of "LOGEVL" rather than "LOGEV1"
IP  MESSAGE
```

In PMA, if the code is outside SEG4, the procedure is:

```
CALL LOGEV1
AP  MESSAGE,SL
```

In FORTRAN, the procedure is:

```
CALL LOGEV1 (MESSAGE)
```

19.0

Modifying LOGPRT: LOGPRT must be modified to recognize the new event type and to appropriately format the data associated with the event. LOGEV1 and LOGEV2 do not examine the type field.

Currently, LOGPRT recognizes and formats data for the event types listed above. Types reserved for user definition are accepted, but result in a printout of only:

```
TYPE=type DATA=word-1 word-2 ...
```

The System Administrator defines the data and the data length. To add a new type, add a label to the array TYPLOG. At the new label call the STORE routine to perform the required formatting.

The calling sequence for STORE is as follows:

```
CALL STORE (text,txtlen,array,nw,dec)
```

The meaning of the parameters is as follows:

text — A text string to be printed.

txtlen — The length in characters of text. If zero, no text is printed.

array — An array of words to be translated and entered in the output line. ENTRY(1) is the first data word of the event message. ENTTYP and ENTLEN contain the type and length of the entry.

nw — The number of words in array. If zero, no words are translated.

dec — Octal/decimal flag. If zero, translation is to octal with no leading zero suppression. If nonzero, translation is to decimal with leading zeroes suppressed.

The total length of the text to be stored (txtlen+nw*7) should not exceed 67, the maximum length that can be printed at a terminal with an

indent in effect. (All lines after the first for an entry are indented 5 spaces.) If the length of text is too long, it will be split at a line boundary, subsequent lines being indented.

After formatting the entry, enter the statement:

```
GOTO 99000
```

Code at that label finishes the formatting and obtains the next entry from the input event logging file. 19.0

Rebuilding LOGPRT: To rebuild LOGPRT, run the command file LOGPRT>LOGPRT.BUILD.CPL. This will create a runfile called LOGPRT in the TOOLS UFD. This runfile can be moved to CMDNCO if desired.

Increasing the Size of LOGBUF

LOGBUF is defined in LOGEV1.PMA (supplied in PRIMOS>KS). The first entry in the buffer (label LOGBUF) is a COLD START entry followed by BSZ which defines the remaining size of LOGBUF (at least 63).

NETWORK EVENT LOGGING

First Level Event Logger (NETEV1)

Information about a network event is entered into an event buffer, NETBUF, by NETEV1, an internal PRIMOS subroutine. Each entry in the buffer contains the type and length of the entry and a number of data words passed to NETEV1 by the routine wishing to record the event. (The exact format of event entries is described below.) When NETBUF fills up, NETEV1 discards subsequent entries and increments NETOVF, a counter of the number of events lost.

NETEV1 is called from the communications device interface modules (dims), DOSSUB, and the X.25 network software.

Second Level Event Logger (NETEV2)

The internal subroutine NETEV2 examines NETBUF every 30 seconds and writes any contents of NETBUF to the disk file <0>PRIMENET*>NET_LOG.mm/dd/yy. NETEV2 will not dump NETBUF until the time has been set by the system operator. NETEV2 cannot be called by the user, but users can monitor its output via the LOGPRT -NET command. 19.0

Note

If possible, a warm start should be performed after a machine check halt. This allows NETEV2 to dump NETBUF, either after 30 seconds or on a SHUTDN ALL command.

MODIFYING THE NETWORK EVENT LOGGING MECHANISM

The following paragraphs describe how to modify the event logging mechanism. The only modules that must be changed are the file LOGPRT>LOGPRT.FTN and the PRIMOS module that logs the event.

Adding Event Types

To log a new event type, four actions are necessary:

1. An event message must be built that contains the event type, length of the message, and (optional) data words.
2. NETEV1 must be called to enter the message into NETBUF.
3. LOGPRT must be modified to recognize the new event type and appropriately format the data associated with the event.
4. LOGPRT must be rebuilt.

Building an Event Message: An event message must be built that contains the event type, length of the message, and (optional) data words.

An event message consists of a header word followed by data words. The header word consists of the event type in bits 1-8 and the total message length in bits 9-16. In PMA, a message could be defined by:

```
MSG   DATA   (5.LS.8)+3,DATA1,DATA2
```

This defines a message for event type 5; length of message (including header word) is 3 words.

The network event types currently defined are:

Type	Meaning
COLD	Cold starts
WARM	Warm starts
TIMDAT	Time/date entries
RESET	Circuit resets
BADSEQ	Packets out of sequence
OVERFL	Event logger overflow entries
SHUTDN	Operator shutdowns
LPE	Local procedure errors
RING1	Tokens inserted into the ring
RING2	Ring dims out of receive blocks
RING3	Ring nodes not accepting transmits
NETDMP	NETDMP calls
SMLC1	SMLC status errors
SMLC2	SMLC no STX preceding ETX
SMLC3	No system blocks for SMLC protocol messages
SMLC4	SMLC resets
HOSTDN	Level III protocol down start
POWERF	Power fail checks
INCREQ	Incoming call requests for FAM debug
OUCREQ	Outgoing call requests for FAM debug
REMARK	Operator remark
NPXTHR	NPX throttled on transmit or receive
NPXRCV	NPX got an unanticipated receive status
NPXCLR	Unexpected clearing cause on NPX master's circuit
NPXSEQ	NPX found sequence error in bounce detect
NPXCON	Unexpected circuit status, NPX call setup
BADENT	Bad entries (not of types listed above)

19.0

Calling NETEV1 to enter the message into NETBUF: Use the appropriate call procedure listed below.

To call NETEV1 in PMA, the procedure is:

```
CALL NETEV1
AP   MESSAGE,SL
```

To call NETEV1 in FORTRAN, the procedure is:

```
CALL NETEV1(MESSAGE)
```

19.0

Modifying LOGPRT: LOGPRT must be modified to recognize the new event type and appropriately format the data associated with the event. (NETEV1 and NETEV2 do not examine the type field.)

Currently, LOGPRT recognizes and formats data for the event types listed above. To add a new type, add a label to the array TYPNET. At

19.0 the new label (between $75000 and $98000), call the STORE routine to perform the required formatting.

The calling sequence for STORE is:

```
CALL STORE (text,txtlen,array,nw,dec)
```

text — A text string to be printed.

txtlen — The length in characters of text. If zero, no text is printed.

array — An array of words to be translated and entered in the output line. ENTRY(1) is the first data word of the event message. ENTTYP and ENTLEN contain the type and length of the entry.

nw — The number of words in array. If zero, no words are translated.

dec — Octal/decimal flag. If zero, translation is to octal with no leading zero suppression. If nonzero, translation is to decimal with leading zeroes suppressed.

The total length of the text to be stored (txtlen+nw*7) should not exceed 67, the maximum length that can be printed on a terminal with an indent in effect. (All lines after the first for an entry are indented 5 spaces.) If the length of text is too long, it will be split at a line boundary, subsequent lines being indented.

After formatting the entry, enter the statement:

```
GOTO 98500
```

Code at that label finishes the formatting and obtains the next entry
19.0 from the input network event logging file.

Rebuilding LOGPRT: To rebuild LOGPRT, run the command file LOGPRT>LOGPRT.BUILD.CPL. This will create a runfile called LOGPRT in the TOOLS directory. It may be moved to CMDNCO if desired.

Increasing the Size of NETBUF

NETBUF is defined in COMDEF. The first entry in the buffer (label NETBUF) is a one-word COLD START entry followed by a BSZ which defines
19.0 the remaining size of NETBUF (currently 63).

PART III
Networks

17
PRIMENET Overview

INTRODUCTION

This chapter and Chapter 18 contain information that you will need in order to administer PRIMENET on your system. This chapter summarizes PRIMENET's major services and briefly describes the various types of communications lines that you can configure. Chapter 18 contains specific instructions for administering PRIMENET.

Refer to the PRIMENET Guide for:

- Complete information on PRIMENET's services and how to use them.
- A technical discussion of communications lines.

PRIMENET SERVICES

PRIMENET provides five major services:

- Remote login services
- The NETLINK utility
- The Interprocess Communications Facility (IPCF)

- The File Transfer Service (FTS)
- Remote file access

These services allow users to access and process files that reside on remote systems, transfer files between systems, and log into remote systems. In many cases, PRIMENET makes communications lines completely transparent to users and applications. For example, a user can attach to directories on remote systems as though they were on the local system.

The first four services listed above are directly visible to the user. Remote file access is transparent to the user, and is implemented internally by the File Access Manager (FAM I or FAM II). FAM I and FAM II are primarily of concern to System Administrators and operators.

Note

Throughout this chapter and the next, the term node refers to a computer system that is a member of a network.

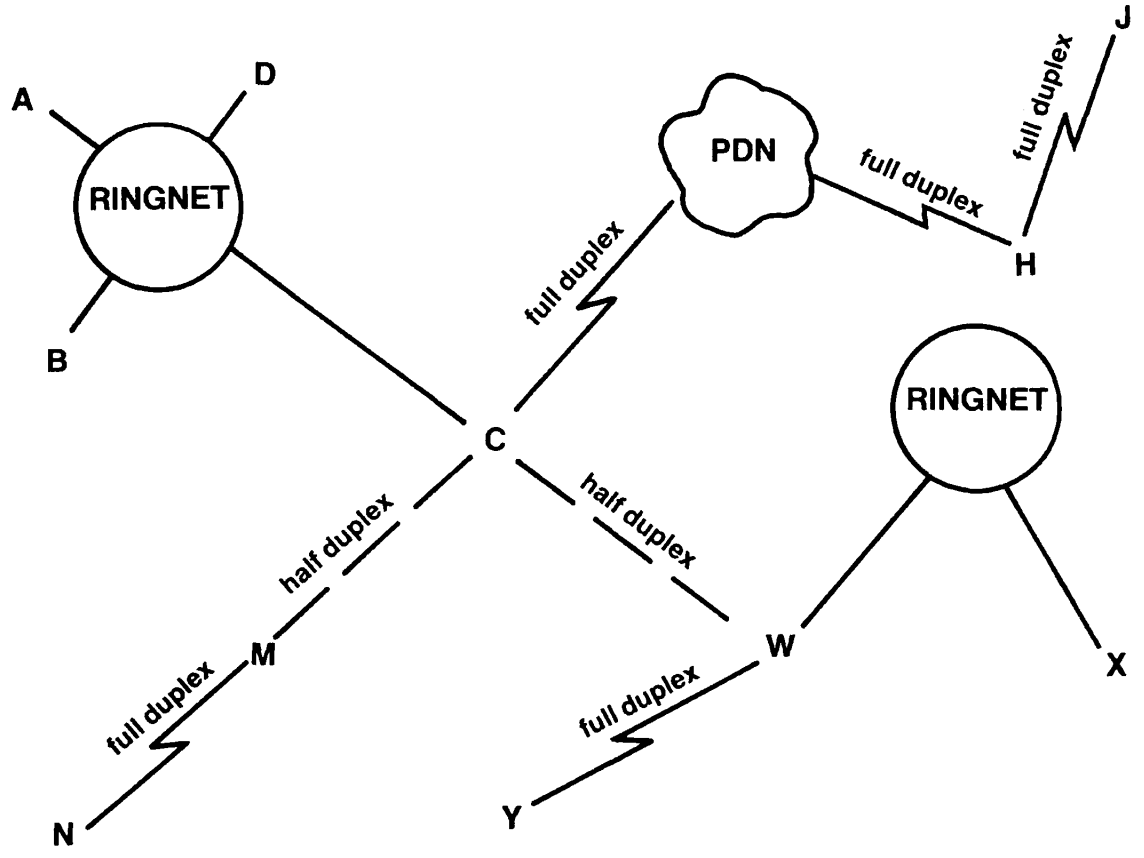
Remote Login Services

PRIMENET's remote login services allow you to log into a remote system directly (i.e., using only the LOGIN command). To perform a direct remote login, you must first log out of the system to which you are currently connected. You can then issue the LOGIN -ON command to log into the remote system. (Refer to the PRIMENET Guide for information on command format.) In order for this kind of remote login to work, the two systems involved must be directly connected. That is, one of the following must be true:

- The two systems are connected to the same PRIMENET ring (RINGNET).
- The two systems are connected by a full duplex synchronous line (possibly via a Public Data Network).
- The two systems are connected by a half duplex synchronous line.

(Rings and synchronous lines are discussed under TYPES OF PRIMENET COMMUNICATIONS LINES, later in this chapter.)

In the network illustrated in Figure 17-1, a user on System C could directly log into Systems A, B, D, M, W, and H -- but not into Systems X, Y, J, or N, because the connections between C and each of the latter systems are not direct. To log into System X, Y, J, or N, a user on System C would have to use the NETLINK utility, which is described below.



A PRIMENET Network
Figure 17-1

In all cases, a user must have a valid user-id on a system in order to log in there. Furthermore, the System Administrator on each system must enable direct remote login to other systems individually by means of the NETCFG utility. (Refer to the section called CONFIGURING THE NETWORK in Chapter 18.) The System Administrator on each system must also enable remote login from other systems by setting the CONFIG directive NRUSR to a positive number. (See Chapters 3 and 18 for information on CONFIG directives.) The issues of remote ids and network security are discussed in Chapter 18.

In some cases, you may want to implement external login programs for remote login. These programs, which can provide extra network security, are discussed in Appendix B.

The NETLINK Utility

The NETLINK utility allows you to be logged into several different systems at once. When you invoke NETLINK, you can log into any system that is directly connected (by ring or by synchronous line) to the system you logged into most recently. No logging out occurs.

Referring back to Figure 17-1, from SYSC you could use NETLINK to log into System M. You could then use it again from M to log into System N. At that point, you would be logged into Systems C, M, and N simultaneously, and would appear on the STATUS USER lists of all three systems. (This example assumes that you have valid user-ids on all three systems.)

Using NETLINK, you can be logged into as many as six different remote systems at a time. (These systems need not be in a chain, as Systems M and N are in Figure 17-1.)

NETLINK can be used to log into a remote system even if direct remote login to that system has not been enabled via the NETCFG utility.

In addition to its remote login services, NETLINK can be used to modify network parameters and to transfer files between network nodes. The file transfer feature is specifically designed for transfers between Prime and non-Prime systems. File transfers between two Prime systems should be accomplished by means of the COPY command, which uses PRIMENET's remote file access services, or the File Transfer Service (see below).

Refer to the PRIMENET Guide for detailed information on NETLINK.

The Interprocess Communications Facility (IPCF)

The Interprocess Communications Facility (IPCF) is a set of subroutines that are directly callable from user programs. With these subroutines, you can establish and pass data over virtual circuits that use the

industry-standard X.25 protocol. You can, for example, develop an application that consists of several different pieces, each piece running as a separate user on a different system, and have the different pieces exchange data across the network. The IPCF subroutines are designed for the Prime 50 Series CPUs in V-mode only.

The File Transfer Service (FTS)

The File Transfer Service (FTS) is a utility that transfers files between Prime systems in a network. One major advantage of FTS is that it can transfer files to or from a disk that is not on your system's device list. Furthermore, you can submit a file transfer request to FTS even if the node involved (or the line to it) is down. FTS holds the request in a queue until the transfer can be performed. You can direct FTS to create a log of the transfer and to notify the sender and/or receiver of the transfer. Full information on FTS is included in the PRIMENET Guide.

As System Administrator, you are responsible for several tasks regarding FTS. For example, you assign ACL rights to the FTS servers, monitor the size of FTS's FTSQ* directory, and use the FTGEN utility to configure the FTS system. These tasks are described in Chapter 18.

Remote File Access

PRIMENET's remote file access services allow users and application programs to access files on remote systems as though the files were on the local system. For example, remote files can be copied, edited, and otherwise manipulated exactly as if they were local. The user sees no difference between attaching to a directory on the local system and attaching to a directory on a remote node.

Remote file access is implemented internally by means of the File Access Managers (FAMs). The FAMs are subsystems that extend the functions of the PRIMOS file system to an entire network. Whenever a user issues a command that requires access to a remote system, a FAM intercedes and manages the necessary network functions.

There are currently two versions of FAM: FAM I and FAM II. FAM II is available at Rev. 18.2 and later Revs. At present, the two versions can co-exist on the same system. However, use of FAM II is strongly recommended, since it is much more efficient than FAM I. FAM I will eventually become obsolete.

Communication between two systems requires that the same version of FAM be running on both systems. In fact, each pair of systems in the network must agree to use either FAM I or FAM II in communicating with one another. Thus, SYSA and SYSB might use FAM I to communicate, while SYSA and SYSC use FAM II. In this case, both FAMs would have to be running on SYSA.

As System Administrator, you must confer with the System Administrator of each remote node on your network to decide which FAM will be used between your two nodes. When you run the NETCFG utility to configure your network, you are asked which FAM is to be enabled for each remote node. (Refer to the section called CONFIGURING THE NETWORK in Chapter 18.)

Note

You should configure FAM I only if your system is pre-Rev. 18.2 or if you need to communicate with a pre-Rev. 18.2 system.

FAM II: FAM II is actually not a single "manager", but rather a collection of "master" and "slave" processes. When a process (user or program) on your system requests access to a file on a remote device (for example, by attaching to a remote directory), your file system initiates a call to the remote system. The process on your system that requests remote access is called a master. On the remote system, the call is received by one of a number of slaves. A slave is a phantom that is dedicated to receiving calls from masters on other systems. Every system that uses FAM II has a certain number of slaves configured by means of the NSLUSR CONFIG directive. (As System Administrator, you must decide on the number of slaves to configure for your system. Refer to Chapter 3 and to the section called INCLUDING NETWORK-RELATED DIRECTIVES IN THE CONFIG FILE in Chapter 18.) The slave performs the requested operation on its system, and returns data to the master via subroutine parameters. All of this is transparent to the user.

A master may have at most one slave on a given remote system at one time. All of the master's operations on the remote system are handled by that one slave. However, a single master can have a slave on each of up to 15 remote systems simultaneously. Furthermore, many masters on one system can simultaneously call slaves on the same remote system.

A FAM II slave is dedicated to its calling master process while the master is attached to the remote system. Slaves that are not being used by masters remain "dormant", using minimal system resources, until they are called. A dormant slave has no user-id, and does not appear on the STATUS USERS list. However, a slave that has been called by a master takes on a user-id and becomes a full-fledged user, appearing on the STATUS USERS list.

The issue of which ids your system's slaves acquire, and how they acquire them, is of crucial importance for system security. In some cases, slaves log into the system and go through the standard user validation process. In other cases, a slave may log in in a non-standard fashion, and take on a user-id without going through user validation. This issue is discussed in detail in the section called MAINTAINING NETWORK SECURITY in Chapter 18. That section also discusses the related topic of node-node passwords (passwords that you can assign for use between masters and slaves).

If your system uses FAM II, you should note that the ADDISK, SHUTDN, and STATUS DISKS commands have slightly different formats and/or actions for disks accessed via FAM II (as opposed to FAM I). For details, refer to the System Operator's Guide.

Remote Disk Access Under FAM II: In order for a remote disk to be accessible to your system (and thus to the users on your system) under FAM II, the following must be true:

1. The disk must be started up on the remote system by means of the ADDISK command.

2. The disk must be added to the device list on your system by means of the ADDISK command with the -ON option.

ADDISK is discussed in the System Operator's Guide.

Note

The REMOTE command does not affect disks that are accessed under FAM II.

FAM I: If you use FAM I to communicate with one or more remote systems, FAM I exists as a single phantom process on your system. When a user or program on your system requests access to a remote file, your FAM I directs the request to the FAM I on the remote system. The remote FAM I submits the request to its system, and then receives the desired data and sends it back to your FAM I, which in turn sends it to the original user. Thus, each FAM I talks only to its own system and to the other FAM I phantoms on the network:

Local user <---> Local FAM I <----> Remote FAM I <----> Remote system

The single FAM I phantom on your system handles all incoming and outgoing file access requests to and from other systems (except those systems with which you use FAM II instead of FAM I). FAM I can communicate with 15 different remote nodes simultaneously, performing one operation on each node. However, FAM I cannot handle more than one operation on the same node at the same time. Requests for each particular node form a queue and are handled one at a time.

Remote Disk Access Under FAM I: In order for a remote disk to be accessible to your system (and thus to the users on your system) under FAM I, the following must occur:

1. The disk must be started up (by means of ADDISK) on the remote system. (ADDISK is discussed in the System Operator's Guide.)

2. An operator on your system must issue the ADDISK command (with the nodename argument) to add the disk to your system's device list. In order for this to work, one of the following must be true:

   - The System Administrator on the remote system answered YES to NETCFG's PERMIT REMOTE FAM TO START DISKS? prompt when configuring your node, or
   - An operator on the remote system issued the REMOTE PERMIT command to grant your system access to the disk. (The REMOTE command overrides the default permission/protection granted by the PERMIT REMOTE FAM TO START DISKS? prompt.)

   (NETCFG is discussed in Chapter 18.)

TYPES OF PRIMENET COMMUNICATIONS LINES

As System Administrator, you are responsible for configuring the network: informing your system of the characteristics of the various communications lines it will use. The tool that helps you configure the network is the NETCFG utility, which is described in Chapter 18. NETCFG prompts you for information about the nodes of your network and the communications lines that connect them to your system.

This section describes the types of lines you can configure through NETCFG. For a more technical discussion of communications lines, refer to the PRIMENET Guide.

Each remote system in your network is connected to your own system by one of the following:

- Ring network (RINGNET)
- Full duplex synchronous line (possibly via a PDN)
- Half duplex synchronous line

Whether you are setting up a network of your own or joining an existing network, your Prime field representative will help you decide which types of lines to use. Your decisions will be affected by factors such as cost, available hardware, the distance between nodes, expected frequency of network use, and (in the case of an existing network) the types of lines already in use in the network.

In almost all cases, PRIMENET's services do not distinguish between rings, full duplex lines, and half duplex lines. These differences exist at the hardware and protocol levels. Line type is transparent to users, except in the case of certain IPCF subroutines, which allow the user to specify line type when establishing a virtual circuit. (Refer to the PRIMENET Guide for more information.)

In addition to deciding on line types, you should be aware of whether your system or any of the nodes it will communicate with are connected to a Public Data Network (PDN). PDNs are discussed later in this section.

The PRIMENET Ring Network (RINGNET)

A PRIMENET ring network is a group of up to 15 computer systems that are connected by cable in a ring configuration. Each system is, in effect, directly connected to all other systems on the ring. Even if two systems are not adjacent, a message from one to the other does not actually have to pass through the CPU that is physically between them. All signals circulate around the ring in one direction to their assigned destinations. If one system is disabled for any reason, the rest of the ring is not affected.

On the hardware level, RINGNET is controlled by the Prime Node Controller (PNC). All of the systems in the ring must be in close physical proximity (with a maximum of 750 feet between each pair of active systems) because of the nature of the electrical connection. Thus, RINGNET is generally used within one building or building complex.

Each node on a ring must be assigned a ring node ID, which is a number from 1 to 247 that uniquely identifies the node. Ring node ids are assigned by means of the NETCFG utility, which is described in Chapter 18.

Full Duplex Synchronous Lines

PRIMENET's full duplex synchronous lines are usually dedicated, leased lines between two Prime systems or between a Prime system and a Public Data Network. Full duplex lines are permanent connections between the nodes they connect, and cannot be reallocated for purposes other than PRIMENET. Each full duplex synchronous line that you configure is associated either with a PDN or with one (and only one) Prime system.

PRIMENET's full duplex synchronous lines use High-level Data Link Control (HDLC) protocol, with either HDLC framing or Bisynchronous (BSC) framing. (Framing refers to the method used to separate packets of data that are transferred.) In general, HDLC framing is recommended because it conforms fully with industry-standard X.25 protocol. However, if your system is equipped with SMLC or MDLC boards, you will

have to use BSC rather than HDLC. Your Prime field representative can help you determine which kind of framing to use. You specify the framing type for each full duplex line when you run NETCFG (see Chapter 18).

Half Duplex Synchronous lines

PRIMENET's half duplex synchronous lines are usually temporary, dial-up connections that are made and broken by the system operator. Half duplex lines are often used over long distances or when line use is infrequent, because in these cases dedicated lines are more expensive. Half duplex lines can also be "unassigned" from PRIMENET by the system operator, so that they can be used by RJE, DPTX, or other communications software.

Since half duplex lines are temporary rather than dedicated, half duplex lines and nodes need not be in one-to-one correspondence. For example, you could configure two HDX lines and three HDX nodes. One line could be used at different times to reach two different nodes.

When you configure half duplex lines (through NETCFG), you can specify an incoming password and an outgoing password for each remote node. These passwords are used internally between two nodes whenever a half duplex connection is made. If Node A calls Node B over a half duplex line, then the outgoing password that Node A has specified for Node B must match the incoming password that Node B has specified for Node A.

The operator command NET and the user command HDXSTAT are used to monitor and control half duplex lines. Refer to the System Operator's Guide for more information on these commands.

Numbers of Lines and Nodes Allowed

The following list provides guidelines for the numbers of lines and nodes that you can configure in a standard PRIMENET network. For more information, consult your Prime field engineer.

- 15 nodes in all (including yourself).
- 15 nodes (including yourself) on a ring network.
- 14 half duplex nodes.
- 14 nodes that you connect to through a PDN. Note that this is the limit for nodes configured through the NETCFG utility (see Chapter 18). You need not configure a PDN node in order to communicate with it. You can connect to any node on your PDN, as long as you identify it by address to NETLINK.

- 8 synchronous lines. Up to 2 may be full duplex (Prime-to-Prime or Prime-to-PDN). Up to 8 may be half duplex. Only two synchronous lines may be active at a time.

- 15 physical connections in all. Count one physical connection for each ring node and one physical connection for each synchronous line (half or full duplex).

A specific node may be associated with several different lines. For example, you might have both full duplex and half duplex connections to another system. PRIMENET automatically uses the most efficient and least expensive line available. (The order of priority is as follows: ring, direct full duplex, half duplex, full duplex via a PDN.) If the first-choice line fails, PRIMENET automatically switches to the next most efficient type. The IPCF subroutines allow you to override PRIMENET's choice of active line type.

Multiplexing: When a user or program on one system accesses another system, a virtual circuit (logical connection) is created between the two systems. Virtual circuits do not correspond in one-to-one fashion with actual physical lines. Instead, in a process called multiplexing, many PRIMENET virtual circuits share one physical line. Multiplexing is transparent to users and applications.

Public Data Networks (PDNs)

Any node on a network may be connected to a Public Data Network (PDN). PDNs connect Prime systems with both Prime and non-Prime systems. Connections to PDNs are made over synchronous lines.

Each network node that has access to a PDN receives a PDN address. You need to know the PDN addresses (if any) of the nodes on your network when you configure the network. A PDN address consists of up to 14 digits. Not all PDNs use all 14 digits. For example, a TELENET address has the format 3110AAANNNNN, where AAA is the area code and NNNNN is the standardized DTE address of the node. Digits 13 and 14 are sometimes used as a sub-address. For example, if your system's TELENET address is 311040799920, your system receives calls sent to 31104079992007 (sub-address 07), 31104079992012 (sub-address 12), and so on.

Among the PDNs currently supported by PRIMENET are TELENET (U.S.A), TYMNET (U.S.A.), DATAPAC (Canada), TRANSPAC (France), EURONET (Europe), and IPSS (United Kingdom). Consult your Prime field representative if you wish to connect to a PDN that is not mentioned here.

18
Administering PRIMENET

INTRODUCTION

This chapter provides information on PRIMENET administration. The major PRIMENET-related tasks of the System Administrator are as follows:

- Setting network-related ACL rights
- Configuring the network by means of the NETCFG utility
- Including network-related directives in the CONFIG file
- Maintaining network security
- Starting the File Access Managers (FAM I and/or FAM II)
- Administering the File Transfer Service (FTS)

This chapter discusses each of these tasks. In addition, the last section of this chapter lists PRIMENET-related operational tasks that are usually performed by the system operator. At some installations, the System Administrator performs these tasks; at others, the System Administrator simply ensures that someone is responsible for them. Further information on operator tasks can be found in the System Operator's Guide.

SETTING NETWORK-RELATED ACL RIGHTS

This section outlines the ACL rights that you must assign on your system in order for PRIMENET to work correctly.

The Network Server Process (NETMAN)

At Rev. 19.0, network activity has been moved to a new network server process called NETMAN. NETMAN appears on the STATUS USERS list as an NSP (network server process). NETMAN does not have to be registered in the user validation file. However, NETMAN comes from the system pool of phantom processes, so your system must have at least one phantom configured exclusively for NETMAN's use. (Refer to the section called INCLUDING NETWORK-RELATED DIRECTIVES IN THE CONFIG FILE, later in this chapter.) NETMAN must be granted All access rights to the PRIMENET* directory, which is described below.

The PRIMENET* Directory

The top-level directory PRIMENET* must exist on your system disk (on the COMDEV pack) at system startup. This directory contains the files NETWORK_SERVER.COMI and NETMAN.SAVE, which are run at system startup as part of the initialization of NETMAN. NETMAN must have All access to PRIMENET*. If insufficient access is specified, the network is not set up. In this case, NETMAN logs out, and the message "Network Server logged out during network startup" is displayed at the supervisor terminal.

If your system uses FAM II, the FAM II slaves also use the PRIMENET* directory. The file SLAVE.COMI in PRIMENET* is a command input file that initiates the startup of the slaves on your system. (FAM I and FAM II are discussed in Chapter 17.)

The user SYSTEM must be assigned Use, List, and Read access to PRIMENET*.

The FAM Phantom

If your system uses FAM I to communicate with a remote node, then you must grant the phantom user FAM appropriate ACLs rights to files on your system that are to be accessed by users on the remote node. Similarly, on the remote node, FAM must be granted ACLs rights to the files your system will access.

On each system that uses FAM I, the FAM phantom also needs All access rights to the top-level directory FAM. User SYSTEM needs Use and Read rights to the FAM directory.

Note

The use of FAM I between systems that use ACLs is strongly discouraged. The reason is that once the FAM phantom is granted access to a directory, any remote user from a FAM I node can gain access to that directory.

In general, two nodes should use FAM I to communicate only if one or both nodes are pre-Rev. 18.2 systems.

The File Transfer Service (FTS)

If your system uses the File Transfer Service (FTS), you must ensure that user SYSTEM, the FTS servers, and all FTS users have appropriate access rights to the FTSQ* directory and to source and destination directories. FTS-related access rights are discussed in the section called ADMINISTERING THE FILE TRANSFER SERVICE (FTS), later in this chapter.

CONFIGURING THE NETWORK

Before you can bring your computer up as part of a network, you must provide it with information about that network. For example, your system must know how many other systems are in the network, what kinds of communications lines will be set up, which FAMs and passwords will be used, whether a Public Data Network is involved, and so on. This information makes up the network configuration.

If your system is or will be part of a network, you have been supplied with the PRIMOS external command NETCFG. NETCFG is a utility that helps you input all of the network configuration information that the system needs. NETCFG asks you questions about each communications line that is to be set up. Using your answers, the utility builds a network configuration file, which is a binary file that the system can use at cold start time to set up the network. NETCFG names the network configuration file NETCON.

NETCFG creates the NETCON file in the directory to which you are attached. By running NETCFG in different directories, or by renaming NETCON files, you could create many different network configuration files. For example, you might want to store various hypothetical configurations for future use. However, at cold start, the system always looks for and uses the file called NETCON in the directory CMDNC0. Thus, you must be sure that the correct network configuration file is in CMDNC0 (and is called NETCON) at cold start time.

Caution

If you use NETCFG to create a new NETCON file in a directory where a NETCON file already exists, NETCFG will overwrite the existing file without warning.

Note

To protect your network configuration data, including the node-node and half duplex passwords, you should specify ACL protection for the NETCON file in CMDNCO. User SYSTEM requires Read access. Any users who may create or review the configuration need Read and Write access. All other users should have no access.

In a directory that is not controlled by ACLs, NETCFG automatically sets NETCON's protection to no rights for non-owners, all rights for owners.

In addition to creating NETCON files, NETCFG allows you to review the network configuration in an existing NETCON file. More information and an example are provided later in this section.

Preparing to Run NETCFG

Before you use NETCFG to create your network configuration file, it is recommended that you read through the dialog and examples in this section. It is also strongly recommended that you read the section called MAINTAINING NETWORK SECURITY, later in this chapter. The discussion there may affect your answers to some of the NETCFG prompts (especially the FORCE USER VALIDATION? and NODE-NODE PASSWORD? prompts).

If you make a mistake while running NETCFG, you can always use CONTROL-P to exit and then start over again, overwriting the incorrect NETCON file.

For information on the numbers and types of lines that can be configured, refer to the section called TYPES OF PRIMENET COMMUNICATIONS LINES in Chapter 17.

Invoking NETCFG

To invoke NETCFG, enter the following command line:

```
NETCFG [-NOCHECK] [-PASSWORD] [-DSC]
```

-NOCHECK Suppresses checking of the number of nodes and/or connections. If you have a non-standard version of PRIMOS, this option allows you to configure more nodes than the standard Prime-supplied PRIMOS will support. In this case, you must allocate sufficient table space in the operating system, or fatal errors could occur at cold start. Consult your Prime field representative for more information.

-PASSWORD Invokes a special NETCFG dialog that allows Node-Node passwords to be changed. This option is discussed later in this section.

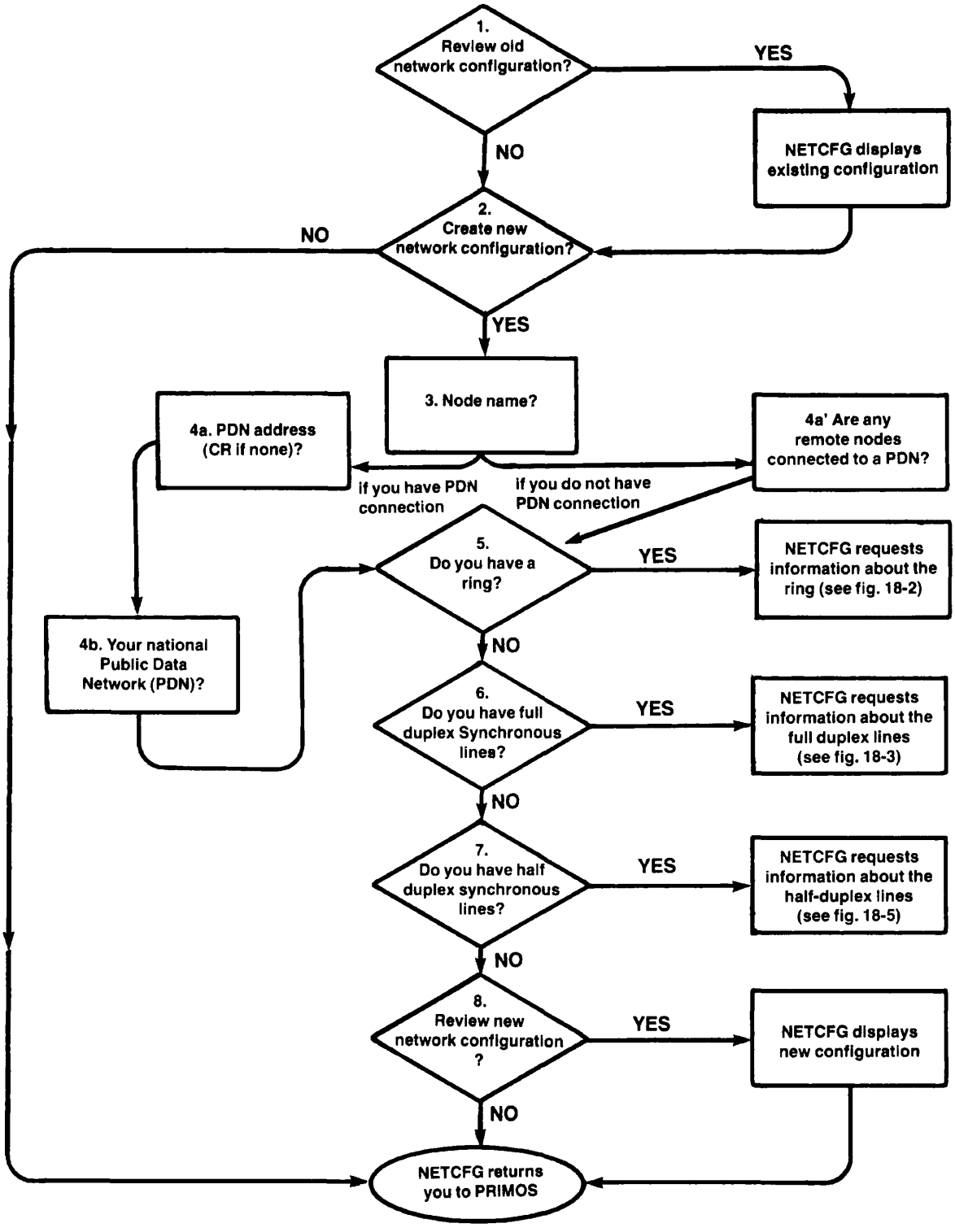
-DSC Allows specification of non-standard Data Set Control parameters for full duplex synchronous lines. This option is discussed later in this section.

Note

If you use the -NOCHECK option to configure more than 15 remote nodes, any nodes that use FAM I must be among the first 15 nodes configured. Remote access via FAM I to any node beyond the fifteenth will not work.

The Standard NETCFG Dialog

The following series of flowcharts illustrate the dialog that NETCFG produces when neither the -PASSWORD option nor the -DSC option is specified. The dialog is explained in the numbered Notes that accompany the flowcharts. Figure 18-1 shows the overall structure of the dialog. Figures 18-2 through 18-5 show the details of the sub-dialogs for ring networks, full duplex lines, and half duplex lines. Within each chart, NETCFG's prompts are numbered, allowing you to match each prompt in the chart with its explanation in the Notes.



The NETCFG Dialog
Figure 18-1

Notes on Figure 18-1

1. Review old network configuration?

   If you answer YES, NETCFG looks in the directory to which you are attached for a network configuration file called NETCON. If such a file does not exist, or if it is incompatible with the version of NETCFG you are using, NETCFG issues an error message and continues with the next prompt. If the file does exist and is compatible, NETCFG displays a summary of the network configuration information contained there. (Refer to Reviewing a Network Configuration, later in this section.)

   If you answer NO, NETCFG continues with the next prompt:

2. Create new network configuration?

   If you answer YES, NETCFG begins to prompt you for information about the network you want to set up. If you answer NO, NETCFG returns you to PRIMOS.

3. Node name?

   Provide the node name of your system. Node names are 1-6 characters long, with the same restrictions as PRIMOS filenames.

4a. PDN address (CR if none)?
   or
4a'. Are any remote nodes connected to a PDN?

   NETCFG asks question 4a if support for PDNs has been purchased for your system. Question 4a' is substituted if PDN support has not been purchased.

   In response to question 4a, enter your system's PDN address (if it has one) or a carriage return (if it does not). If you enter a PDN address, NETCFG continues with question 4b. If you enter a carriage return, NETCFG continues with question 5. (PDN addresses are discussed in the section called TYPES OF PRIMENET COMMUNICATIONS LINES in Chapter 17.)

   If you answer YES to question 4a', NETCFG includes prompts for the PDN addresses of remote nodes in the remaining dialog. If you answer NO to question 4a', prompts for PDN addresses are omitted from the remaining dialog.

4b. Your national Public Data Network (PDN)?

   Supply the name of your PDN. Currently, NETCFG accepts the following PDN names: TELENET, TYMNET, IPSS, DATAPAC, TRANSPAC. If you wish to configure a PDN not included in this list, consult your Prime field engineer.

5. Do you have a ring?

If you answer YES, NETCFG continues with question 5a. (See Figure 18-2.) If you answer NO, NETCFG continues with question 6.

6. Do you have full duplex synchronous lines?

   If you answer YES, NETCFG continues with question 6a. (See Figure 18-3.) If you answer NO, NETCFG continues with question 7.

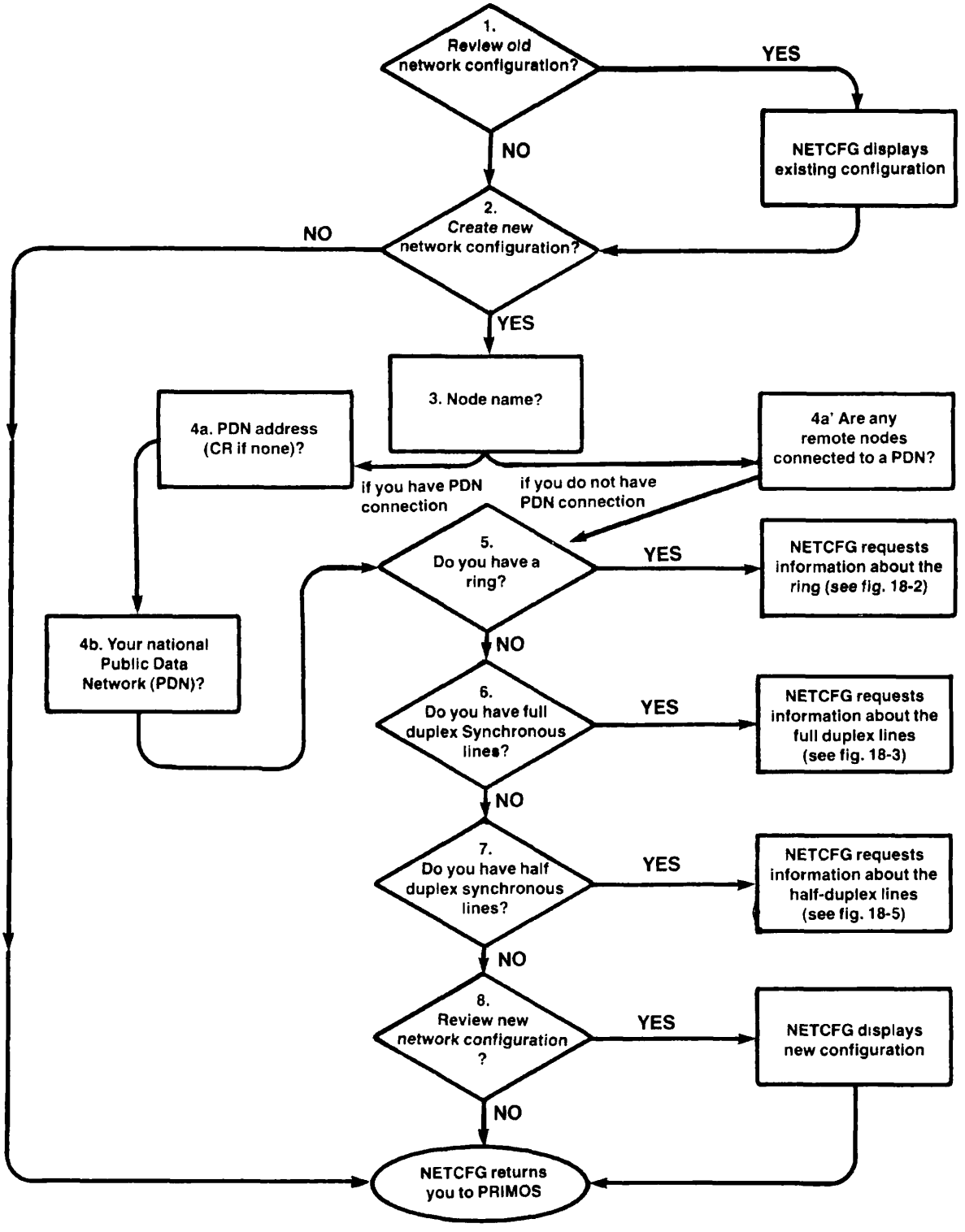
7. Do you have half duplex synchronous lines?

   If you answer YES, NETCFG continues with question 7a. (See Figure 18-5.) If you answer NO, NETCFG continues with question 8.

8. Review new network configuration?

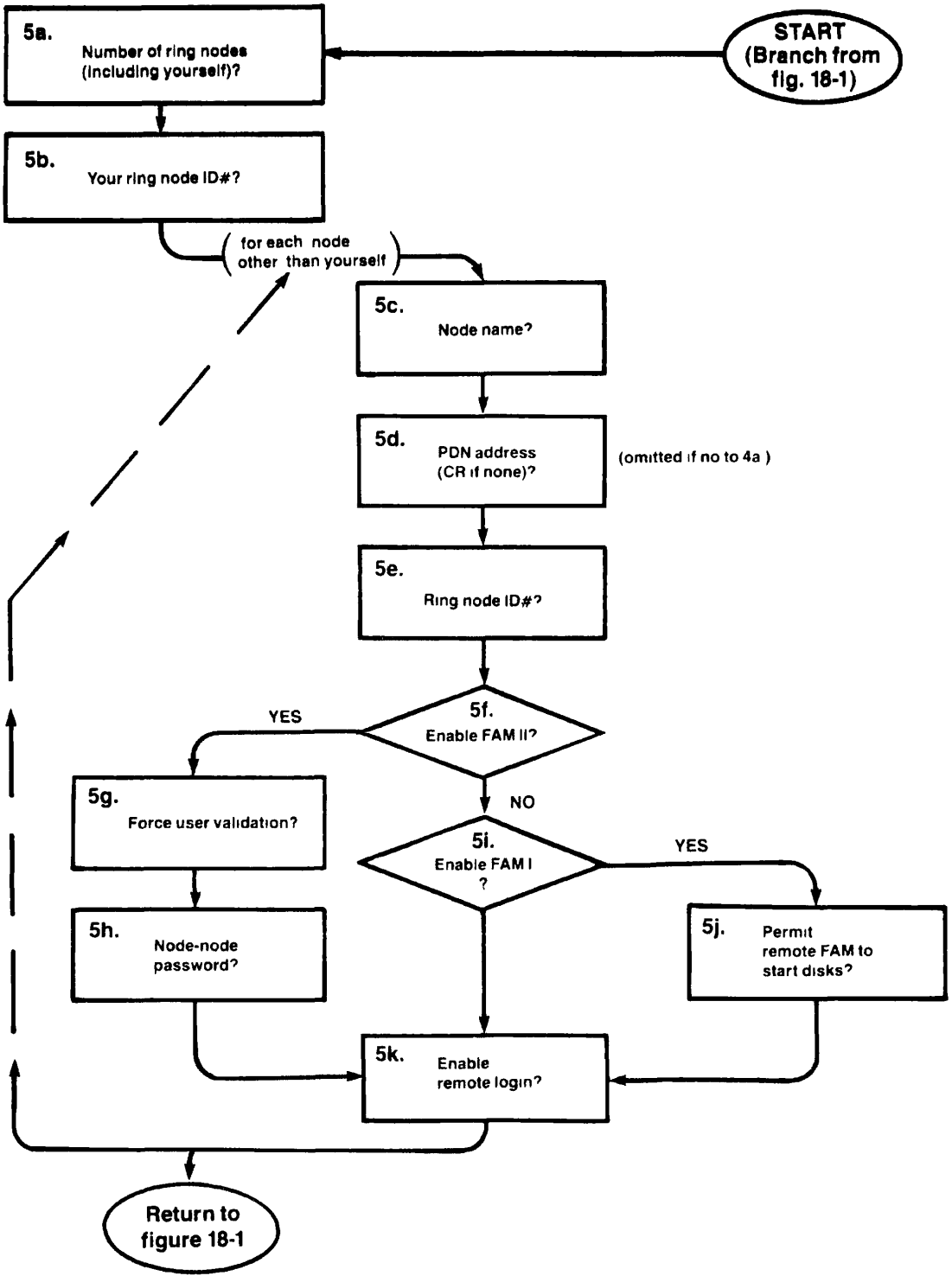
   If you answer YES, NETCFG displays a summary of the network configuration you have just specified, and then returns you to PRIMOS. (Refer to Reviewing a Network Configuration, later in this section.)

   If you answer NO, NETCFG returns you to PRIMOS.



The NETCFG Dialog
Figure 18-1

Repeated from page 18-6 for reference.



The Sub-dialog for Ring Networks
Figure 18-2

Notes on Figure 18-2

5a. Number of ring nodes (including yourself)?

Enter the number (0 to 15) of systems that make up your ring network.

5b. Your ring node ID #?

The ring node ID is a number between 1 and 247 which uniquely identifies the node. Your answer to this prompt assigns your system a ring node ID. Be sure that all of the systems on your ring agree on which node has which ID. It is recommended that you use the lowest possible numbers for IDs; this facilitates recovery from network errors on the ring.

5c. Node name?

Enter the name of the ring node you wish to configure next.

5d. PDN address (CR if none)?

Supply the PDN address of the node, or a carriage return if the node has no PDN address. (This prompt is omitted if you answered NO to question 4a'.)

5e. Ring node ID #?

Supply the ring node ID # (1 - 247) of the node.

5f. Enable FAM II?

Answer YES or NO. If you answer YES, the System Administrator of the remote node must also enable FAM II to your node. If you answer NO, questions 5g and 5h, which concern FAM II, are omitted from the dialog. For information on FAM II, see Chapter 17.

5g. Force user validation?

(This prompt appears only at Rev. 19.0 or later.) Answer NO if your system and the remote node coordinate user ids (that is, if user ids are unique across the two systems). Answer YES if your system and the remote node do not coordinate user ids (that is, if duplicate ids are allowed to exist across the two systems). For information on forced user validation, see the section called MAINTAINING NETWORK SECURITY, later in this chapter.

5h. Node-node password?

Supply a node-node password, or enter a carriage return if no password is to be used. The System Administrator of the remote node defined in 5c must specify the same password for your system. Node-node passwords are discussed in the section called

MAINTAINING NETWORK SECURITY, later in this chapter.

5i. Enable FAM I?

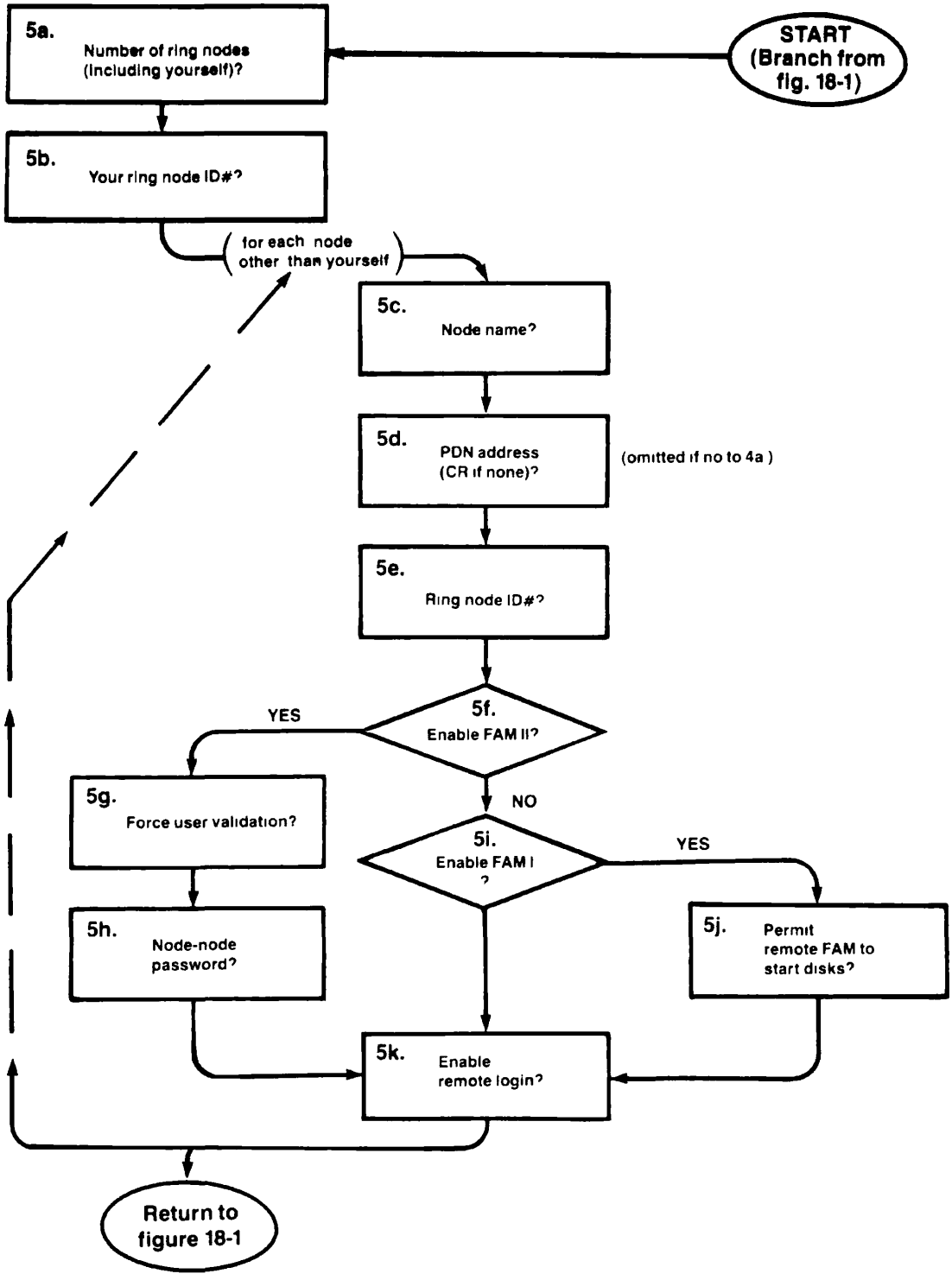
Answer YES or NO. For information on FAM I, refer to Chapter 17. If you answer YES, the System Administrator of the remote node must also enable FAM I to your system. (If you answer NO, question 5j is omitted from the dialog.)

5j. Permit remote FAM to start disks?

Answer YES to allow operators on the remote system to add your disks to the remote system's device list. (Once a disk of yours is added there, the remote FAM I can access it.) Answer NO to deny this permission. This option is not symmetric; for example, you can answer YES even if the remote System Administrator answered NO for your system. Your reply to this prompt can later be overridden for selected disks by means of the REMOTE command.

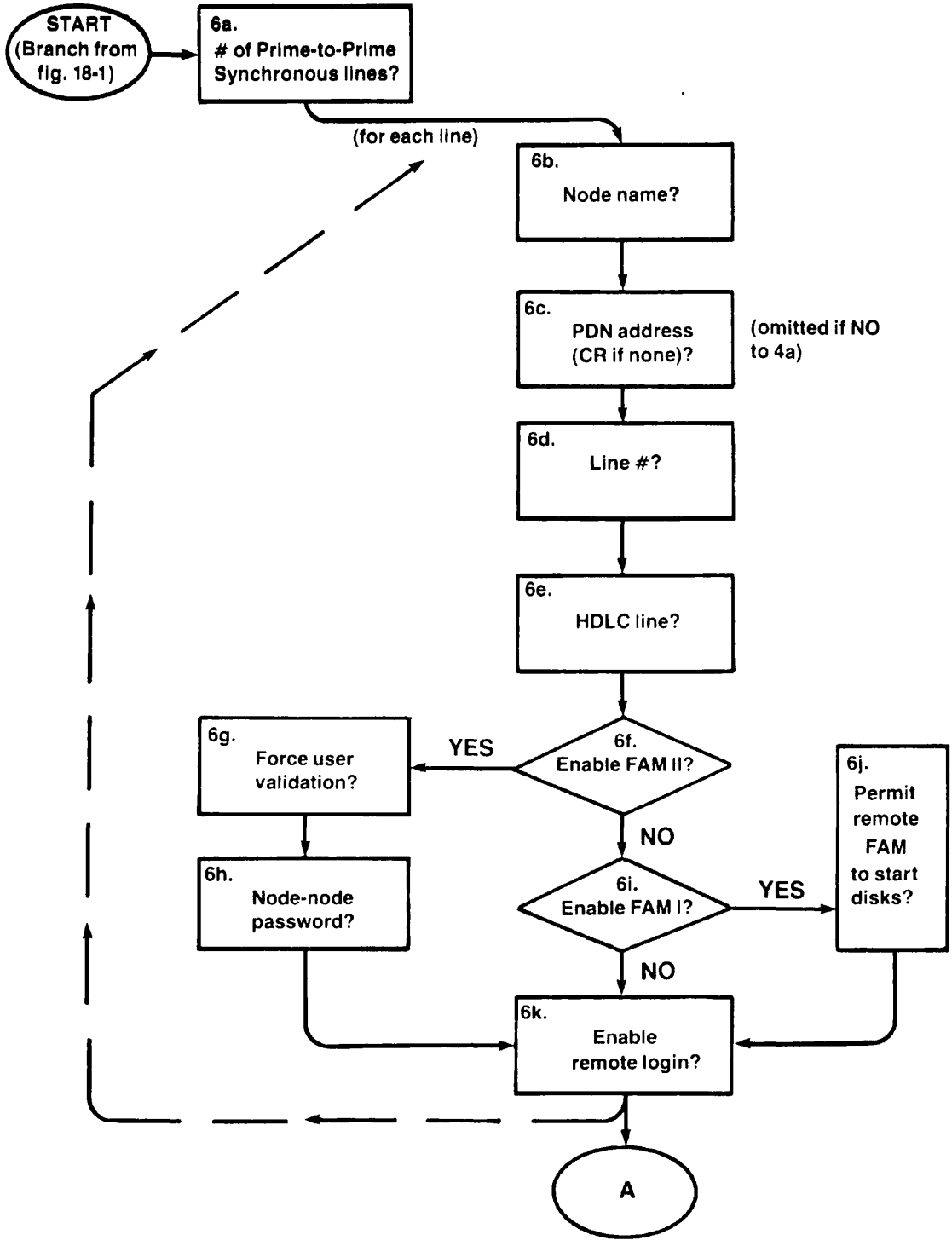
5k. Enable remote login?

Answer YES to allow users on your system to log into the remote system. Answer NO to deny this permission (except to users who perform remote logins via NETLINK). This option is not symmetric; for example, you can allow your users to log into the remote system even if users on the remote system are not allowed to log into your system.



The Sub-dialog for Ring Networks
Figure 18-2

Repeated from page 18-10 for reference.



The Sub-dialog for Full Duplex Synchronous Lines (Part 1)
Figure 18-3

Notes on Figure 18-3

6a. # of PRIME to PRIME Synchronous lines?

Enter 0, 1, or 2. If you enter 0, NETCFG skips to question 6l.

6b. Node name?

Enter the name of the node you wish to configure next.

6c. PDN address (CR if none)?

Supply the PDN address of the node, or a carriage return if the node has no PDN address. (This prompt is omitted if you answered NO to question 4a'.)

6d. Line #?

Supply the identifying line number (0 - 7) for this synchronous line.

6e. HDLC line?

Answer YES if this line uses HDLC framing, and NO if it uses Bisynchronous (BSC) framing. If you answer NO, NETCFG assumes that the line uses ASCII for the BSC framing character set. You can override this default by using the -DSC command line option. (Refer to The -DSC Option, later in this section.)

6f. Enable FAM II?

Answer YES or NO. If you answer YES, the System Administrator of the remote node must also enable FAM II to your node. If you answer NO, questions 6g and 6h, which concern FAM II, are omitted from the dialog. For information on FAM II, see Chapter 17.

6g. Force user validation?

(This prompt appears only at Rev. 19.0 or later.) Answer NO if your system and the remote node coordinate user ids (that is, if user ids are unique across the two systems). Answer YES if your system and the remote node do not coordinate user ids (that is, if duplicate ids are allowed to exist across the two systems). For information on forced user validation, see the section called MAINTAINING NETWORK SECURITY, later in this chapter.

6h. Node-node password?

Supply a node-node password, or enter a carriage return if no password is to be used. The System Administrator of the remote node defined in 6c must specify the same password for your system. Node-node passwords are discussed in the section called MAINTAINING NETWORK SECURITY, later in this chapter.

6i. Enable FAM I?

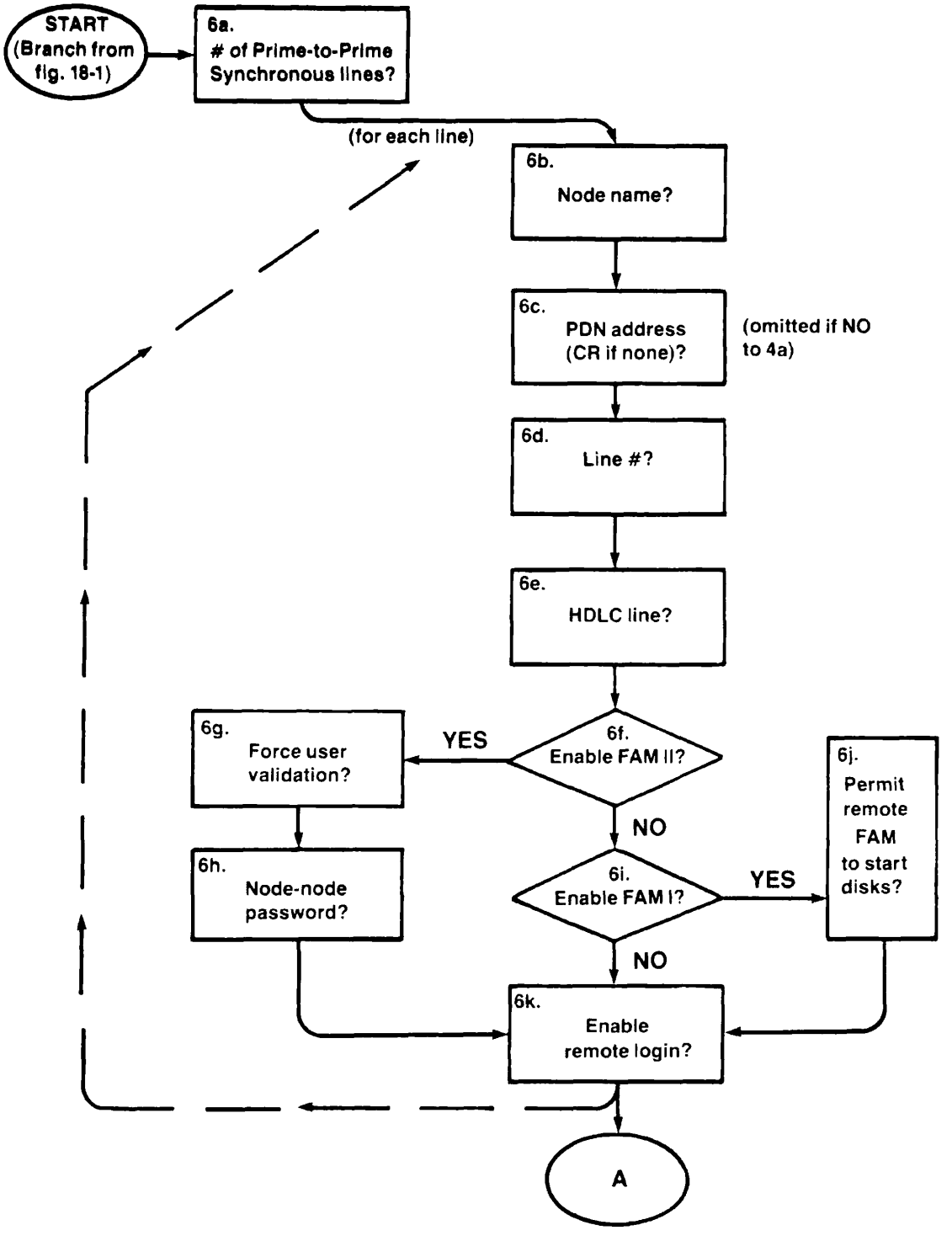
Answer YES or NO. For information on FAM I, refer to Chapter 17. If you answer YES, the System Administrator of the remote node must also enable FAM I to your system. (If you answer NO, question 6j is omitted from the dialog.)

6j. Permit remote FAM to start disks?

Answer YES to allow operators on the remote system to add your disks to the remote system's device list. (Once a disk of yours is added there, the remote FAM I can access it.) Answer NO to deny this permission. This option is not symmetric; for example, you can answer YES even if the remote System Administrator answered NO for your system. Your reply to this prompt can later be overridden for selected disks by means of the REMOTE command.

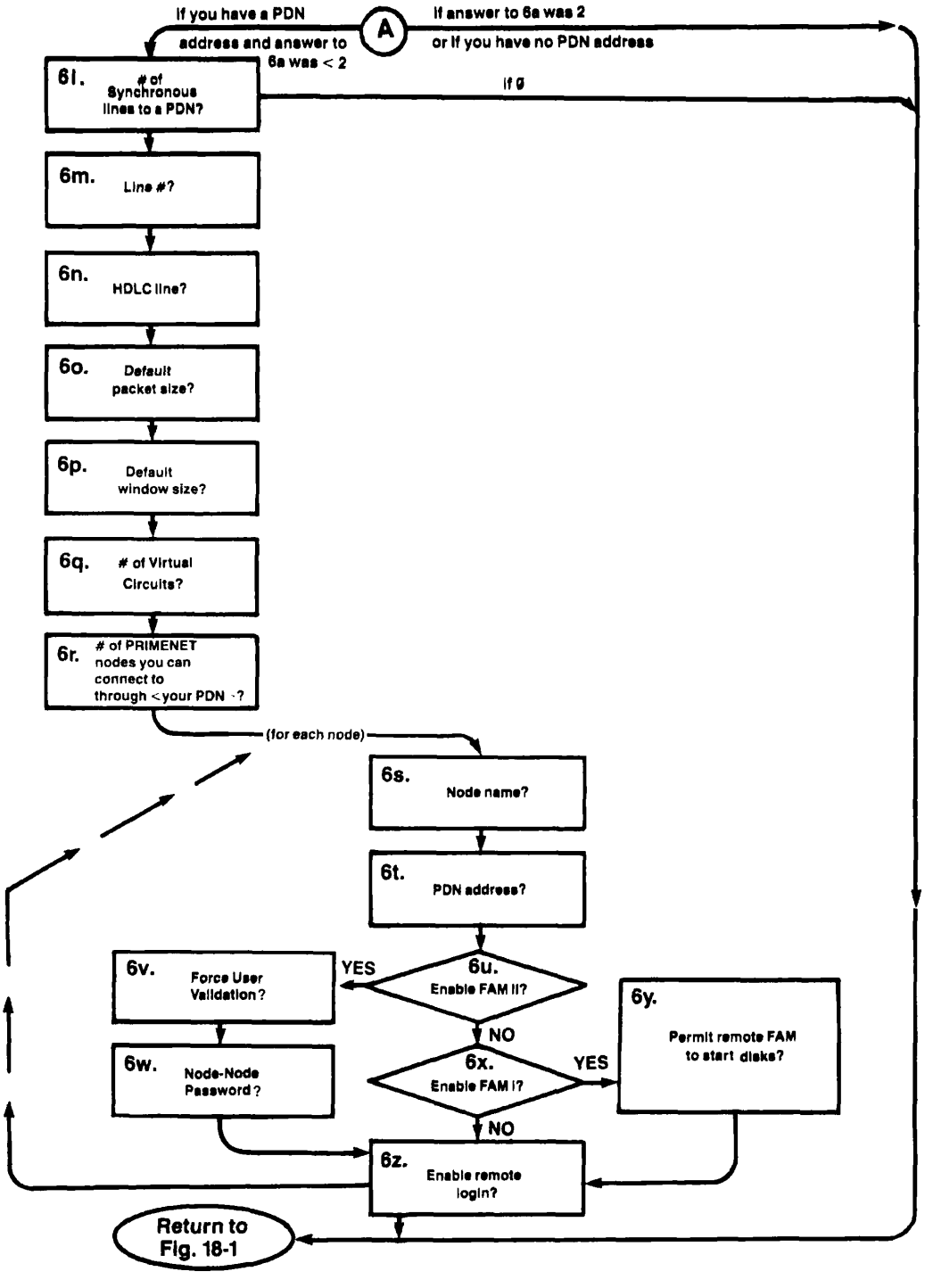
6k. Enable remote login?

Answer YES to allow users on your system to log into the remote system. Answer NO to deny this permission (except to users who perform remote logins via NETLINK). This option is not symmetric; for example, you can allow your users to log into the remote system even if users on the remote system are not allowed to log into your system.



The Sub-dialog for Full Duplex Sychnchronous Lines (Part 1)
Figure 18-3

Repeated from page 18-14 for reference.



The Sub-dialog for Full Duplex Synchronous Lines (Part 2)
Figure 18-4

Notes on Figure 18-4

Questions 61 - 6z appear only if you have specified a PDN address for your system (question 4a) and your answer to question 6a was 0 or 1.

61. # of Synchronous lines to a PDN?

    Answer 0 or 1. If you answer 0, NETCFG continues with question 7. (Refer to Figure 18-1.) If you answer 1, NETCFG continues with question 6m.

6m. Line #?

    Supply an identifying line number (0 - 7) for this synchronous line.

6n. HDLC line?

    Answer YES if this line uses HDLC framing, and NO if it uses Bisynchronous (BSC) framing. If you answer NO, NETCFG assumes that the line uses ASCII for the BSC framing character set. You can override this default by using the -DSC command line option. Refer to The -DSC Option, later in this section.)

6o. Default packet size?

    Supply the number (16 to 256) of bytes the PDN administration assumes are in a packet when no facilities are present in the call request packet. Consult your PDN representative for this information.

6p. Default window size?

    Supply the window size in number of packets (2 to 7), for calls with no facilities. Consult your PDN representative for this information.

6q. # of Virtual Circuits?

    Supply the maximum virtual circuit number (1 to 63) for this line, as agreed upon with the PDN administration. Virtual circuits are numbered beginning with 0. Thus, if the line has 32 virtual circuits, specify 31.

6r. # of PRIMENET nodes you can connect to through your PDN?

    NETCFG fills in the name of your PDN when it produces this prompt. Enter a number between 0 and 50.

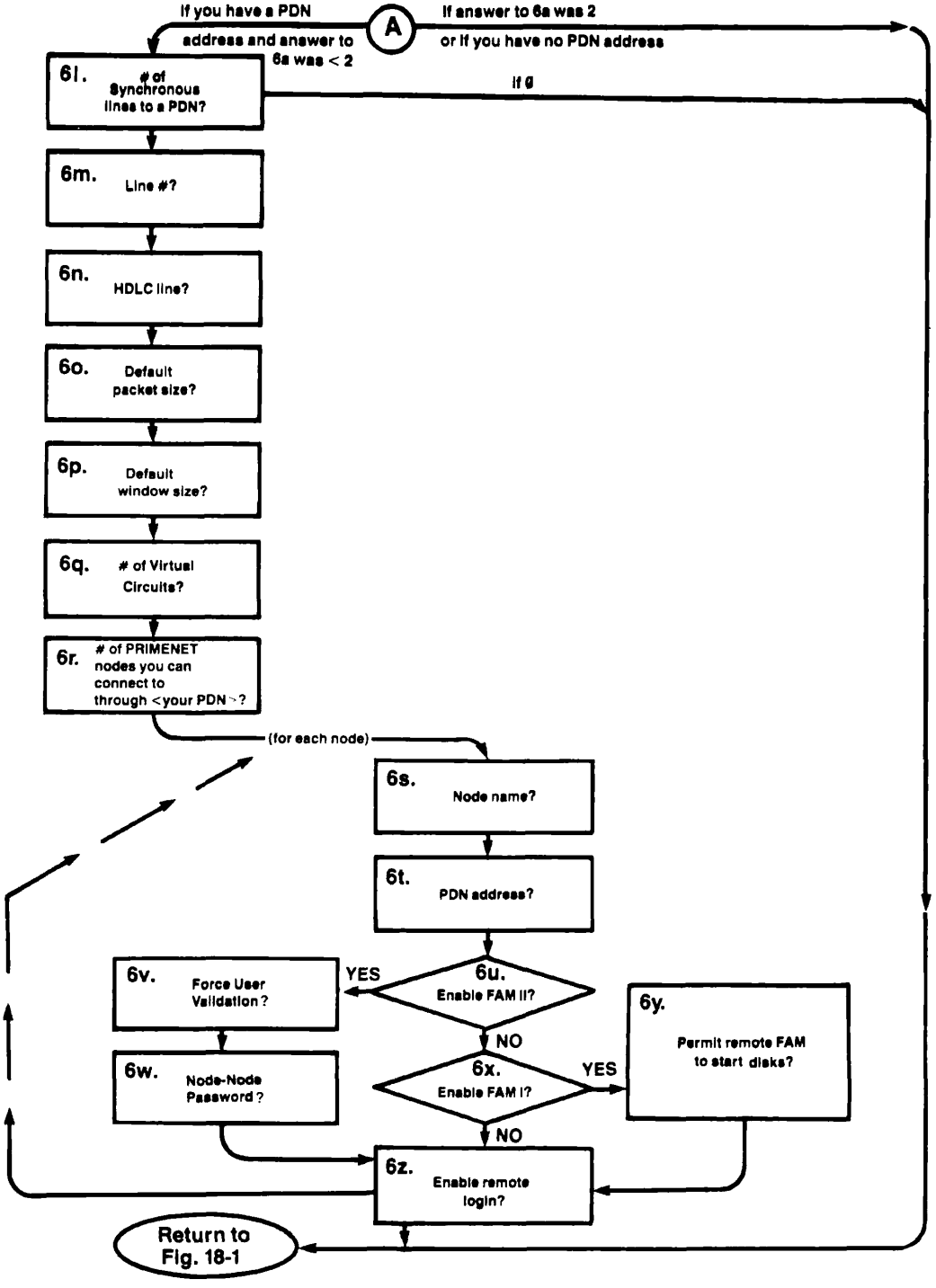
6s. Node name?

    Enter the name of the node you wish to configure next.

6t. PDN address?

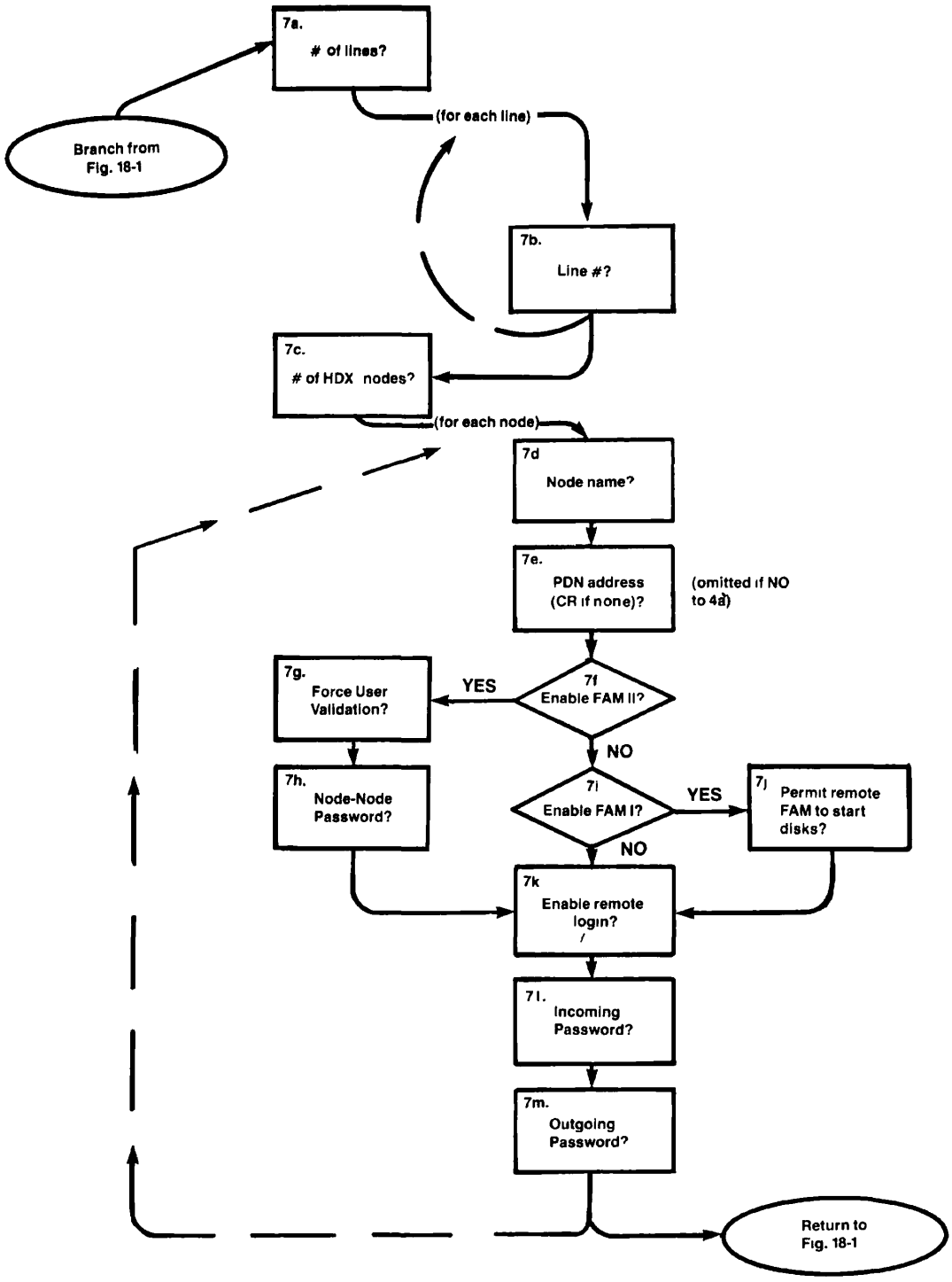
Enter the PDN address of this node. (The address is required, since you have specified that you connect to this node through the PDN.)

6u - 6z: Refer to questions 6f - 6k.



The Sub-dialog for Full Duplex Synchronous Lines (Part 2)
Figure 18-4

Repeated from page 18-18 for reference.



The Sub-dialog for Half Duplex Synchronous Lines
Figure 18-5

Notes on Figure 18-5

7a. # of lines?

Enter the number (0 to 8) of half duplex synchronous lines you wish to configure.

7b. Line #?

Enter an identifying line number (0 - 7). Do not duplicate the line numbers you have already specified for full duplex lines.

7c. # of HDX nodes?

Enter the number of HDX nodes you wish to configure.

7d. Node name?

Enter the name of the node you wish to configure next.

7e. PDN address (CR if none)?

Supply the PDN address of the node, or a carriage return if the node has no PDN address. (This prompt is omitted if you answered NO to question 4a'.)

7f. Enable FAM II?

Answer YES or NO. If you answer YES, the System Administrator of the remote node must also enable FAM II to your node. If you answer NO, questions 7g and 7h, which concern FAM II, are omitted from the dialog. For information on FAM II, see Chapter 17.

7g. Force user validation?

(This prompt appears only at Rev. 19.0 or later.) Answer NO if your system and the remote node coordinate user ids (that is, if user ids are unique across the two systems). Answer YES if your system and the remote node do not coordinate user ids (that is, if duplicate ids are allowed to exist across the two systems). For information on forced user validation, see the section called MAINTAINING NETWORK SECURITY, later in this chapter.

7h. Node-node password?

Supply a node-node password, or enter a carriage return if no password is to be used. The System Administrator of the remote node defined in 7c must specify the same password for your system. Node-node passwords are discussed in the section called MAINTAINING NETWORK SECURITY, later in this chapter.

7i. Enable FAM I?

Answer YES or NO. For information on FAM I, refer to Chapter 17. If you answer YES, the System Administrator of the remote node must also enable FAM I to your system. (If you answer NO, question 7j is omitted from the dialog.)

7j. Permit remote FAM to start disks?

Answer YES to allow operators on the remote system to add your disks to the remote system's device list. (Once a disk of yours is added there, the remote FAM I can access it.) Answer NO to deny this permission. This option is not symmetric; for example, you can answer YES even if the remote System Administrator answered NO for your system. Your reply to this prompt can later be overridden for selected disks by means of the REMOTE command.

7k. Enable remote login?

Answer YES to allow users on your system to log into the remote system. Answer NO to deny this permission (except to users who perform remote logins via NETLINK). This option is not symmetric; for example, you can allow your users to log into the remote system even if users on the remote system are not allowed to log into your system.

71. Incoming password?

Enter the incoming password, or enter a carriage return if no incoming password is to be used. For information on HDX passwords, see Chapter 17.

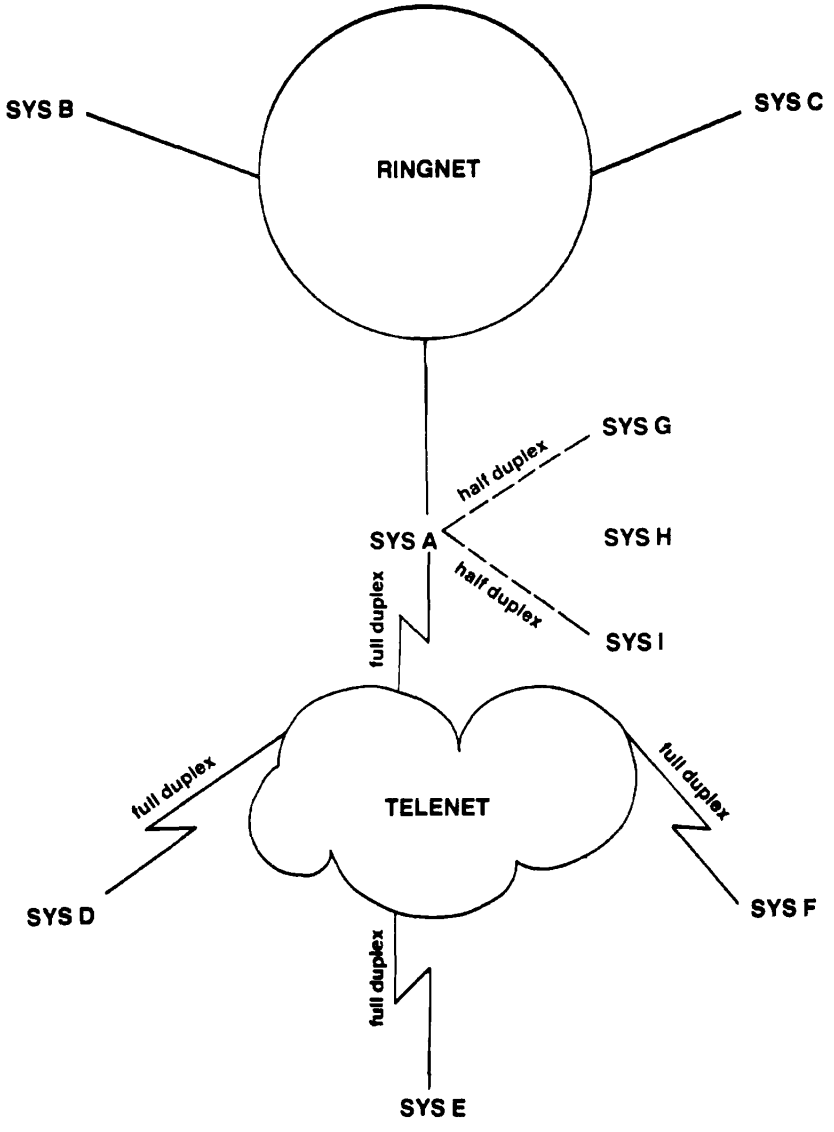
7m. Outgoing password?

Enter the outgoing password, or enter a carriage return if no outgoing password is to be used. For information on HDX passwords, see Chapter 17.

Example of Creating a Configuration

This section presents a sample NETCFG session in which a network configuration is created. The system on which NETCFG is being run, SYSA, is connected to a ring network with two other nodes (SYSB and SYSC). SYSA also has a full duplex connection to TELENET, through which it connects to SYSD, SYSE, and SYSF. Finally, SYSA has two half duplex (HDX) lines, which it uses to connect to three different HDX nodes (SYSG, SYSH, and SYSI).

Figure 18-6 is a diagram of the network created in the session.



Network Configured in Sample NETCFG Session
Figure 18-6

```
OK, netcfg
Review old network configuration? n

Create new network configuration? y

Please describe your node

Node name? sysa
PDN address (CR if none)? 311050812345
Your national Public Data Network (PDN)? telenet

Do you have a ring? y
Number of ring nodes (including yourself)? 3

Your ring node ID #? 1

Please describe the other nodes

Node name? sysb
PDN address (CR if none)?
Ring node ID #? 2
Enable FAM II? y
Force user validation? n
Node-Node password? abbey
Enable remote login? y

Node name? sysc
PDN address (CR if none)? 311050854321
Ring node ID #? 3
Enable FAM II? n
Enable FAM I? y
Permit remote FAM to start disks? y
Enable remote login? n

Do you have full duplex synchronous lines? y

# of PRIME to PRIME Synchronous lines? 0

# of Synchronous lines to a PDN? 1

Please describe each line

Line #? 1
HDLC line? y
Default packet size? 128
Default window size? 2
# of Virtual Circuits? 63
```

```
# of PRIMENET nodes you can connect to through  TELENET ? 3

Node name? sysd
PDN address? 311070644444
Enable FAM II? n
Enable FAM I? y
Permit remote FAM to start disks? y
Enable remote login? y

Node name? syse
PDN address? 311070666666
Enable FAM II? y
Force user validation? y
Node-Node password? yurt
Enable remote login? y

Node name? sysf
PDN address? 311061988888
Enable FAM II? y
Force user validation? y
Node-Node password? sammy
Enable remote login? y

Do you have half duplex synchronous lines? y

# of lines? 2

Please describe each line

Line #? 2

Line #? 3

# of HDX nodes? 3

Please describe each node

Node name? sysg
PDN address (CR if none)?
Enable FAM II? y
Force user validation? y
Node-Node password? elsie
Enable remote login? y
Incoming password? warp
Outgoing password? woof
```

```
Node name? sysh
PDN address (CR if none)?
Enable FAM II? y
Force user validation? y
Node-Node password? silver
Enable remote login? y
Incoming password? ho
Outgoing password? hum

Node name? sysi
PDN address (CR if none)?
Enable FAM II? n
Enable FAM I? y
Permit remote FAM to start disks? n
Enable remote login? y
Incoming password? sleepy
Outgoing password? dopey

Review new network configuration? n
OK,
```

Reviewing a Network Configuration

NETCFG organizes the information in the NETCON file into four parts. In reviewing the configuration, the utility produces up to four separate displays, depending on which kinds of lines you have configured:

Display Title	Display Contents
Ring Net	Information on the ring network.
Synchronous Net	Information on full duplex synchronous lines (Prime-to-Prime and Prime-to-PDN) and the PRIMENET nodes they connect.
HDX Net	Information on half duplex lines and nodes.
PDN Net	Information on the nodes you can connect to via your PDN.

After producing each display, NETCFG prints the prompt "—More?—" and waits for a carriage return before proceeding. If you type q or n, NETCFG returns you to PRIMOS.

For each node in the network, the following information is printed:

Label in Display	Meaning
Name	The node name.
Addr	The PDN address. If the node is not connected to a PDN, then this field is blank.
FAM INFO	This field may contain one of the following:

Entry	Meaning
II/VALID.	FAM II enabled with forced user validation.
II/NO-VALID	FAM II enabled, no forced user validation.

Entry	Meaning
I/RDP	FAM I enabled, disk start-up by remote FAM ok.
I/NO-RDP	FAM I enabled, no disk start-up by remote FAM.
None	Neither FAM enabled.

RLOG	Specifies whether or not remote login is enabled to the remote node.
Node-Node Password	If a node-node password is assigned (FAM II nodes only), it is displayed below the line with the node name.
Ring ID (Ring Net only)	Ring ID.
ME (Ring and Synchronous Nets only)	Identifies the local system.
PDN (Synchronous Nets only)	Identifies a Prime-to-PDN synchronous line. The PDN name appears in the "Name" column.
Line # (Synchronous and HDX Nets only)	In Synchronous Net display, line numbers appear in a column headed "Line #". In HDX Net display, line numbers are listed in top line.
Framing (Synchronous Nets only)	Framing type (HDLC, ASCII, or EBCDIC). The default for non-HDLC lines (i.e., lines that use Bisynchronous framing) is ASCII. This default can only be changed through the -DSC option. (Refer to The -DSC Option, later in this section.)

Virtual circuits, Window size, Packet size (Prime-to-PDN lines in Synchronous Net only)	Entry appears as follows: Max Vcn = k Window/Pkt = m / n where k is the maximum virtual circuit number, m is the default window size in packets, and n is the default packet size.
Incoming Password (HDX Net only)	Incoming HDX password, if any.
Outgoing Password (HDX Net only)	Outgoing HDX password, if any.

If you specified the -DSC option (see The -DSC Option, later in this section) in the NETCFG command line, the Synchronous Net display contains the following entry for each line:

Interrupts/Pattern/DS order = a/b/c

In this entry, a, b, and c have the following meanings:

Entry	Meaning
a	Indicates whether Interrupt on Data Set Status Change is to be enabled. (Value is "Yes" or "No".)
b	Value of the Data Set pattern expected. If a is "No", b is "None". If the network was configured without the -DSC option, or if the -DSC option was used but the defaults were not changed, then b is "Defaults".
c	Value of the Data Set order to be issued. If a is "No", c is "None". If the network was configured without the -DSC option, or if the -DSC option was used but the defaults were not changed, then c is "Defaults".

In the sample NETCFG session that follows, the configuration created in the previous example is reviewed.

```
OK, netcfg
Review old network configuration? y

Rev    19.0 network configuration file

Ring Net
       Name          Addr          Ring ID   FAM INFO   RLOG
       ------        ------------  -------   ---------  ----
*ME*   SYSA          311050812345        1
       SYSB                              2   II/NO-VALID Yes
                     Node-Node password: ABBEY
       SYSC          311050854321        3    I/RDP       No

--More?--

Synchronous Net
       Name          Addr          Line #    FAM INFO   RLOG   Framing
       ------        ------------  -------   ---------  ----   -------
*ME*   SYSA          311050812345
*PDN*  TELENET                          1                       HDLC
                     Max Vcn =    63  Window/Pkt =     2 /   128

--More?--

HDX Net   Lines:      2     3
       Name          Addr                    FAM INFO   RLOG
       ------        ------------            ---------  ----
       SYSG                                  II/VALID.   Yes
                     Incoming password: WARP
                     Outgoing password: WOOF
                     Node-Node password: ELSIE
       SYSH                                  II/VALID.   Yes
                     Incoming password: HO
                     Outgoing password: HUM
                     Node-Node password: SILVER
       SYSI                                    I/No-RDP  Yes
                     Incoming password: SLEEPY
                     Outgoing password: DOPEY

--More?--

PDN Net
       Name          Addr                    FAM INFO   RLOG
       ------        ------------            ---------  ----
       SYSD          311070644444              I/RDP     Yes
       SYSE          311070666666            II/VALID.   Yes
                     Node-Node password: YURT
       SYSF          311061988888            II/VALID.   Yes
                     Node-Node password: SAMMY

Create new network configuration? n
OK,
```

The -PASSWORD Option

If you include the -PASSWORD option on the NETCFG command line, NETCFG responds with an alternate dialog that lets you create, delete, or change the node-node password for any node in the configuration. The following is an example of the NETCFG -PASSWORD dialog:

```
OK, netcfg -password
Review old network configuration? y

Rev    19.0 network configuration file

Ring Net
       Name          Addr          Ring ID   FAM INFO   RLOG
      ------        ------         -------   --------   ----
*ME*  SYSA                            1
      SYSB                            2   II/VALID.    Yes
                  Node-Node password: SLEEPY
      SYSC                            3   II/VALID.    Yes
                  Node-Node password: HOLLOW

Do you want to update node passwords? y

Node name (CR if done)? sysb
New Node-Node password? core

Node name (CR if done)? sysc
New Node-Node password? calliope

Node name (CR if done)?

Review new network configuration? y

Rev    19.0 network configuration file

Ring Net
       Name          Addr          Ring ID   FAM INFO   RLOG
      ------        ------         -------   --------   ----
*ME*  SYSA                            1
      SYSB                            2   II/VALID.    Yes
                  Node-Node password: CORE
      SYSC                            3   II/VALID.    Yes
                  Node-Node password: CALLIOPE
OK,
```

The -DSC Option

The -DSC option is intended primarily for use by communications specialists. If you include the -DSC option on the NETCFG command line, the following questions are asked for each full duplex synchronous line that is configured. (These questions follow the ENABLE REMOTE LOGIN? prompt.)

- Framing character set?

  This question is asked only if you specified a non-HDLC line (a line that uses Bisynchronous framing) in question 6e or 6n of the NETCFG dialog. The reply must be either ASCII or EBCDIC.

- Enable DSS change interrupts?

  A reply of YES indicates that the CPU should be interrupted on Data Set Status changes. This is the normal mode of operation, corresponding to a Configuration word of '363.

  A replay of NO indicates that the CPU should not be interrupted on Data Set Status changes. This option, which corresponds to a configuration word of '323, is used when a line runs through a modem eliminator.

The following questions are only asked if Data Set Status changes are enabled:

- Dataset pattern expected?

  Enter the number (0-7) for the dataset leads, which must be high to transmit or receive. The bits that make up this number, xyz, have the following meanings:

  x — Carrier Detect bit. (Default is 0.)

  y — Clear To Send bit. (Default is 0.)

  z — Data Set Ready bit. (Default is 1, Data Set Ready.)

  If you enter a number greater than 7 for dataset pattern, only the last three bits of the number are used.

- Dataset order?

  Enter the number (0-3) for the dataset order to issue before transmitting. The bits that make up this number, yz, have the following meanings:

  y Request To Send bit. (Default is 0.)

  z Data Set Ready bit. (Default is 1, Data Set Ready.)

  If you enter a number greater than 3 for dataset order, only the last two bits of the number are used.

When you specify the -DSC option and ask to review a configuration, NETCFG includes status change interrupt, data set pattern, and data set order information in the Synchronous Net display, as discussed under Reviewing a Network Configuration, earlier in this section.

The following is a sample NETCFG session that uses the -DSC option.

```
OK, netcfg -dsc
Review old network configuration? y

Rev    19.0 network configuration file

Synchronous Net
       Name          Addr          Line #    FAM INFO  RLOG   Framing
       ------        ------------  ------    ---------- ---   -------
*ME*   HOME
       CHI                                1   II/VALID.   Yes    HDLC
                  Interrupts/Pattern/DS order = Yes/ Defaults
       ALB                                2   II/VALID.   Yes    ASCII
                  Interrupts/Pattern/DS order = Yes/ Defaults

Create new network configuration? y

Please describe your node

Node name? home
PDN address (CR if none)?

Do you have a ring? n

Do you have full duplex synchronous lines? y

# of PRIME to PRIME Synchronous lines? 2

Please describe each line

Node name? chi
```

```
PDN address (CR if none)?
Line #? 1
HDLC line? y
Enable FAM II? y
Force user validation? y
Node-Node password?
Enable remote login? y

Enable DSS change interrupts? y
Dataset pattern expected? 7
Dataset order? 3

Node name? alb
PDN address (CR if none)?
Line #? 2
HDLC line? n
Enable FAM II? y
Force user validation? y
Node-Node password?
Enable remote login? y

Framing character set? ebcdic
Enable DSS change interrupts? n

Do you have half duplex synchronous lines? n
Review new network configuration? y

Rev    19.0 network configuration file

Synchronous Net
       Name          Addr          Line #    FAM INFO  RLOG   Framing
      ------        ------        ------    --------- ----   -------
*ME*  HOME
      CHI                                1  II/VALID.   Yes     HDLC
                Interrupts/Pattern/DS order = Yes/0007/0003
      ALB                                2  II/VALID.   Yes   EBCDIC
                Interrupts/Pattern/DS order =  No/None/None
OK,
```

NETCFG Error Messages

The following are NETCFG's error messages and their meanings:

- BAD NETCON FILE

  You tried to review a NETCON file that is incompatible with the current revision of NETCFG. After issuing the message, NETCFG continues with the normal dialog.

- END OF FILE. NETCON

  You tried to review a network configuration when there was no information in the NETCON file. After issuing the message, NETCFG continues with the normal dialog.

- NOT FOUND. NETCON

  You tried to review a network configuration when there was no NETCON file in the current directory. After issuing the error message, NETCFG continues with the normal dialog.

- TOO MANY HOSTS

  You have configured more than 15 physical connections in all, and you have not specified the -NOCHECK option. After issuing the error message, NETCFG returns you to PRIMOS without creating the NETCON file.

- TOO MANY NAMES

  You have configured more than 15 network nodes (including yourself), and you have not specified the -NOCHECK option. After issuing the error message, NETCFG returns you to PRIMOS without creating the NETCON file.

INCLUDING NETWORK-RELATED DIRECTIVES IN THE CONFIG FILE

The network-related CONFIG file directives are listed below for easy reference. As System Administrator of a network node, you should ensure that these directives are included in the CONFIG file as necessary. For full information on the CONFIG file and how to use these directives, refer to Chapter 3.)

Directive	Meaning
NET ON	Specifies that the system is to be brought up as a network node, and that the NETCON file in the CMDNCO directory is to be used to configure the network. If NET ON is omitted from the CONFIG file, the NETCON file is ignored and the network is not set up.
NETREC	Enables/disables network event logging. (Event logging is discussed in the System Operator's Guide.)

NPUSR Specifies the maximum number of phantom users allowed on the system at one time. At least one phantom must be configured for exclusive use by NETMAN, the network server process. If FAM I is to be used on your system, at least two phantoms must be configured: one for NETMAN and one for the FAM.

NRUSR Specifies the maximum number of remote users allowed on the system at one time.

NSLUSR Specifies the maximum number of FAM II slaves that will reside on the system. This is the same as the number of simultaneous remote file accesses your system will support via FAM II. If you wish to receive messages from operators on remote systems via FAM II, you must configure at least one slave. You do not need this directive if your system uses only FAM I.

REMBUF Sets terminal input/output buffer sizes for remote users.

SMLC Enables and configures SMLC lines. Provides logical to physical line mapping for these lines. You need this directive only if your system has SMLC, HSSMLC, or MDLC boards.

MAINTAINING NETWORK SECURITY

As System Administrator on a network, you face three major security issues:

- You must control logins by remote users.
- You must control remote users' access to the disks attached to your system.
- You must control remote users' access to your system's files.

Before configuring your network, you should consider the degree of security required between your system and each remote system on your network. In some cases, security needs may be minimal; for example, users on two different systems may be members of a single department, and may need to access one another's data freely. In other cases, you may want to exert strict control over remote logins and remote data access. Most security measures can be applied to remote nodes on a system-by-system basis.

Controlling Logins by Remote Users

Remote users who wish to log into your system must have valid ids there. Your method of supplying ids to remote users should reflect the degree of security required between systems. For example:

- If security is not a strong concern, you might inform the remote system's System Administrator and/or users of some of the existing ids on your system.
- If security is a relatively strong concern, you might set up special ids (with limited ACL rights) for guests.
- If security is a very strong concern, you might decide that each remote user who wishes to log into your system must be assigned a unique id and User Profile.

Similarly, users on your system need ids on other systems in order to log in there. You may want to confer with the System Administrators of the other network nodes to decide how remote logins will be managed.

To enable users on your system to log into a remote node directly (without using NETLINK), you must answer YES to NETCFG's ENABLE REMOTE LOGIN? prompt when you configure that node. (Refer to the section called CONFIGURING THE NETWORK, earlier in this chapter.) Users on your system can log into the remote node via NETLINK even if you answered NO to the ENABLE REMOTE LOGIN? prompt.

To enable remote users to log into your system, you must set the NRUSR directive in your CONFIG file to a positive number. (See Chapter 3 and the previous section of this chapter for information on CONFIG directives.)

Controlling Remote Access to Your System's Disks

FAM II: If your system uses FAM II to communicate with a remote node, any disk that is started up on your system can be added to the remote node's device list by an operator there. (The REMOTE command does not affect disks accessed under FAM II.) Once added to the remote device list, the disk can be accessed on the remote system. Data is protected at the level of file or directory, by means of ACLs and/or passwords.

FAM I: Under FAM I, you must decide whether to grant remote systems permission to add your disks. If you answer YES to NETCFG's PERMIT REMOTE FAM TO START DISKS? prompt, an operator on the remote system may add any of your system's started disks to the remote system's device list. If you answer NO, that permission is denied. In either case, an operator on your system can override this default permission/protection by issuing the REMOTE command, specifying the remote node name and specific disk name(s). (For information on REMOTE, refer to the System Operator's Guide.)

Controlling Remote Access to Your System's Files: FAM II

To access your system's files under FAM II, a process on a remote node must call a slave on your system. When the slave is called, it logs into your system and takes on a user id and accompanying access rights. Thus, controlling remote access to your files boils down to controlling the ids and access rights that slaves can acquire when they log in. (Slaves are described in Chapter 17.)

Forced User Validation: As System Administrator, you exert control over slave logins by deciding whether or not to force user validation on the remote nodes you access via FAM II. A separate decision must be made for each node. This section describes forced user validation and explains when it should be used.

Note

This discussion assumes that your system and the remote node in question are both Rev. 19.0 systems. If your system and/or the remote node is a Rev. 18 system, refer to the section called Rev. 18 Issues, below.

Your decisions with regard to forced user validation should be made before you configure the network. When you configure a remote node for access under FAM II by means of the NETCFG utility, you must answer either YES or NO to NETCFG's FORCE USER VALIDATION? prompt. (For information on NETCFG, refer to the section called CONFIGURING THE NETWORK, earlier in this chapter.)

If you answer YES to the FORCE USER VALIDATION? prompt, then:

- A user on the remote node who wishes to access a file or directory on your system must first issue the ADD_REMOTE_ID (ARID) command on the remote system. On the ARID command line, the user must specify a user id that is valid on your system. (A password and project name may also be specified.) This id is used by any slave(s) subsequently called on your system by that user. (For information on the ADD_REMOTE_ID command, refer to the Prime User's Guide).

- When a slave is called from the remote node, a complete, standard login occurs. The slave's id is checked for validity on your system. If the id is valid, the slave takes on all ACL rights assigned to the id. If the id is not valid, then the master receives an error message, the slave does not log in, and no remote access occurs.

Note

If a remote node forces user validation on your system, then any user on your system who wishes to attach to the remote node must issue an ARID command first. As System Administrator, you might suggest that users who access remote files frequently should incorporate the ARID command into their LOGIN.CPL files.

If you force user validation on a remote node, you must supply users on that node with valid user ids to use in their ARID commands. Similarly, if a remote node forces user validation on your system, you should obtain valid user ids from the remote System Administrator and distribute them to your users for use in ARID commands.

If you answer NO to the FORCE USER VALIDATION? prompt, then:

- The calling user (master) may still issue the ARID command before making the remote call. In this case, slave login occurs just as in the case of forced user validation, above.

- If the master does not issue the ARID command, the slave takes on the same id as the master. In this case, the slave bypasses the standard login procedure on your system. Instead, a special, "fast" login takes place. Neither validation nor password checking occurs. (In fact, the master does not even pass a password to the slave.) The master's id need not appear on your User Profile List. However, if your system uses ACLs, and if the master (or another user with the same id) has ACL rights there, then the slave takes on those rights.

You should answer YES to the FORCE USER VALIDATION? prompt if:

- Security is a strong concern between your system and the remote node, and/or

- Your system and the remote node do not coordinate user ids — that is, ids do not uniquely identify users across the two systems.

The reason for the second condition above is as follows. Suppose your system and the remote node do not coordinate user ids. For example, suppose Harry Rosen has id HARRY on your system, and Harry Smith has id HARRY on the remote system. If you do not force user validation, then when Harry Smith attaches to a directory on your system, the slave that logs into your system will take on user id HARRY, and will receive all of Harry Rosen's ACL rights. Harry Smith can then access any files that Harry Rosen can. This situation cannot be circumvented by assigning the two Harrys different passwords, since the usual validation process is bypassed.

You should answer NO to the FORCE USER VALIDATION? prompt if:

- Security is not a strong concern between your system and the remote node, and

- User ids uniquely identify users across the two systems. (This means that Harry Smith on the remote node and Harry Rosen on your system should be assigned different ids, such as HARRYS and HARRYR. (On the other hand, if Erica Jones needs ids on both systems, she could be assigned the id ERICA on both systems.)

Even with coordination of ids, when you do not force user validation, you are essentially trusting that masters on the remote node are valid users who will not try to gain unauthorized access to your system.

If you do not administer all of the systems on your network, you will probably have to confer with the other System Administrators to decide whether or not to coordinate ids. If you do coordinate ids with a remote node, you and the other System Administrator must keep one another informed of all new ids. It is usually easiest if one Administrator takes sole charge of assigning ids.

The EDIT_PROFILE command VERIFY_USER can help you find out which ids have already been assigned on the system(s) you administer. (Refer to Chapter 4 for information on EDIT_PROFILE.)

In most cases, the use of forced user validation will be symmetric. That is, if you force user validation on a remote node, that node's System Administrator should probably also force user validation on your system. The reason for this is that id coordination is symmetric, and security needs are usually symmetric. However, NETCFG does not enforce this symmetry. It is possible for SYSA to force validation on SYSB while SYSB does not force validation on SYSA. In this case, SYSB users would have to use ARID to access SYSA files, but SYSA users would not have to use ARID to access SYSB files. This arrangement should only be made if user ids uniquely identify users across SYSA and SYSB.

Note

On a system that does not use ACLs, slaves' access rights are governed by standard password protection rules. (Masters supply directory passwords when attaching to the remote system.)

Rev. 18 Issues: If your system is a Rev. 19.0 system and a remote node is a Rev. 18 system, you should answer NO to NETCFG's FORCE USER VALIDATION? prompt for that node. Answering YES would require users on the remote node to issue the ARID command before accessing your system; but ARID is not available to Rev. 18 systems.

If your system is a Rev. 18 system, your NETCFG utility does not include the FORCE USER VALIDATION? prompt.

If your system and/or the remote node is a Rev. 18 system, the following are true:

- All slaves on one system (for masters on the other system) take on the user id SLAVE$. On a Rev. 18 system, SLAVE$ is identified as a phantom on the STATUS USERS list. On a Rev. 19.0 system, SLAVE$ is identified as a slave on the STATUS USERS list.

- On a Rev. 19.0 system, SLAVE$ need not be registered on the User Validation List.

- On a Rev. 19.0 system that uses ACLs, SLAVE$ must be assigned ACL rights to any files that are to be accessible to masters on a remote Rev. 18 system.

- On a system that does not use ACLs, SLAVE$'s access to files is governed by standard password protection rules. (The master supplies directory passwords when attaching to the remote system.)

Node-Node Passwords: As an added protection against security breaches through slaves, you can impose special node-node passwords to be used between masters and slaves. The node-node password ensures that a process that calls a slave is actually a FAM II master, and not a user-written emulator. Two systems that specify node-node passwords for one another must agree on the password. Node-node passwords are specified in the NETCFG dialog, in response to the NODE-NODE PASSWORD? prompts. The use of node-node passwords is strongly recommended.

Your node-node passwords appear in the NETCON file. It is therefore extremely important to ensure that only authorized users have Read access to NETCON.

A node-node password can be up to 32 characters long, and can be changed through the -PASSWORD command line option of NETCFG. (Refer to the section called CONFIGURING THE NETWORK, earlier in this chapter.) Once you assign the password, it is used internally by FAM II.

Controlling Remote Access to Your System's Files: FAM I

You should only use FAM I if your system is pre-Rev. 18.2 or if you need to communicate with a pre-Rev. 18.2 system.

If your system is a Rev. 19.0 system, your FAM phantom must be granted ACL rights to any files that are to be accessed remotely under FAM I. However, these files will then be accessible to any remote user on a FAM I node. Similarly, the FAM phantom on a remote node must be granted access to any files that your users wish to access remotely. The FAM phantom on each system needs ALL access rights to the FAM directory.

On a non-ACL system, remote file access under FAM I is governed by passwords, just as local file access is.

STARTING THE FILE ACCESS MANAGERS

If your system will use FAM II to communicate with other network nodes, you must be sure that:

- The NSLUSR directive has been included in the CONFIG file to configure FAM II slaves on your system. (Refer to the section called INCLUDING NETWORK-RELATED DIRECTIVES IN THE CONFIG FILE, earlier in this chapter.)

- You have answered YES to the ENABLE FAM II? NETCFG prompt for any post-Rev. 18.1 systems that agree to use FAM II to talk to your system.

- The PRIMENET* top level directory is present on your system and contains the file SLAVE.COMI.

You need do nothing else to start the FAM II slaves. Note that slaves cannot be force-logged out except in the case of a system shutdown.

If your system will use FAM I, the top level directory FAM must be present on your system. You must start the FAM I phantom by:

1. Typing "PH FAM>PH_FAM" at the supervisor terminal. The system responds with:

   PHANTOM IS USER nn

   where nn is the user number of the FAM I phantom (User FAM).

2. Typing "CHAP -nn 2" at the supervisor terminal. (nn is the user number returned in Step 1). This command sets the priority of the FAM to 2.

When FAM I starts, it checks to see whether there is a network node configured for FAM I communication. If not, it logs itself out and displays the following message at the supervisor terminal:

No nodes are enabled for FAM I.
The FAM phantom is not needed and is automatically logging out.

Should the FAM I phantom log out under other circumstances, restart it by repeating steps 1 and 2 above.

Note that if your system uses ACLs, the FAM I phantom user FAM must be granted appropriate access rights to any files that will be accessed (via FAM I) by remote users. In addition, FAM needs All access rights to the top-level directory FAM.

Note

The use of FAM I between systems that use ACLs is strongly discouraged.

ADMINISTERING THE FILE TRANSFER SERVICE (FTS)

This section describes the duties of the System Administrator with regard to FTS. Briefly, those duties are:

- Configuring and maintaining the FTS data base by means of the FTGEN utility.
- Assigning FTS-related access rights.
- Allocating enough space to the FTSQ* directory.
- Ensuring that FTS is properly shut down when necessary.

These tasks are described briefly below. For more detailed information, refer to the PRIMENET Guide.

As System Administrator, you should also ensure that FTS-related operator tasks are performed. Those tasks are described in the System Operator's Guide.

The FTGEN Utility

FTGEN is an interactive utility for defining, modifying, and displaying the characteristics of file transfer servers, queues, and sites. A file transfer server is a phantom process which handles file transfer requests. You can configure up to eight file transfer servers on your system. Each server takes requests from its own queue of file transfer requests. Each server can handle up to eight requests from its queue simultaneously. An additional special server, YTSMAN, receives file transfer requests from remote sites, and passes them to appropriate local servers. File transfer sites are the network nodes between which files can be transferred using FTS.

The file transfer servers and YTSMAN are started at the supervisor terminal by means of the FTOP command, which is discussed in the System Operator's Guide. Also discussed in that manual are other day-to-day maintenance tasks related to FTS.

The FTGEN utility accepts four categories of commands:

- Queue configuration commands
- Server configuration commands
- Site configuration commands
- General commands

The queue configuration commands are used to add, modify, and delete queues; block submissions to a queue; purge all requests from a queue; set the maximum size of a queue; set up a log file for a queue; and list or file the queue configuration.

The server configuration commands are used to add, modify, and delete servers; set up server log files; set server names, passwords, priority levels, timeslices, and port numbers; assign each server a queue; and list or file the server configuration. (Ports and port numbers are discussed in the PRIMENET Guide.)

Note

It is recommended that you configure FTS servers with server passwords to guard against unauthorized file transfers from remote FTS sites. You will have to confer with the System Administrators at the other FTS sites to ensure that they use matching passwords. Refer to the PRIMENET Guide for more information.

The site configuration commands are used to add, modify, and delete FTS sites; define your site's address; set up your site's log file; and list or file the site configuration.

The general commands apply to the entire FTS subsystem. They allow you to initialize the FTS subsystem database and to display the current status of the FTS configuration (sites, servers, and queues).

For specific information on these commands, refer to the PRIMENET Guide.

Access Rights

As System Administrator, you should be sure that the FTS servers and the users on your system have the access rights they need with regard to FTS.

ACL Systems: If your system uses ACLs, you must be sure to assign appropriate ACL rights to the FTS servers. (The user ids of the servers are defined when the servers are started by means of the FTOP command.) You must also ensure that the directory FTSQ* on your master disk is properly protected. Use the following guidelines when assigning access rights:

- All servers, including YTSMAN, need All access rights to FTSQ*.
- The user SYSTEM needs All access rights to FTSQ*.
- The $REST default access for FTSQ* must be set to ALURW rights, because users access FTSQ* through the FTR command.
- FTS servers must have DALURW rights to both source file and destination file directories. You should inform users of the servers' ids so that users can grant the servers access to their directories as necessary.

Non-ACL Systems: If your system does not use ACLs, you should inform users that they need the following access rights:

- Read access to source directories.
- Write and Delete access to any directories where they will create log files.
- Read, Write, and Delete access to destination directories.
- Owner status in any directories where they will be creating new files (for example, in destination directories, and in any directories where log files will be created).

On non-ACL systems, access rights are obtained according to the normal rules for using passwords. If the pathname of a source, destination, or log file contains passwords, the user must include the passwords on the FTR command line (and include the entire pathname in single quotes). It is recommended that owner passwords be used as a matter of course, particularly in view of the last point in the list above. Source and destination directories that are password-protected should be protected in such a way that the passwords confer the rights indicated above.

Users may be reluctant to communicate their passwords to people who want to send them files via FTS. As an alternative, users can create non-passworded directories exclusively for FTS use, and inform potential senders of files to use those directories.

FTSQ* Directory Size

The FTSQ* directory is used to hold copies of files that are being transferred, as well as to hold the server, queue, and site log files that you configure through the FTGEN utility. As System Administrator, you should therefore allocate a generous amount of space for FTSQ* (as for the SPOOLQ directory).

During the everyday running of FTS, the server, queue, and site log files grow. The only limit on their size is the maximum disk quota of the directory or the size of the partition. You should therefore ensure that these logs are archived at regular intervals so that FTSQ* does not fill up.

Shutting Down FTS

The proper method for shutting down FTS is as follows:

1. Close down any running FTS servers by means of the FTOP -STOP_SRVR command or the FTOP -ABND_SRVR command.

2. Log out the YTSMAN phantom using the LO command.

The force-logging out of FTS servers is strongly discouraged.

PRIMENET-RELATED OPERATOR TASKS

The following PRIMENET-related tasks are generally performed by the system operator, and are described in the System Operator's Guide:

- Using the ADDISK and SHUTDN commands to start up and shut down remote disks on your system.
- Using the REMOTE command to permit or deny remote nodes access to the disks on your system (if your system uses FAM I).
- Monitoring the day-to-day use of the File Transfer Service (FTS).
- Monitoring half duplex lines (assigning, unassigning, starting, and stopping them).
- Communicating with the operators on other nodes of the network.
- Monitoring the Network Event Log.

APPENDIXES

A

Physical Device Numbers

INTRODUCTION

Each physical disk or disk partition has a physical device number identifying the type of storage device, the drive unit on which it is mounted, and, for partitions, the size of the partition and its location on the disk pack. These physical device numbers are used in the commands: ADDISK, ASSIGN DISK, CONFIG, COPY, DISKS, FIX_DISK, FIXRAT, MAKE, SHUTDN, and UNASSIGN DISKS.

This appendix provides information on commonly used devices. A complete reference, including information on obsolete devices, can be found in the System Operator's Guide.

DRIVE UNIT NUMBERS

The drive unit number identifies the physical drive on which the disk is mounted. It is important to keep a record in the system logbook of drive unit numbers and of the physical device numbers (including partitioning) for disks mounted on these drives.

Drive unit numbers for storage modules are set by removable buttons. The system installer should have labeled these units; in many cases the drive unit number will be on one of the push button switches on the front of the drive unit.

STORAGE MODULES

Storage modules exist in the three sizes indicated below. (They have 1040 words per record.)

Size of Module	Number of Heads	Type
40 MBytes	5	6
80 MBytes	5	6
300 MBytes	19	6

Storage modules are usually partitioned (subdivided), with each partition being treated as if it were an actual physical device. Partitions must be an integral number of heads in size and must be offset an even number of heads from the start of the disk pack. However, the last partition on the disk may contain an odd number of heads.

The physical device number is constructed as a 16-bit number, in octal (see Figure A-1).

A complete list of valid physical device numbers for storage modules is given in Table A-1.

Example: A system contains three drive units; drives 0 and 1 have 300 MByte storage modules, and drive 2 has an 80 MByte storage module (see Figure A-2). The modules are to be partitioned as follows:

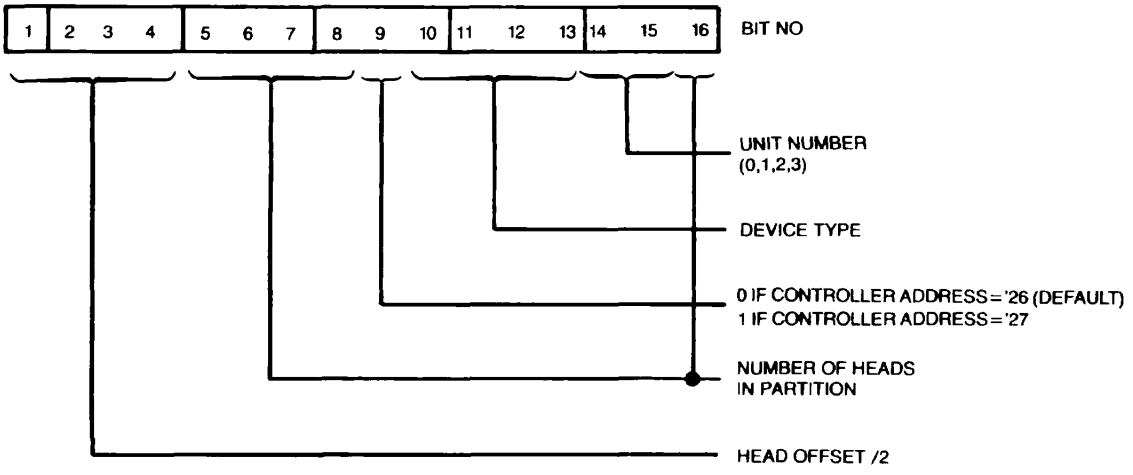
Drive 0 Partitions of 2, 2, 6, 2, 2, 2, and 3 heads
Drive 1 Partitions of 14 and 5 heads
Drive 2 Partitions of 2 and 3 heads

The physical device numbers are:

Drive 0	Drive 1	Drive 2
000460	003462	000464
010460	071063	010465
021460		
050460		
060460		
070460		
100461		

This example is illustrated in Figure A-2.

In all cases the drives are connected to the default controller address of '26. Each partition is treated by PRIMOS as if it were a separate physical device.



Construction of Physical Disk or Partition Number
Figure A-1

Table A-1

Physical Device Numbers for Storage Modules and Fixed Media Devices

Number of Heads in Partition	Starting Head Number							
	2	4	6	8	10	12	14	16
1	———	———	02006z	———	———	———	———	———
2	00046y	01046y	02046y	03046y	04046y	05046y	06046y	07046y
3	———	01046z	———	———	———	———	———	———
4	00106y	01106y	02106y	03106y	04106y	05106y	06106y	07106y
5	00106z	———	———	———	———	———	———	07106z
6	00146y	01146y	02146y	03146y	04146y	05146y	06146y	07146y
7	———	———	———	———	———	———	06146z	———
8	00206y	01206y	02206y	03206y	04206y	05206y	06206y	07206y
9	———	———	———	———	———	05206z	———	———
10	00246y	01246y	02246y	03246y	04246y	05246y	06246y	07246y
11	———	———	———	———	———	———	———	———
12	00306y	01306y	02306y	03306y	04306y	05306y	06306y	07306y
13	———	———	———	———	———	———	———	———
14	00346y	01346y	02346y	03346y	04346y	05346y	06346y	07346y
15	———	———	———	———	———	———	———	———
16	00406y	01406y	02406y	03406y	04406y	05406y	06406y	07406x
17	———	———	———	———	———	———	———	———
18	00446y	01446y	02446y	03446y	04446y	05446y	06446x	———
19	———	———	———	———	———	———	———	———
20	00506y	01506y	02506y	03506y	04506y	05506x	———	———
21	———	———	———	———	———	———	———	———
22	00546y	01546y	02546y	03546y	04546x	———	———	———
23	———	———	———	———	———	———	———	———
24	00606y	01606y	02606y	03606x	———	———	———	———
25	———	———	———	———	———	———	———	———
26	00646y	01646y	02646x	———	———	———	———	———
27	———	———	———	———	———	———	———	07646z
28	00706y	01746x	———	———	———	———	———	———
29	———	———	———	———	———	———	06706z	———
30	00746x	———	———	———	———	———	———	———

All partitions marked with x must be combined with an 11-head partition starting at head 30 for full utilization of the disk capacity. Combining these partitions with a partition of less than 11 heads will reduce the storage capacity by approximately 16K bytes per unused head.

y is twice the unit number of the drive unit on which the disk is mounted. z is twice the drive unit number plus one.

Table A-1 (continued)

Physical Device Numbers for Storage Modules
and Fixed Media Devices

Starting Head Number									
18	20	22	24	26	28	30	32		
———	11006z	———	———	———	———	———	———	1	Number of Heads in Partition
10046y	11046y	12046y	13046y	14046y	15046y	06046x	———	2	
10046z	———	———	———	———	———	———	———	3	
10046z	11106y	12106y	13106y	14106y	15106x	———	———	4	
———	———	———	———	———	———	———	———	5	
10146y	11146y	12146y	13146y	14146x	———	———	———	6	
———	———	———	———	———	———	———	———	7	
10206y	11206y	12206y	13206x	———	———	———	———	8	
———	———	———	———	———	———	———	———	9	
10246y	11246y	12246x	———	———	———	———	———	10	
———	———	———	———	———	———	———	17246z	11	
10306y	11306x	———	———	———	———	———	———	12	
———	———	———	———	———	———	16306z	———	13	
10346x	———	———	———	———	———	———	———	14	
———	———	———	———	———	15346z	———	———	15	
———	———	———	———	———	———	———	———	16	
———	———	———	———	14406z	———	———	———	17	
———	———	———	———	———	———	———	———	18	
———	———	———	13446z	———	———	———	———	19	
———	———	———	———	———	———	———	———	20	
———	———	12506z	———	———	———	———	———	21	
———	———	———	———	———	———	———	———	22	
———	11546z	———	———	———	———	———	———	23	
———	———	———	———	———	———	———	———	24	
10606z	———	———	———	———	———	———	———	25	
———	———	———	———	———	———	———	———	26	
———	———	———	———	———	———	———	———	27	
———	———	———	———	———	———	———	———	28	
———	———	———	———	———	———	———	———	29	
———	———	———	———	———	———	———	———	30	

Table A-1 shows all the valid physical device numbers for the 40, 80, 300, and 600 MB disks. To use Table A-1:

1. Decide upon the number of surfaces in the partition.
2. Decide upon the head number of the first head in the partition.
3. Look up the physical device number in the table.

Notes

If the partition defined is not in Table A-1, then it is not a legal partition.

All partitions must begin on an even head number.

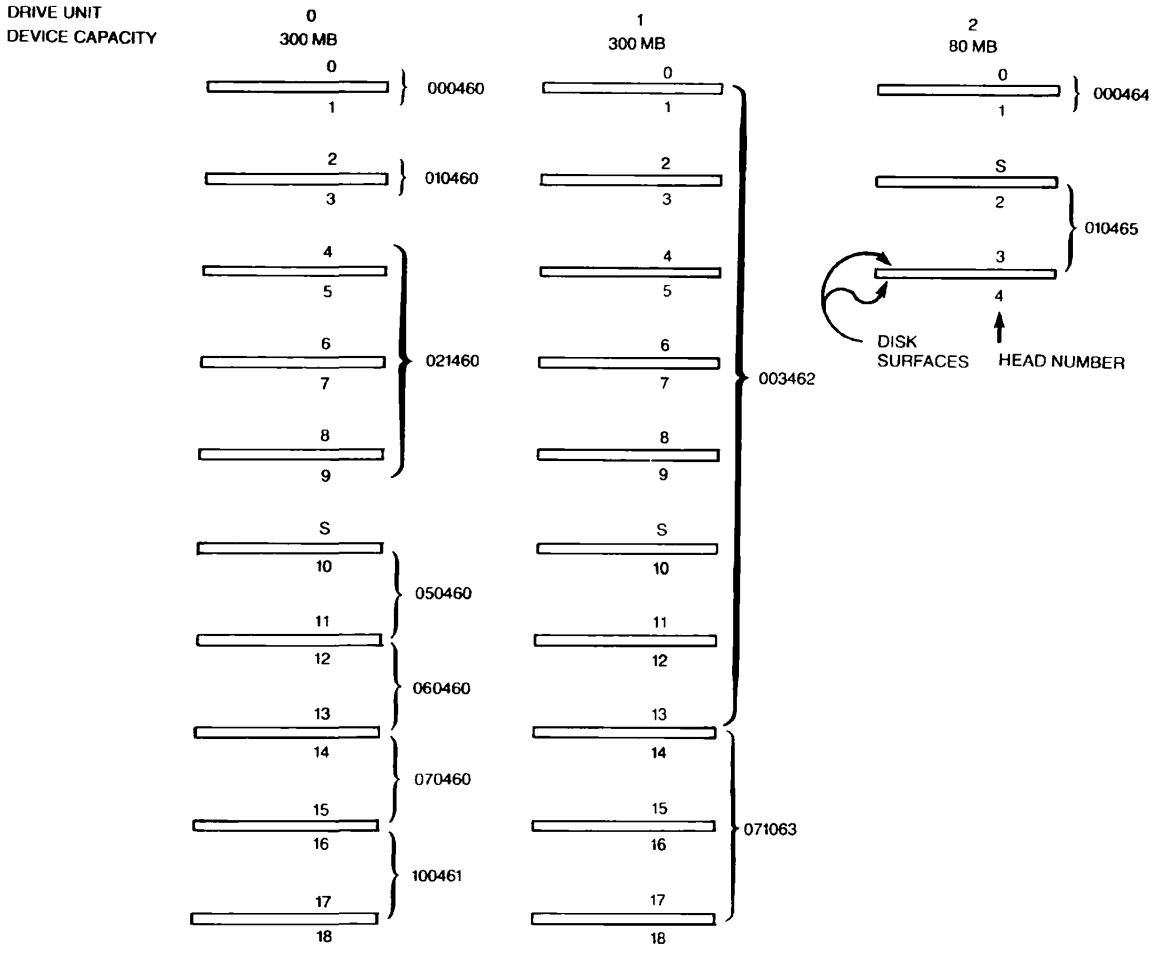
To make the most efficient use of the storage device, only the last partition on the disk pack should have an odd number of surfaces.

FIXED MEDIA DEVICES (WINCHESTER DISKS)

Fixed media devices exist in the four sizes indicated below:

Size of Device	Number of Heads
34	3
68	3
160	10
600	40

Their device numbers are calculated as shown in Table A-1.



Example of Storage Module Partitions
Figure A-2

CARTRIDGE MODULE DEVICES (CMDs)

Cartridge module devices (CMDs) exist in three sizes: 32 MBytes, 64 MBytes, and 96 MBytes. They may be partitioned as indicated below.

CMD Type	Platter(s)	First Controller	Second Controller
32 MB	Removable	6z (16 MB)	26z (16 MB)
	Nonremovable	10006z (16 MB)	10026z (16 MB)
64 MB	Removable	6z (16 MB)	26z (16 MB)
	Nonremovable	10046y (32 MB)	10066y (32 MB)
		11006z (16 MB)	11026z (16 MB)
		or	or
		10046z (48 MB)	10066z (48 MB)
96 MB	Removable	6z (16 MB)	26z (16 MB)
	Nonremovable	10046y (32 MB)	10066y (32 MB)
		11046y (32 MB)	11066y (32 MB)
		12006z (16 MB)	12026z (16 MB)
		or	or
		10106y (64 MB)	10126y (64 MB)
		12006z (16 MB)	12026z (16 MB)
		or	or
		10106z (80 MB)	10126z (80 MB)
		or	or
		10046y (32 MB)	10066y (32 MB)
		11046z (48 MB)	11066z (48 MB)

Notes

y is twice the drive unit number (0-3) on which the disk is mounted. z is twice the drive unit number plus one.

The nonremovable surfaces of the 64MB CMD can be organized as 1 or 2 partitions.

The nonremovable surfaces of the 96MB CMD can be organized as 1, 2, or 3 partitions.

B

External LOGIN And LOGOUT Programs

INTRODUCTION

At Rev. 19, separate external programs may be written to monitor logins and logouts. Prior to Rev. 19, one program had to serve both needs.

At login time, the system looks for an program in CMDNCO named LOGIN. At logout time, it looks first for a program named LOGOUT. If LOGOUT doesn't exist, it then looks for LOGIN. (Suffixes are not allowed on either name.)

Note

At Rev. 20, the system will look only for CMDNCO>LOGOUT at logout time. It will look for CMDNCO>LOGIN only at login time.

This appendix contains:

- Some guidelines for writing external login programs.
- A sample external login program.
- A sample COMINPUT program to compile and load the sample external login program.

GUIDELINES

The external login program may regulate the use of the operating system in addition to the LOGIN facility provided by PRIMOS. It may access confidential system information such as valid user-ids, project-ids, and per-user accounting information. Therefore, precautions must be taken when writing an external login program to prevent inadvertent or malicious misuse of the program. The following are factors to be considered by an administrator when writing an external login program.

- External login programs must reside in the directory CMDNCO.
- They must be named LOGIN or LOGOUT. No suffix is allowed.
- Access to the program should be strictly controlled.
- All files that are opened by the external login program must be closed before the program completes execution.
- CONTROL-P (used to BREAK out of programs) is inhibited when the external login program begins execution. The external login program must re-enable breaks when it completes execution.

  Breaks are disabled while the external login program is running because the following undesirable conditions can occur if a user is allowed to QUIT in the middle of the external login program:

  — Files left open.

  — User logged in without having gone through all validation checks.

- At Rev. 19, a new subroutine exists to allow external login programs to record project-ids given at login time. The calling sequence is:

```
DCL PRJID$ ENTRY (CHAR (32) VAR)

CALL PRJID$ (PROJECT_ID)
```

  The sample program given in this appendix shows how to use this subroutine in a FORTRAN program.

- If user input from the user terminal is required during the external LOGIN process, the external login program must set a timeout condition in order to avoid an indefinite wait for a user response.
- If a password or other validation code is required for a login, the user should be given a finite number of chances to enter the correct password. If the correct password is not given in this number of trials, the external login program should log the user out.

- The external login program may forcibly log a user out if the user does not pass the validation process.

- The external login program should be designed to meet the needs of the individual site. Since these needs may vary over time, the program should be easy to modify and understand.

SAMPLE EXTERNAL LOGIN PROGRAM

```
C  EXTLOG.FTN, MARY>UTILS, PP&M, 07/12/82
C  Sample external login program for rev 19.0.
C  Copyright (c) 1982, Prime Computer, Inc., Natick, MA 01760

C  Description:  Logs user_id, project_id, date/time, cpu/io
C                Displays system news
C
C  Abnormal conditions:  None.
C
C  Implementation:
C
C  Modifications:
C  Date       Programmer     Description of modification
C  7/10/82    Mary Kroening  Updated for rev 19.
C  6/18/80    John Cowles    Updated for rev 18.
C  11/01/78   Barry Burke    Initial coding.
C
       SUBROUTINE MAIN

$INSERT SYSCOM>KEYS.INS.FTN
$INSERT SYSCOM>ERRD.INS.FTN

       INTEGER*2          LOGDAT(45), INFO(8), BUFF(20), CODE, NWR,
      +                   CMDPAS (3), LOGOUT, PRJID(17), TYPE, DATA(80)

       LOGICAL BRKON,FANTOM,LOGIN,NAMEQ$

       PARAMETER          BRKON=.FALSE.

       DATA      CMDPAS/'PASSWD'/              /* YOUR 'LOGIN_ACCTG' PASSWORD

C
C---START.
C
       LOGOUT = $60

C
C--- NOW GET SOME VITAL INFO FROM USERS COMAND LINE.
C
       CALL RDTK$$(3,INFO,BUFF,20,CODE)       /* RESET COMMAND LINE POINTER
       CALL RDTK$$(K$READ,INFO,BUFF,20,CODE) /* READ 1st TOKEN
       IF (INFO(1).EQ.5) GO TO LOGOUT         /* CHECK FOR NULL TOKEN
       LOGIN = (NAMEQ$('LOGI',4,BUFF,4))      /* CHECK FOR LOGIN/LOGOUT

C
C--- NOW CHECK IF THIS A PHANTOM USER.
C--- PHANTOM USERS WILL ONLY HAVE THE WORD 'LOGIN' IN THEIR COMMAND
C--- LINE ON LOGIN.
C
       IF (LOGIN) CALL RDTK$$(K$READ,INFO,BUFF,20,CODE)  /*READ NEXT TOKEN
       FANTOM = .FALSE.
       IF (INFO(2) .EQ. 0)    FANTOM=.TRUE.
```

```
C
C--- NOW GET SYSTEM DATA FOR THIS USER (FROM TIMDAT).
C
      CALL TIMDAT(LOGDAT,28)

C
C--- HERE WE ATTACH TO 'LOGIN_ACCTG' TO DO SOME SECURITY CHECKING.
C--- IN THIS UFD WE KEEP THE FOLLOWING FILE:
C---           LOGFIL = FILE IN WHICH IS STORED EACH USER'S LOGIN
C---                    LOGOUT TIMDAT INFO.  THIS IS USED AS THE DATA
C---                    FOR A WEEKLY ACCOUNTING PACKAGE.
C
      CALL ATCH$$('LOGIN_ACCTG',11,K$ALLD,CMDPAS,K$IMFD,CODE)   /* DO NOT SET HOME.
      IF (CODE.EQ.0) GOTO 10       /* ON ERROR GIVE USER ERROR MESSAGE
      CALL ERRPR$(K$IRIN,CODE,'External Login',14,'ATCH$$',6)
      GO TO LOGOUT                 /* AND LOG HIM OUT (HE WILL CALL)

C
C--- SET UP ENTRY FOR LOGFIL.
C
10    CALL PRJID$(PRJID)                              /* GET USER'S PROJECT_ID
      CALL FSUB$A(PRJID,34,PRJID(1)+3,34,' ')         /* PAD NAME WITH SPACES
      CALL MSUB$A(PRJID,34,3,34,LOGDAT,90,57,88) /* MOVE PRJID INTO LOGDAT
      LOGDAT(45) = 'LO'                               /* LOGOUT KEY
      IF (LOGIN)  LOGDAT(45) = 'LI'                   /* LOGIN

C
C--- NOW OPEN FILE.
C---       SUBROUTINE 'OPENF' IS ACTUALLY A CALL TO SRCH$$, BUT INSTEAD
C---       OF SUPPLYING A 'TYPE' VAR., A NUMBER SPECIFYING THE NUMBER OF
C---       RETRIES TO ATTEMPT ON A FILE-SYSTEM ERROR. THE MAIN REASON
C---       FOR DOING THIS IS BECAUSE MORE THAN ONE USER MAY BE ATTEMPTING
C---       TO WRITE TO THE FILE AT THE SAME TIME, RESULTING IN 'FILE IN USE'
C---       ERROR
C
      CALL OPENF(K$RDWR,'LOGFIL',6,1,5,CODE)
      CALL ERRPR$(K$IRIN,CODE,'External Login',14,'SRCH$$',6)
      CALL ATTDEV(5,8,1,100)          /* INCREASE RECORD SIZE

C
C--- POSITION TO THE END OF THE FILE.
C
      CALL PRWF$$(K$POSN+K$PREA,1,LOC(0),0,1000000,NWR,CODE)

C
C--- AND WRITE ENTRY
C
      WRITE(5,1000) LOGDAT
1000  FORMAT(3A2,8I6,I3,33A2)
      CALL SRCH$$(K$CLOS,0,0,1,0,CODE)     /* CLOSE THE FILE.
      CALL ERRPR$(K$IRIN,CODE,'External Login',14,'SRCH$$',6)

30    IF (LOGIN) GOTO 100
```

```
31    CALL EXIT              /* WAS A LOGOUT, LET HIM GO.
      GOTO 31                /* TO KEEP HIM FROM GETTING BACK IN

C
C--- LOG THE USER OUT
C
60    CALL LOGO$$(0,0,0,0,0000000,CODE)   /*BYE-BYE

C
C--- WELCOME THE USER IN
C
100   CALL SRCH$$(K$CLOS,0,0,1,0,CODE)
      CALL TNOU ('Welcome to PRIME Computer!',26)

C
C--- DISPLAY SYSTEM BANNER FILE
C
      CALL SRCH$$(K$READ,'DATA',4,1,TYPE,CODE)
      IF (CODE.NE.0) GO TO 70
50    CALL RDLIN$(1,DATA,40,CODE)
      IF (CODE.NE.0) GOTO 70
      CALL TNOU(DATA,80)

C
C--- HERE WHEN END OF DATA IN MESSAGE FILE DETECTED
C
70    CALL SRCH$$(K$CLOS,'DATA',4,1,TYPE,CODE)

C
C--- ATTACH TO USER'S ORIGIN UFD AND ENABLE BREAKS
C
      CALL AT$OR(K$SETH,CODE)
      CALL  BREAK$(BRKON)
999   CALL  EXIT
      END

C
C  SUBROUTINE OPENF
C  FUNCTIONS EXACTLY AS SRCH$$ BUT HAS AN
C  EXTRA PARAMETER (NUM) DEFINING THE NUMBER OF ATTEMPTS
C  AT AN OPERATION BEFORE GIVING UP
C
      SUBROUTINE OPENF(KEY,NAME,NAMLEN,UNIT,NUM,CODE)
      INTEGER KEY,NAME(16),UNIT,TYPE,CODE,I,NUM
C
$INSERT SYSCOM>KEYS.INS.FIN
$INSERT SYSCOM>ERRD.INS.FIN
      NOLIST
C
      DO 20 I=1,NUM
      CALL SRCH$$(KEY,NAME,NAMLEN,UNIT,TYPE,CODE)
      IF (CODE.EQ.E$FIUS) GOTO 10
      RETURN
10    CALL SLEEP$(0001010)     /* SLEEP FOR 1.01 SECONDS
```

```
20    CONTINUE
      RETURN
      END
```

SAMPLE COMINPUT PROGRAM

The following CPL program compiles and loads the sample external login program shown above. It also copies the external login program into CMDNC0 and names it LOGIN. (This final step is necessary if the system is to run the external login program whenever a user logs in.)

```
/*  extlog.comi, mary>utils, pp&m, 07/12/82
/*  compile and load sample external login program
/*  then copy to cmdnc0>login (must not have .save suffix)
/*
ftn extlog -64v
seg -load
split 40000
mix on
common 4000
s/lo extlog 0 4000 4000
d/li vapplb
d/li
map
save
return
share
ex
quit
copy ex4000 cmdnc0>login -nq
co -end
```

C
Reverting Disks

INTRODUCTION

Rev. 18-format disks will run on Rev. 19 systems. However, Rev. 19-format disks will not run on Rev. 18 systems. In general, this should pose no problems. However, there may arise some rare cases in which you want to use a Rev. 19-format disk on a Rev. 18 system.

There are two methods for handling this, only one of which is recommended. They are as follows:

Recommended Method

1. Back the disk up to tape, using the MAGSAV utility with the -NO_ACL option. This saves the data in a Rev.-18-compatible format.

   Note

   If you think you might want the Rev.-19 formatted material again, with its ACLs and quotas intact, you might also want to do either a MAGSAV without the -NO_ACL option, or a physical backup with PHYSAV or COPY_DISK.

2. Mount the disk on the Rev. 18 system.

3. Run the Rev. 18 MAKE on the disk, to remake it in Rev. 18 format.

4. Use MAGRST to reload the saved data from tape.

Emergency Method

If you don't have time to back up your disk, you can use the following emergency procedure. Please check with your analyst before doing so. Remember, this procedure is not recommended.

Password-protected Disks: If the disk in question has no ACLs on it, you can simply mount it on your Rev. 18 system and run it. Be careful, though, NOT to run FIXRAT, COPY_DISK, or PHYSAV on the disk: they will not work properly and may do damage. In addition, if the disk is a split disk, you cannot use it for paging on the Rev. 18 system.

ACL-protected Disks: If the disk does use ACL protection, you will have to remove the ACLs before you can mount the disk on a Rev. 18 system. To assist you in doing this quickly, there exists a tool called STRIP_ACLs (detailed below). Use this tool to remove ACLs from the disk (or remove them by hand, if there are only a few) and then mount the disk on the Rev. 18 system. As above, DO NOT run FIXRAT, COPY_DISK, or PHYSAV on the disk while it is mounted on the Rev. 18 system; and do not attempt to use the disk for paging.

STRIP_ACLS

STRIP_ACLS is a tool which removes all ACLs from a directory tree. By default, it removes ACLs from the subtree defined by the current attach point, but any tree may be specified on the command line. The user has the option of having STRIP_ACLS run FIX_DISK after it has finished removing the ACLs.

To use STRIP_ACLS, either attach to the TOOLS directory or copy STRIP_ACLS.CPL into the directory from which it is to be used.

Note

Because STRIP_ACLS is a recursive CPL program, it must reside in the directory from which it is invoked. Do not attempt to invoke the STRIP_ACLS program via a pathname. (For example, the command line R TOOLS>STRIP_ACLS will not work.)

The command syntax of STRIP_ACLS is:

```
R STRIP_ACLS { -HELP
             { [pathname] [-FIX_DISK pdev [-COMDEV]] [-REPORT] }
```

The first format causes STRIP_ACLS to list its syntax. The second actually executes STRIP_ACLS. If no pathname is supplied, the current directory is assumed as the starting point. Protect access is required on all directories in the subtree; if ACLs are to be removed from an entire partition, it is recommended that a Priority ACL be placed on the partition before running STRIP_ACLS.

If the -FIX_DISK option is given, FIX_DISK will be run after the ACLs have been removed from the disk. The pdev must be supplied with this option. Use the -COMDEV option if you are fixing the command device. STRIP_ACLS supplies the -FIX, -CMPR, and -DUFE options automatically.

Note that if the -COMDEV option is given and STRIP_ACLS was invoked from the command partition, STRIP_ACLS will terminate abnormally after FIX_DISK has run. This does not cause a problem since all conversion and FIX_DISK activities have been completed at this point.

> **Caution**
>
> If STRIP_ACLS is being used to remove all ACLs from a disk, we recommend that you use the -FIX_DISK option to clean up the disk after the ACLs have been removed. If you wish to use the disk under a Rev. 18 system after removing the ACLs, you MUST invoke FIX_DISK with the -CMPR option (supplied automatically by the -FIX_DISK option) before the disk may be used under Rev. 18.

The -REPORT option causes STRIP_ACLS to list each directory it converts to a password directory. The directory is listed when it is converted.

Note that ACLs are not the only Rev. 19 structure which cannot be supported by Rev. 18 systems. If you run a disk which has had all its ACLs removed by STRIP_ACLS on a Rev. 18 system, the following restrictions are still imposed by the new BADSPT file format:

- You may not use any Rev. 18 physical disk utilities (such as COPY_DISK or PHYSAV).
- You may not use the disk for paging. (This applies to split partitions only).

Note also that all ACLs removed by STRIP_ACLS are lost; in order to restore ACL protection to the disk, the ACLs must be replaced manually.

INDEX

Index

USER SURVEY

Tell us how we're doing, and we'll send you a free Programmer's Companion.

Your name _____

Company or School _____

Address _____

City, State, Zip _____

1. What is your job title or function? _____

2. What specific task describes what you do? _____

3. Does your company or school own a Prime computer? ☐ YES ☐ NO

a. If YES, which model? ☐ 450 ☐ 550 ☐ 650 ☐ 750 ☐ OTHER

b. Is it networked with other Prime computers? ☐ YES ☐ NO

c. Is it networked with any of these? ☐ IBM ☐ CDC ☐ UNIVAC ☐ HONEYWELL

d. Which of these software packages do you use?

☐ FORTRAN	☐ COBOL	☐ BASIC/VM
☐ FORTRAN 77	☐ PL/I-G	☐ POWER
☐ MIDAS	☐ DBMS	☐ SPSS
☐ RPGII	☐ FORMS	☐ PRIMENET
☐ RJE	☐ PASCAL	☐ OAS
☐ DBG	☐ DPTX	

e. Have you read any other Prime documents? ☐ YES ☐ NO

f. If YES, which ones? _____

4. Are you presently evaluating Prime? ☐ YES ☐ NO

Is the documentation playing a part? ☐ YES ☐ NO

5. What book are you reviewing? _____

6. My initial reaction to this book was:

☐ EXCELLENT ☐ GOOD ☐ FAIR
☐ VERY GOOD ☐ FAIR

7. After reading it my reaction was: ☐ BETTER ☐ THE SAME WORSE

If BETTER or WORSE why? _____

8. How often have you used this book?

☐ EVERY DAY ☐ FAIRLY OFTEN
☐ VERY OFTEN ☐ JUST GOT IT

9. Did the book have the content you expected? ☐ YES ☐ NO

If NO, why? _____

10. Did you find the organization useful? ☐ YES ☐ NO

If NO, why? _____

11. How did you find the examples? ☐ TOO MANY ☐ ABOUT RIGHT ☐ TOO FEW

12. How did you find the illustrations? ☐ TOO MANY ☐ ABOUT RIGHT ☐ TOO FEW

13. Could you locate the information you needed? ☐ YES ☐ NO

14. Was the index adequate? ☐ YES ☐ NO

15. Please evaluate our writing style and edit quality?

a. Clarity ☐ VERY CLEAR ☐ AVERAGE ☐ UNCLEAR

b. Tone ☐ CORRECT ☐ NEUTRAL ☐ STILTED

c. Technical level ☐ TOO HIGH ☐ ABOUT RIGHT ☐ TOO LOW

d. General writing quality ☐ EXCELLENT ☐ AVERAGE ☐ FAIR ☐ POOR

e. Editorial quality ☐ EXCELLENT ☐ AVERAGE ☐ FAIR ☐ POOR

16. Which of these manufacturers documentation have you used the most?

☐ IBM ☐ DIGITAL ☐ DG
☐ HP ☐ CDC ☐ UNIVAC
☐ HONEYWELL ☐ BURROUGHS ☐ WANG
☐ GE ☐ XEROX ☐ _____

How is Prime compared to theirs?

☐ MUCH BETTER ☐ A LITTLE WORSE
☐ A LITTLE BETTER ☐ MUCH WORSE
☐ SAME

Any comment on your rating? _____

17. Please evaluate the graphic quality:

a. Rate the general presentation ☐ EXCELLENT ☐ GOOD ☐ POOR
☐ VERY GOOD ☐ FAIR

b. Do you like the paper color? ☐ YES ☐ NO

c. Did you find any quality defects? ☐ YES ☐ NO

d. Do you like the way we've used color for abbreviations and user input? ☐ YES ☐ NO

e. Did the shading over the tables and charts make them EASIER or HARDER to read? ☐ EASIER ☐ HARDER

f. Do you like the Programmer's Companion concept? ☐ YES ☐ NO ☐ HAVEN'T SEEN ONE

g. Which form of bindery do you find most useful? ☐ BOUND ☐ LOOSE-LEAF

18. Do you know about the AIDUS program? ☐ YES ☐ NO

19. Any other comments:

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

20. What book don't we offer that you'd like to see?

_____

Thank you for filling out the survey. Check off which Programmer's Companion you would like to receive.

- ☐ PRIMOS
- ☐ FORTRAN 77
- ☐ ASSEMBLY
- ☐ ADMINISTRATOR
- ☐ FORTRAN
- ☐ BASIC/VM
- ☐ POWER
- ☐ WORD PROCESSING

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 10B
Prime Park, Natick, MA 01760