900 Series HP 3000 Computer Systems

HP Security Monitor/iX Manager's Guide

HEWLETT
PACKARD

HP Part No. 32650-90455
Printed in U.S.A. April 1994

First Edition
E0494

Hewlett-Packard Company
3000 Hanover Street
Palo Alto, CA 94304 U.S.A.

Printing History

The following table lists the printings of this document, together with the respective release dates for each edition. The software version indicates the version of the software product at the time this document was issued. Many product releases do not require changes to the document. Therefore, do not expect a one-to-one correspondence between product releases and document editions.

Edition	**Date**	**Software Version**
First Edition	April 1994	C.50.00

Preface

MPE/iX, Multiprogramming Executive with Integrated POSIX, is the latest in a series of forward-compatible operating systems for the HP 3000 line of computers.

In HP documentation and in talking with HP 3000 users, you will encounter references to MPE XL, the direct predecessor of MPE/iX. MPE/iX is a superset of MPE XL. All programs written for MPE XL will run without change under MPE/iX. You can continue to use MPE XL system documentation, although it may not refer to features added to the operating system to support POSIX (for example, hierarchical directories).

Finally, you may encounter references to MPE V, which is the operating system for HP 3000s, not based on PA-RISC architecture. MPE V software can be run on the PA-RISC (Series 900) HP 3000s in what is known as "compatibility mode".

Organization of This Manual

This manual consists of eight chapters, two appendixes, and an index as follows:

Chapter 1 *Introduction*

Chapter 2 *Overview of System Security* provides an overview of security and a brief discription of the file structures on MPE systems.

Chapter 3 *Managing System Users With Passwords and Logon Restrictions* explains the use of passwords and logon restrictions.

Chapter 4 *Protecting Your System With Access Control Definitions (ACDs)*

Chapter 5 *Protecting Files with File Access Restrictions and Lockwords* describes the use of file access restrictions and lockwords.

Chapter 6 *Protecting the System with Capabilities* describes special abilities.

Chapter 7 *Auditing System Use* describes monitoring of the system with logging.

Chapter 8 *Using the Security Configurator (SECCONF)* describes security using the HP Security Monitor.

Appendix A *FOS Security Maintenance Checklist* is a checklist for FOS system users.

Appendix B *Error Messages* contains a brief explaination of each error mes age.

Conventions

`UPPERCASE` In a syntax statement, commands and keywords are shown in uppercase characters. The characters must be entered in the order shown; however, you can enter the characters in either uppercase or lowercase. For example:

```
COMMAND
```

can be entered as any of the following:

```
command        Command        COMMAND
```

It cannot, however, be entered as:

```
comm           com_mand       comamnd
```

italics In a syntax statement or an example, a word in italics represents a parameter or argument that you must replace with the actual value. In the following example, you must replace *filename* with the name of the file:

`COMMAND` *filename*

bold italics In a syntax statement, a word in bold italics represents a parameter that you must replace with the actual value. In the following example, you must replace ***filename*** with the name of the file:

`COMMAND(`***filename***`)`

punctuation In a syntax statement, punctuation characters (other than brackets, braces, vertical bars, and ellipses) must be entered exactly as shown. In the following example, the parentheses and colon must be entered:

`(`*filename*`):(`*filename*`)`

underlining Within an example that contains interactive dialog, user input and user responses to prompts are indicated by underlining. In the following example, yes is the user's response to the prompt:

```
Do you want to continue? >>  yes
```

`{ }` In a syntax statement, braces enclose required elements. When several elements are stacked within braces, you must select one. In the following example, you must select either `ON` or `OFF`:

$$\texttt{COMMAND} \begin{Bmatrix} \texttt{ON} \\ \texttt{OFF} \end{Bmatrix}$$

`[ ]` In a syntax statement, brackets enclose optional elements. In the following example, `OPTION` can be omitted:

`COMMAND` *filename* `[OPTION]`

When several elements are stacked within brackets, you can select one or none of the elements. In the following example, you can select `OPTION` or *parameter* or neither. The elements cannot be repeated.

$$\texttt{COMMAND}\ \textit{filename} \begin{bmatrix} \texttt{OPTION} \\ \textit{parameter} \end{bmatrix}$$

Conventions (continued)

`[ ... ]` In a syntax statement, horizontal ellipses enclosed in brackets indicate that you can repeatedly select the element(s) that appear within the immediately preceding pair of brackets or braces. In the example below, you can select *parameter* zero or more times. Each instance of *parameter* must be preceded by a comma:

`[,`*parameter*`][...]`

In the example below, you only use the comma as a delimiter if *parameter* is repeated; no comma is used before the first occurrence of *parameter*:

`[`*parameter*`][,...]`

`| ... |` In a syntax statement, horizontal ellipses enclosed in vertical bars indicate that you can select more than one element within the immediately preceding pair of brackets or braces. However, each particular element can only be selected once. In the following example, you must select **A, AB, BA**, or **B**. The elements cannot be repeated.

$\left\{\begin{matrix}\mathtt{A}\\ \mathtt{B}\end{matrix}\right\}$ `| ... |`

... In an example, horizontal or vertical ellipses indicate where portions of an example have been omitted.

`Δ` In a syntax statement, the space symbol `Δ` shows a required blank. In the following example, *parameter* and *parameter* must be separated with a blank:

`(`*parameter*`)Δ(`*parameter*`)`

□ The symbol □ indicates a key on the keyboard. For example, RETURN represents the carriage return key or Shift represents the shift key.

CTRL *character* CTRL *character* indicates a control character. For example, CTRL **Y** means that you press the control key and the Y key simultaneously.

Contents

Figures

Tables

1

Introduction

HP Security Monitor/iX Manager's Guide

This manual is written for the System Managers or System Security Managers. It contains information about managing passwords, managing ACDs, security in the Hierarchical File System and the *HP Security Monitor/iX*.

The *User's Guide to MPE/iX Security* and the *Manager's Guide to MPE/iX Security* comprise the fundamental operating system(FOS) set of HP 3000 MPE/iX security documents.

The *HP Security Monitor/iX User's Guide* and the *HP Security Monitor/iX Manager's Guide* replace the generic FOS manuals when you purchase the HP Security Monitor/iX software.

2

Overview of Security on the MPE/iX Computer System

Facilities for implementing security measures on HP 3000 Computers are contained in the MPE/iX Fundamental Operating System (FOS). This chapter describes the components of computer system security:

- Physical security - control of access to system components.
- Procedural security - establishment and control of security procedures.
- System security - control of system access using the security features provided by the MPE/iX operating system.
- Components of the Account Structure.
- Security policy.
- General security threats.
- Defense against security threats.

Physical Security

Physical security involves the prevention of physical damage to system hardware, and prevention of the corruption of software . The term "hardware" includes the central processing unit (CPU), System Console, terminals, and other peripherals, such as printers, disc drives, and tape drives. The term "software" includes the operating system, programs, and data.

The causes of damage to hardware and software can range from deliberate sabotage or vandalism, to inadvertent damage caused by unskilled users. Regardless of the cause, such damage usually can be prevented by restricting physical access to hardware and logon access to software.

Physical access to hardware is usually effected by perimeter controls, which restrict entry into areas in which computer equipment is located, including system consoles. Perimeter controls include locked computer rooms, fenced building sites, and guard stations at building entrances. Access to the terminal servers and the network wires leading between the system unit and the terminal servers must be protected. Physical access can be controlled by issuing keys and ID badges only to authorized persons.

Access to software is usually controlled by logon restrictions. Such restrictions include the use of passwords, establishment of accounts and groups, and control of user capabilities. Access to programs and files can be provided by assigning users to accounts, issuing appropriate capabilities, enforcing the use of passwords, lockwords and by creating programs and files in groups that belong to special accounts. The physical aspect of securing access to software involves prevention of physical access to terminals, and limitations on or prevention of access via communication lines.

Procedural Security

Procedural security deals with the establishment and enforcement of security procedures. Some of these procedures may be independent of the type or types of computers involved. Others may not. For example, perimeter security controls are usually similar for all type of systems. But desktop computers may require forms of antitheft protection not required by mainframes.

Procedural security regulates the performance of duties associated with system operation and use, and with the physical storage of system information. Common security practices include partitioning computer operating duties, using several operators, and storing backup tapes at bonded, offsite depositories. Procedural security also encompasses and may regulate company policies that deal with information security, such as policies that regulate the way individuals manage their own passwords.

System Security

System security is provided by security features built into MPE/iX, by the ways in which the account structure of the system is organized. System security features fall into five categories:

- Identification of users.
- Authentication of users.
- Authorization of users.
- Control of access to system resources.
- Auditing system usage.

Identification

Every user must have a unique logon identity, or ID, by which he or she is identified as a legitimate system user. Without a valid ID, a user cannot log on to the system. Commonly, user IDs consist of a user name and account name.

Authentication

When a user logs on, the system attempts to authenticate the logon ID. The system checks its directory for the existence of the ID, then verifies the user's identity by checking the password. Entry of an incorrect ID or password is enough to prevent access to the system.

Authorization

System access is provided at several levels, from the lowest, available to all users, to the highest, open only to system and security management. When users are first authorized to use the system, they are assigned codes that identify the level of access to which they are permitted. As users execute system functions and tasks, the system constantly checks their authority to do so. The various levels of user authority are described below, under *User Roles*.

The system checks a user's identity and capabilities to determine access level. For example, some commands are available to all users (lowest level of capability). Other commands are available only to System Managers (SM capability), or System Operators (OP capability). Each time a user issues a command, the system checks the user's capabilities to make sure he or she is allowed to use that command.

Programs also have capabilities, which are assigned by the programmer at the time the program is created. The capabilities assigned to a program allow it to access particular functions. When a program that has special capabilities run, the system does not require the user to have those capabilities. The program runs and exercises its capabilities in conjunction with those of the user. In addition to the capabilities just described, some programs check user capabilities before issuing certain functions.

Certain commands are reserved to the Console, and can be issued only from the System Console. This includes a category of commands that can be issued only after entering a CTRL A at the System Console. There is an exception to this rule. Through the use of the `:ALLOW` command, the System Operator (Console Operator) can give other users the ability to run specified Console commands (but not CTRL A commands) from their own terminals.

Some Console commands are associated with devices. One example is the `:DOWN` command, which makes devices unavailable. The use of device associated commands also can be given to users at terminals other than the Console. This is accomplished via the ASOCTBL utility and the `:ASSOCIATE` command. System Manager (SM) capability is required to run the ASOCTBL utility, and System Operator or System Supervisor (OP) capability is required to use the `:ASSOCIATE` command.

User Roles

Assigned capabilities and account membership determine a person's role as an MPE/iX user. In general, roles fall into one of three categories: system administrators, account managers, or general user.

- System administrators are responsible for system operations. Titles include System Manager, System Supervisor, and System Operator (the operator at the console). Each type of system administrator has a different role, different capabilities, and different responsibilities.
- Account Managers usually have the title Account Manager. Account Managers are responsible for administering an account. Each account has at least one manager.
- A general user has no administrative capabilities other than managing his or her own password, files, and UDCs (User Defined Commands).

The System Manager

A System Manager is a user with System Manager (SM) capability. SM capability lets you manage the system and create accounts, groups, and users. In MPE/iX, SM capability is associated with the `SYS` account. The system tape you receive with your HP 3000 Computer System designates an initial System Manager (`MANAGER.SYS`). The initial System Manager can assign SM capability to other users.

The System Manager's functions include:

- Creating and maintaining accounts, groups, and users.
- Changing account, group, and user passwords.
- Obtaining reports of account use for billing and other purposes.
- Managing regular system backups and establishing standard backup procedures. (The System Supervisor performs backups.)
- Designating system level User Defined Commands (UDCs).
- Configuring, managing, and auditing system security.
- Creating and managing Access Control Definitions for files and devices.
- Supervising other System Administrators.

The System Manager automatically has all capabilities. A System Manager can perform all System Supervisor, System Operator, Account Manager, and general user tasks.

The System Supervisor

The System Supervisor (OP capability) exercises day-to-day control of the system. OP capability permits you to:

- Store and restore files.
- Manage system scheduling subqueues.
- Alter the system configuration.
- Maintain system and user logging facilities.

- Display certain items of system information.

The System Manager assigns OP capability to accounts. An Account Manager who has OP capability in his or her account can assign it to other users in the account.

The System Operator

The System Operator is the user logged on to the System Console. The System Operator derives his or her capabilities from the System Console, not from any capabilities inherent in the title. The System Operator also may be known as the Console Operator. In many systems, users with System Supervisor capability serve as System Operator. The System Operator is responsible for:

- Monitoring the status of the system.
- Monitoring the console.
- Responding to console requests.

The Account Manager

An Account Manager (AM capability) manages all users and groups in an account. The System Manager assigns an Account Manager for an account when creating that account. The Account Manager can, in turn, assign Account Manager capability to other users within the account.

An Account Manager's functions include:

- Creating and maintaining groups.
- Changing user passwords within the group.
- Creating and maintaining users.
- Creating and managing ACDs for files in the account.
- Managing account level UDCs.
- Insuring the security of the account.
- Storing and restoring account files (some files may also require SM, OP, or PM capability).

General Users

General users are those who are not System Managers, System Supervisors, System Operators, or Account Managers. General users' responsibilities with respect to account structure and security include:

- Managing and maintaining the security of the files they create.
- Protecting their own user passwords.
- Establishing and maintaining their own UDCs.

Components of the Account Structure

The account structure consists of four components: accounts, groups, users, and files.

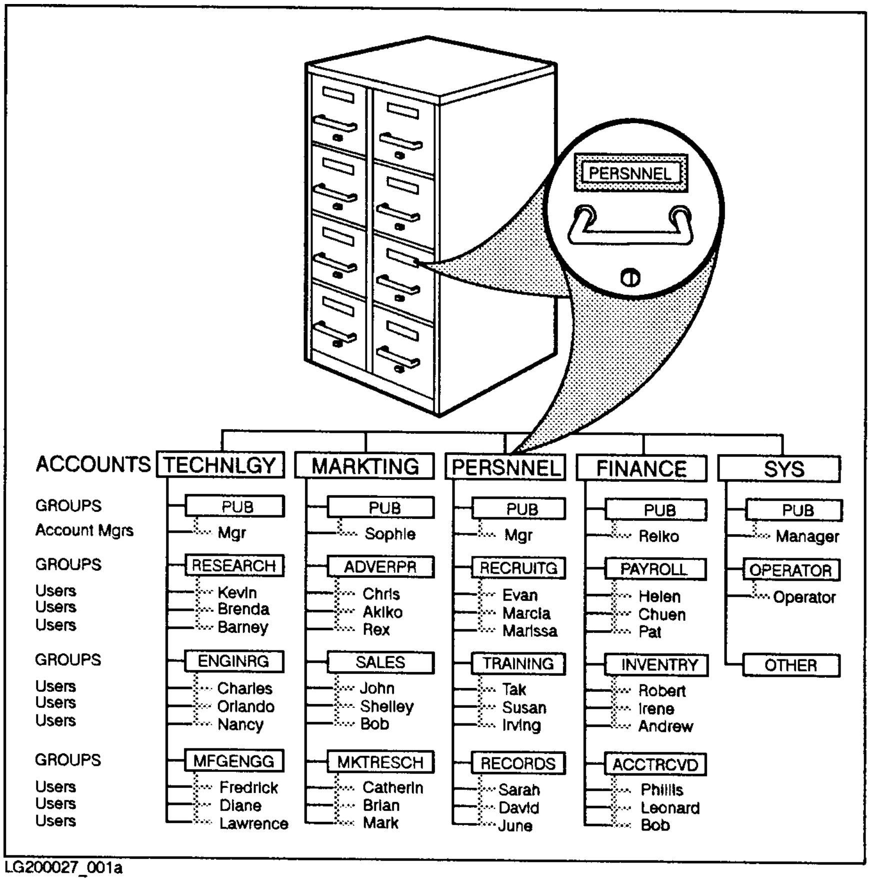
- Accounts are the basic structure for organizing users and information in the system. System users and system information belong to accounts.
- Groups further organize users and information within accounts.
- Users belong to the account, but access files by logging on to a group. If they know the appropriate group passwords, users can log on to any group within the account.

  Generally, users are associated with a home group to which the system logs them on when they do not specify a group name in their logon command.
- Files store the information. Any time that you run a program, use a spreadsheet, or compose a letter, you are using files. Files belong to groups within an account.

The system directory is the system's internal list of accounts, groups, users, and files. It keeps track of their characteristics and their relationships.

Figure 2-1 illustrates the relationship between accounts, groups, and users. Accounts (TECHNLGY, MARKTING', SYS, for example) are shown horizontally, across the top of the diagram. Groups (RESEARCH, SALES, RECORDS, for example) are stacked vertically under their accounts. Users (KEVIN, CHARLES, DIANE, for example) appear under their home groups. The solid black lines in Figure 2-1 indicate firm, primary relationships.

Notice that all users have their strongest relationships with their accounts, and all groups have their strongest relationships with their accounts. The gray lines indicate less solid relationships; although users have a solid relationship with the account, they also have a convenience relationship with a home group. Users are most likely to work in and to have files stored in their home group.



LG200027_001a

Figure 2-1. Account Relationships

Notice in Figure 2-1 the occasional odd spelling, like TECHNLGY and RECRUITG. All account, group, user, and file names must be eight characters or fewer in length.

The Individual Account

Figure 2-2 shows the structure of an individual account. Not all accounts look like the one in Figure 2-2, but most are similar. Every account has a name, a PUB (PUBLIC) group, and an account manager. When you first create an account, the account manager has the PUB group as a home group.

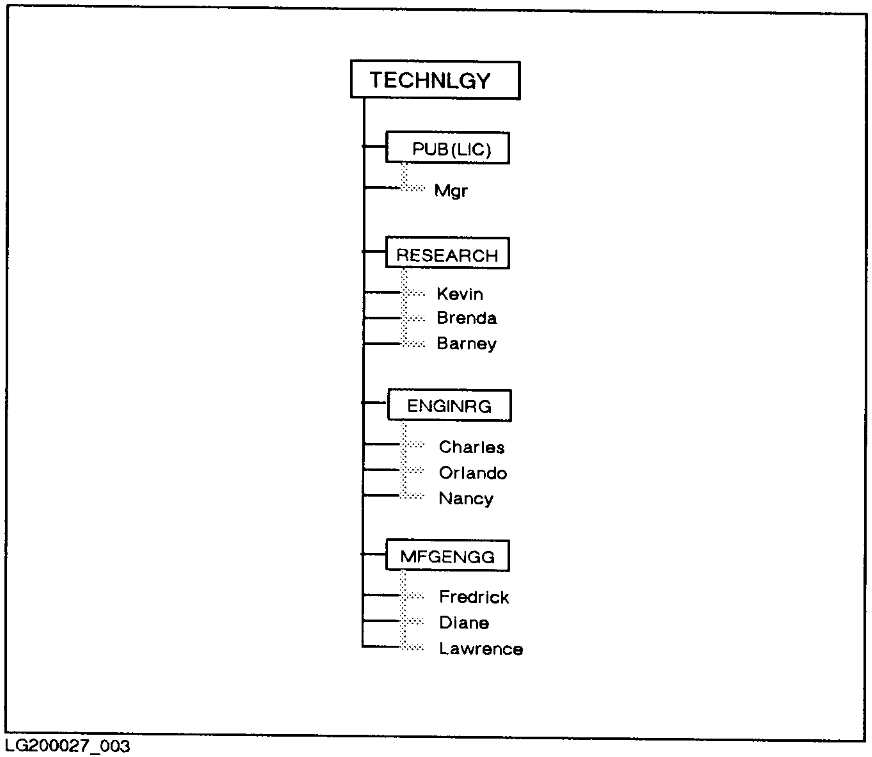


Figure 2-2. An Individual Account

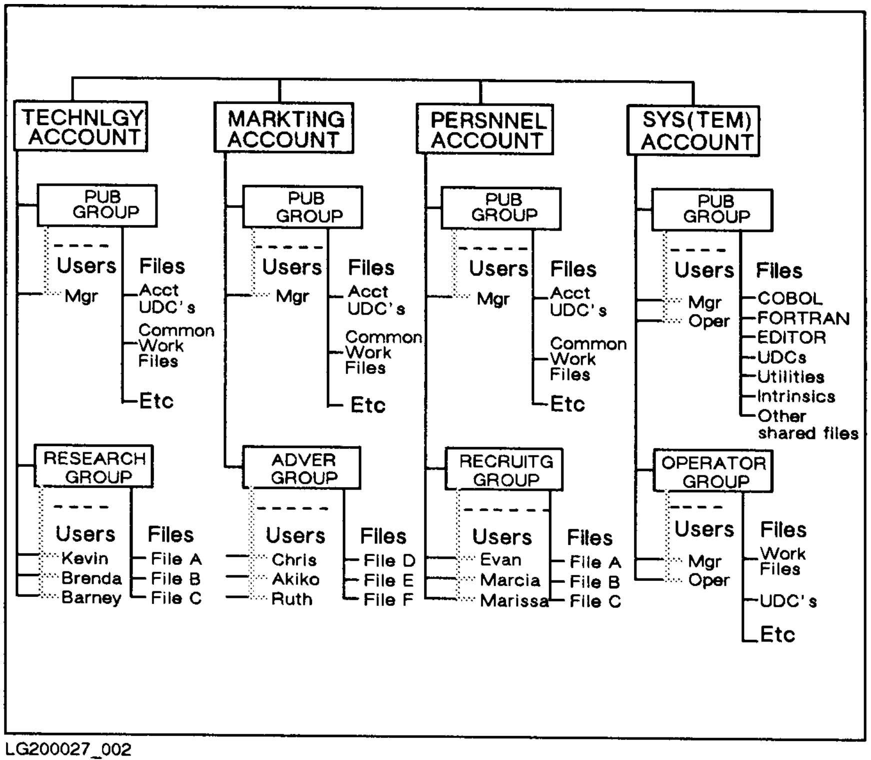
The account manager is responsible for establishing the groups and users within the account. In the example above, the group named RESEARCH is the home group for three users, ENGINRG is the home group of three users, and MFGENGG is the home group of three users. In each case, the users are likely to do their work in their home group. Because their main relationship is to the account, they can log on to any group in the account if they know the group passwords.

You can also create users who do not have a home group. These users can log on to any group, but must specify the desired group and its password when they log on.

Using Files

When you do almost any kind of work with a computer, you work with files. Reports, spreadsheets, program listings, letters, management tools, and more all exist within the system in the form of files.

The files belong to the groups in an account as shown in Figure 2-3.



LG200027_002

Figure 2-3. Groups, Users, and Files

The system stores the files necessary for operating the computer. For example, utilities, system libraries, program subsystems, languages, compilers, user-defined commands, and the system itself are in the SYS (SYSTEM) account's PUB group.

The PUB groups in other accounts contain files that the users of those accounts share. Files in other groups are usually the private files of that group's users.

Standard Characteristics

Every system has standard accounts, groups, and users. Each system has a SYS (for system) account. It contains the operating system, shared programs, and files shared by the members of all accounts. Each account has a group named PUB (for public). The PUB account contains certain publicly accessible files. For example, the PUB group of the SYS account contains system programs available to all users. The user MANAGER is built in to the SYS account. MANAGER is the initial system manager.

Creating Naming Conventions

Notice that each account, group, and user in Figure 2-3 has a name. Files also have names. An account, group, user, or file name must be eight characters or fewer in length. It must begin with an alphabetic character. Subsequent characters can be alphabetic or numeric.

Account names must be unique, but notice that each account has a group named PUB. Group names must only be unique, within an account. Files must have unique names within a group, but two files in different groups might have the same name within an account. User names must be unique within an account, but two users in different accounts might have the same user name.

For example, in Figure 2-1, there is a user named BOB in both the FINANCE and MARKTING accounts.

User Names

The system distinguishes between users with the same name by using both the user and account name as the user's fully qualified name. By convention, fully qualified user names take the form:

username.accountname

For example, the fully qualified name of the user BOB in the FINANCE account is BOB.FINANCE. The BOB in MARKTING has the full name BOB.MARKTING. The two BOBs may or may not be the same person, but to the system they are different users. When users log on to the system, they use their fully qualified names. For example:

```
HELLO BOB.FINANCE
```

Group Names

Groups have fully qualified names that are similar to fully qualified user names. A fully qualified group name has the following form:

groupname.accountname

For example, the PUB group of the TECHNLGY account has the fully qualified name PUB.TECHNLGY. The PUB group of the SYS account has the fully qualified name of PUB.SYS. Think of the notation PUB.SYS as short for the PUB group of the SYS account.

File Names

Fully qualified file names include the file's name, its group, and its account. A fully qualified file name has the following format:

filename.groupname.accountname

For example, a file named FILEA in the RESEARCH group of the TECHNLGY account has the fully qualified name FILEA.RESEARCH.TECHNLGY. A file's fully qualified name distinguishes it from any other file in the system. You can use a file's fully qualified name to access it from anywhere in the system (if you pass the file access restrictions described later in this chapter).

Hierarchical file system (HFS)

As of Release 4.5, the MPE/iX file system is *hierarchical* (tree structured) and can contain files at many different levels. This organization provides a special kind of file called a **directory**. Instead of holding data, directories contain lists of files and pointers to those files. A directory can also contain other directories. This organization is similar to the file systems on UNIX® or MS-DOS® systems.

The new file organization still includes the familiar accounts, groups, and users. The hierarchical file system (called HFS, for short) extends the traditional MPE file system features so the operating system is more flexible.

You're used to referring to files, groups, and accounts using the traditional MPE syntax: `FILE1.PUB.SYS`. You can still use MPE syntax. You can also make use of a new syntax called HFS syntax, which looks like this: `/SYS/PUB/FILE1`.

The MPE/iX Release 4.5 enhancements are compared to previous releases in Table 2-1.

**Table 2-1.
Where Accounts, Groups, Directories, and Files Can Be Located**

Location	Before Release 4.5	Release 4.5 and After
Highest level	Accounts	Root
Under root	Root not visible	Accounts, directories, or files
Under accounts	Groups	Groups*
Under groups	Files	Directories or files
Under directories	Directories not available	Directories or files
* This is an initial release restriction that may be lifted in a future release.		

Figure 2-4 shows how you can organize files, accounts, groups, and directories in the file system. Notice that accounts, directories, groups, and files all connect back to one directory designated by a "/" (slash). This is referred to as the *root* or the *root directory*.

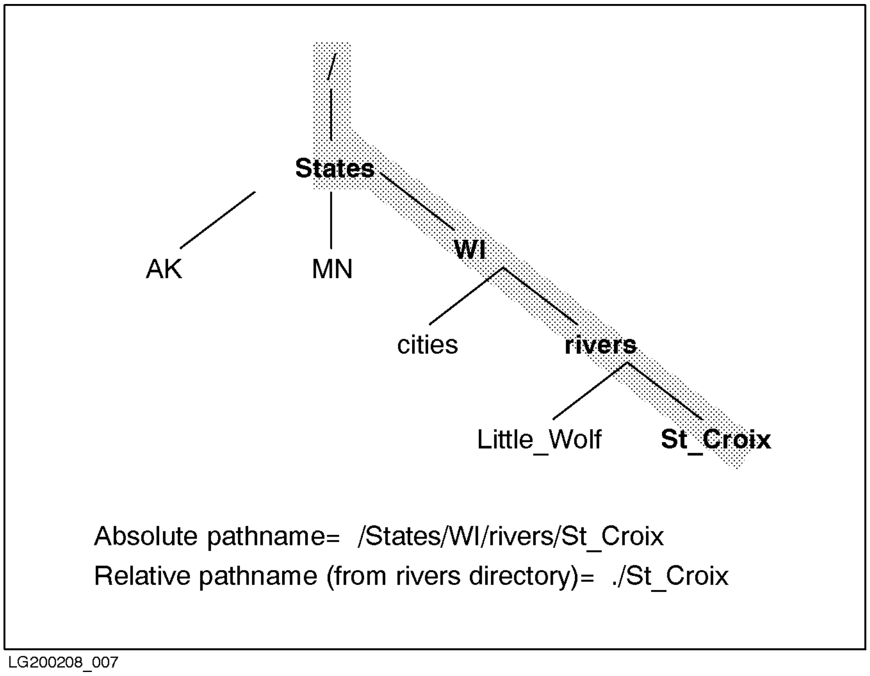

/
States
AK
MN
WI
cities
rivers
Little_Wolf
St_Croix
Absolute pathname= /States/WI/rivers/St_Croix
Relative pathname (from rivers directory)= ./St_Croix
LG200208_007

Figure 2-4. MPE/iX File System Example

HFS file names

MPE/iX Release 4.5 allows you to assign longer file names than in previous versions of MPE/iX. Table 2-2 summarizes name lengths for accounts, groups, directories, and files previous to Release 4.5 and after Release 4.5.

Table 2-2. Maximum Lengths of Account, Group, Directory, and File Names

Type	MPE Syntax	HFS Syntax
Account name	8 uppercase characters	8 uppercase characters
Group name	8 uppercase characters	8 uppercase characters
Directory name	Not available	16 mixed case characters (if directly under root or directly under a group). Up to 255 characters (elsewhere).
File name	8 uppercase characters	16 mixed case characters (if directly under root or directly under a group). Up to 255 characters (elsewhere).

HFS syntax

Table 2-3 summarizes some of the syntax enhancements introduced by the MPE hierarchical file system. The syntax that you are used to still works for files in groups and accounts. So to use HFS syntax, you must precede file and directory names with `./` or `/`. Otherwise, MPE/iX treats the names using traditional MPE syntax rules.

This manual refers to files that are named using HFS syntax as *HFS files*.

Table 2-3. Syntax Summary

Item	MPE Syntax	HFS Syntax
Specify file name	No special beginning character required: `FILE.GRP.ACCT`	Name must be preceded by a ./ (dot slash) or / (slash): `/ACCT` or `./dir1`
Name separators	. (period); / separates lockwords	/ (slash)
Way of specifying files	Bottom up: `FILE.GRP.ACCT`	Top down: `/ACCT/GRP/FILE`
Case sensitivity	Not case sensitive; all characters are shifted to uppercase	Case sensitive: `/DIR/FILE1` and `/DIR/file1` are two different files
Special characters	Only alphanumeric characters	Alphanumeric, - (hyphen), . (dot), and _ (underscore) are allowed
First character	Must be alphabetic	Can be alphanumeric, _ (underscore), or . (dot) but not - (hyphen)

Designing an Account Structure

Your account structure should reflect your organization's structure and the way in which you intend to use your system. If your firm uses a single computer system, an account structure similar to a corporate organization chart (like the one in Figure 2-1) makes sense. Your computer system uses the same structure as your organization, because it identifies and tracks the same kinds of information.

If your system belongs to a functional division of your firm, for example, engineering or purchasing, your accounts might correspond to projects or products. If your firm is a service bureau, your system might have an account for each customer.

Controlling Access to System Resources

System performance can suffer if too many jobs and/or sessions are running at the same time. Setting limits on the number of jobs and sessions that can run concurrently protects the system from inadvertent or deliberate attempts to degrade its performance. Setting limits on the number of active devices in use at any time helps control the user load, and also helps prevent access by unauthorized users.

Additional methods for controlling resources and security are provided by the *Access Control Definition Facility*.

Auditing System Usage

When activated, the MPE/iX system logging facility maintains log records of system use. For example, these records can tell you how often abortive logons are attempted, and identify the devices from which the attempts were made. For systems in which security is important, log records should be reviewed regularly and often. For more information on system logging see the *Auditing System Use* chapter of this manual.

Security Policy

A computer security policy is a set of laws, rules, and practices that regulate how an organization manages, protects, and distributes sensitive information.

A security policy will cover the following aspects of computer operations:

- Types of facilities in which systems can be located.
- Who is allowed physical access to the system.
- Who is allowed to log on to the system.
- What audit records are to be logged.
- Types of permissible access to files.
- Types of permissible access to devices.
- Which security features will be enabled (for example):
  - Use of ACDs to protect files and devices.
  - Passwords required.
  - Embedded passwords in jobs not allowed.

This list is not intended to be a comprehensive statement of a security policy, but a guide to what should be included in your security policy. Current and new users must be made familiar with the guideline and indoctrinated in its use. Periodic reinforcement of the message is a must to assure continuing compliance with the policy and its updates.

Security Considerations

This section deals with the overall problem of security as it affects computer installations. It discusses threats to computer security, and provides guidelines for meeting those threats. The table at the end of this chapter synopsizes this material in a quick reference form.

Computer security deals with more than just the security of the computer itself. The environment in which the system operates also must be secure. Otherwise, you may find that all the hardware and software security precautions in the world will not be enough to protect your system from damage or penetration.

General Security Threats

General security threats fall into four broad categories:

1. Loss of use.
2. Loss of performance.
3. Disclosure of information.
4. Loss of integrity.

Loss of Use

This type of loss can affect both equipment and data. It can result from such causes as theft, vandalism, fire, and natural catastrophes, such as earthquakes and floods. Data on magnetic media is particularly susceptible to accidental or deliberate corruption or erasure by magnetic fields, and data on disk can be lost due to head crashes.

Regardless of cause, this type of loss is characterized by the inability to use the property. This is usually accompanied by the need to spend funds to replace it.

Loss of Performance

This type of loss can result from such causes as simple wear and tear, incorrect usage, and sabotage. The loss is characterized by a decrease in operating efficiency, and may go on for some time before being discovered.

Disclosure of Information

This type of loss may result from accident, theft or simple mistakes. The types of information involved can range from business records to scientific and military data. The loss is generally characterized by a loss of some economic, scientific, or military advantage.

Loss of Integrity

Integrity constitutes a loss of quality or trust in data resulting from incorrect or malicious modification.

Recognizing Security Incursions

Evidence of the occurrence of major theft, vandalism, fire, earthquake, and similar causes of loss is usually obvious. Evidence of attempts at unauthorized entry and unauthorized usage is much less so.

The best way to find evidence of attempts at unauthorized entry and unauthorized usage is continuous monitoring of system log files. For example, a Type 115 (Console) Log Record that shows numerous unsuccessful connection attempts can be considered reasonable evidence of attempts at unauthorized entry.

Monitoring the Type 144 (File Open) Log Record can disclose a pattern of unsuccessful attempts to open files. This may mean that an unauthorized person has gained access to the system, or an authorized user is trying to access files to which he or she has no authorization.

Close scrutiny and analysis of log files on a regular basis reveals the frequency of attempts to violate system security, how successful your security measures are in thwarting such attempts, and the location of weaknesses in your defenses.

General Defenses Against Security Threats

Some types of defenses are effective against all three types of general security threats. The second and third types of security threats also may require additional defenses that are specific to the form of the threat.

A major first line of general defense is your company's security guideline . All present users and system administrators should be thoroughly familiar with the guideline and its implementation. All new users should be made familiar with the guideline and its implementation before being allowed on the system.

Defenses Against Loss of Use

Examples of defenses against loss of use include prevention of access, fire prevention and firefighting measures, safeguards against shock and impact in earthquake regions, and storage off site, in antimagnetic containers, of information on magnetic media. Insurance is another form of defense. Although it cannot prevent physical loss, it can mitigate financial loss.

Prevention of Access

Prevention of access is the primary form of defense against theft and vandalism. Such defenses take several forms:

- Physical prevention of access to premises, and physical prevention of access to equipment within the premises.
- Denial of use even though the equipment can be physically approached.

Physical prevention of access takes many forms, including:

- Perimeter defenses, such as fences with controlled access points, intruder warning devices, remote television cameras, searchlights, and guard dogs.
- Internal defenses, such as guarded entry points to buildings and areas, metal detectors, identification badges, sign-in logs, combination or magnetic card locks on laboratory and computer room doors, and locks for desks, cabinets, workstations and personal computers. In addition, physically attaching small equipment to desks can help prevent theft, although not vandalism.

Denial of access even though equipment can be physically approached can apply to machinery of many types. For computers and computer systems, methods include:

- Key locks for workstations and personal computers.
- Passwords, password protection, limitations on the number of logon attempts allowed, and file and device ACDs. Systems connected to external networks and accessible by telephone present particular problems of their own. For example, if a caller fails to log on within the number of times allowed, that person need only hang up and try again. The problem is aggravated by the fact that it is possible to set up a computer to make the calls!
- One way to limit damage is to ensure that a user's access is removed as soon as access is no longer needed. Idle accounts or accounts of user's no longer at the company or organization should be considered a potential security risk.

Defenses Against Loss of Performance

Although wear and tear on equipment certainly is a cause of performance loss, it is a business problem, rather than one of security. System administrators should be aware of it and request the replacement of worn equipment as needed.

In the same sense, loss of performance or data due to incorrect usage also is not a security problem. On the other hand, it is one with which system administrators must be involved. For example, incorrect usage can deny use of the system to other users by tying up too much of the CPU. Solutions include:

- Limitations on access by limiting user capabilities, or giving users access only to the resources they need to execute their tasks.
- User training.

Defenses Against Data and Performance Loss Due to Sabotage

One type of sabotage involves access to the computer or system by unauthorized persons. For the most part, preventative measures are the same as those described under *Prevention of Access,* above. In particular, you should be aware of the fact that anyone who can access the System Console can execute a CTRL A, then execute any command that can be invoked from the "=" prompt. Such commands include `=ABORTJOB`, `=ABORTIO`, `=LOGOFF`, `=LOGON`, and `=SHUTDOWN`.

Another type of access available from the System Console is that provided by executing a CTRL B. This provides access to the system hardware via the system diagnostics. The CTRL B function can be physically disabled. Discuss this with your Hewlett-Packard Service Engineer.

A type of sabotage much harder to prevent is sabotage from internal sources. Examples include disgruntled employees, and accidental sabotage resulting from the inadvertent introduction of destructive software (Trojan horses, viruses) into the system.

Sabotage by users with otherwise legitimate access to the system can be minimized by enforcing limitations on capabilities and access. System logging facilities can be used to establish strict accountability for all users. Such accountability cannot prevent sabotage, but can aid in identifying the culprit. Even users at the highest levels can be made accountable by such techniques as maintaining a log of all who access or modify the system configuration.

Due to the power of the privileged mode capability (PM), System Managers should allocate it only to accounts, groups and users with an imperative need. As an example of the dangers inherent in the PM capability, it permits the use of DEBUG on system files, and lets persons with the capability place unauthorized software on the system.

Prevention of accidental sabotage from destructive software can be minimized or prevented by education, strict rules against using unauthorized software, and well publicized penalties for doing so. Establishment of accountability can, again, aid in identifying the offender in such incidents.

Defenses Against Information Disclosure

Total prevention of accidental information disclosure is rarely possible. Employee education and appeals to employees' sense of company or national loyalty can help mitigate the problem, but not prevent it. Another technique is to disseminate vital information strictly on a need-to-know basis.

Deliberate theft of information in physical form, such as on disk, tape, and paper, can be minimized using the same techniques as those for preventing theft of equipment: prevention of access.

Techniques for preventing access include locking desks, cabinets, and files. Store media in locked cabinets rather than open racks, and enforce strict control over the distribution of sensitive documents.

When the information on media is no longer needed, the media is often reused by simply writing over the existing data. Depending on the medium, the data may be readable until it is overwritten, even if the medium have been reformatted. This is an easily overlooked breach of security.

Before returning disk, disk packs, and tapes to reuse, all labels should be removed in order to prevent a thief from easily picking out the tapes that may contain important information. Each disk or tape should be carefuly erased with a degausser type bulk tape eraser.

Techniques for protecting information in the system itself include locking computers, enforcing the use of passwords, prohibiting embedded passwords, and clearing computer screens and screen buffers.

Avoid storing files containing sensitive information in accounts to which all or many users have access, such as `PUB.SYS` and system libraries. Be particularly aware of the sensitivity of the `PUB.SYS` account and `NL.PUB.SYS`. Only System and Account Managers should ever have the capability to change the accessibility level of the account. Also be sensitive to the fact that programs stored in XL.PUB.SYS" are executable by any user, and that a virus-infected program stored there is in a particularly advantageous place to damage your system.

Finally, use ACDs with all files and devices, and share files only with those who have a need to know.

Table 2-4. Synopsis of Possible Security Threats and Defenses

Possible Threats	Possible Defenses
Loss of use.	Prevent access. Perimeter defenses. Fences. Guarded entries. Lighting. Intruder warning devices. Surveillance devices. Guard dogs. Internal defenses. Guarded entries. Metal detectors. Identification badges. Sign-in logs. Door locks. Locks - desk, storage, computers. Physical restraints on equipment. Denial of use. Mandatory passwords. No embedded passwords. Logon limitations. Restrictions on use of modems. Fire prevention. Shock and impact prevention. Offsite storage. Antimagnetic storage. Insurance.
Loss of performance due to incorrect usage.	Limit user access. Limit user capabilities. User training.
Sabotage.	Prevent access. Limit user access. Limit user capabilities. Prohibit unauthorized software. Accountability. Log operator commands. Maintain system configuration log.

**Table 2-4.
Synopsis of Possible Security Threats and Defenses
(continued)**

Possible Threats	Possible Defenses
Disclosure of information.	Prevent access. Limit document distribution. Limit knowledge distribution. Lock desks, cabinets, computers. Store media in locked cabinets. Degauss media to erase data. Use and maintain passwords. Clear screens and screen buffers. Limit information stored in PUB and library accounts. Provide information on a need-to-know basis. Protect all files with ACDs.

3

Managing System Users with Passwords and Logon Restrictons

This chapter describes the methods and tools available to System Managers (SM capability) and Account Managers (AM capability) for controlling system access with passwords and logon limitations, and for listing security information.

Caution

System managers should use great care to not loose or forget the system manager password. This password can be written down, sealed in a secure envelope, and locked in a safe.

Managing System Access with Passwords

Passwords are an important defense against security breaches. User passwords prevent unauthorized persons from accessing your system. Account passwords protect the information in an account from users who are not members of the account. Group passwords allow account users to work in a given group. They also serve to protect files in the group from users who are not members of it.

Tools for maintaining password security include:

- Making user passwords mandatory.
- Letting users choose their own passwords.
- Making user passwords expire.
- Encrypting passwords.
- Establishing a minimum length for passwords.

If account passwords are created, their use is required for all users. If group passwords are created, their use is required unless the group is a user's home group. Individual user passwords may be set as optional or required. Making individual user passwords required adds another level of security to the system.

Account level passwords are created and maintained by System Managers. Group level passwords are created and maintained by Account Managers and System Managers. User passwords are created and required by System and Account Managers, and can be changed by users.

Note

If files are protected by ACDs, only user passwords should be required, and neither account or group passwords, or file lockwords, should be used.

Commands Used to Create and Maintain Passwords

System Managers (SM capability) use the commands `:NEWACCT`, `:ALTACCT`, and `:PURGEACCT` to create and maintain accounts. Account passwords can be created at the time an account is created or modified.

Both Account Managers (AM capability) and System Managers use the commands `:NEWGROUP`, `:ALTGROUP`, and `:PURGEGROUP` to create and maintain groups. Group passwords can be created at the same time a group is created or modified.

Both Account and System Managers use the commands `:NEWUSER`, `:ALTUSER`, and `:PURGEUSER` to create and maintain users. User passwords can be created at the time users are created or modified.

Guidelines for Selecting Passwords

User accounts on the system must have passwords and all users share the responsibility of protecting their individual passwords to ensure that password integrity is not compromised. You will need to select a password the first time you log into the system. Follow these guidelines when selecting a password.

- Never use passwords that have anything to do with your personal life, such as a spouse or child's name.
- Never use an English word or proper name.
- Never use an English word with a number at the end.
- MPE/iX will not let you start a password with a number.
- Never use your birthday, your street address, or any other number that has anything to do with yourself.
- Never use any word spelled backwards.
- Never share passwords. When two (or more) people use the same account, the system loses its ability to hold users responsible for their actions.
- Never write Passwords down. Some of the most (in)famous penetrations have occurred because a user wrote a password on a terminal.
- Never re-use a password. This increases the probability that someone can guess the password.
- Never type a password while someone is watching. It is easy to obtain a password by observing someone type it.
- Always pick a password that has numbers and/or special characters interspersed, or use the password generator.

- Always use different passwords on different machines, but never make them the name of the machine, nor the name of the machine with a single number at the front or at the back.

Creating a New Account with a Password

To create a new account with an account password enter `:NEWACCT`, followed by the parameters: *accountname* and *managername*; `PASS=`*password*.

For example, enter:

```
:NEWACCT TECHNLGY,MGR;PASS=PROCESS
```

Modifying an Account Password

To modify an account password and make account passwords required, enter `:ALTACCT`, followed by the required parameter *accountname*; `PASS=`*password*.

For example, enter:

```
:ALTACCT TECHNLGY;PASS=MICRONS
```

Removing Account Level Passwords

To remove an account password, enter `:ALTACCT`, followed by the required parameter *accountname* and `PASS=`.

For example, enter:

```
:ALTACCT TECHNLGY;PASS=
```

Creating a New Group With Group Password

An Account Manager must be logged on to his or her account to execute the following commands. System Managers need not be logged on to the account to execute these commands, but must enter the account name when doing so.

To create a new group and group password, log on to the account and enter `:NEWGROUP` followed by the required parameter *groupname* and `PASS=`*password.*

For example, enter:

```
:NEWGROUP RCVBLS;PASS=BOOKS
```

As a System Manager creating the same group while not logged on to the account, enter the account name as well as the group name:

```
:NEWGROUP RCVBLS.ACCOUNTS;PASS=BOOKS
```

Modifying a Group Password

To modify a group password, log on to the account and enter :ALTGROUP, followed by the required parameter *groupname* and PASS=*password.*

For example, enter:

```
:ALTGROUP RCVBLS;PASS=LEDGERS
```

As a System Manager modifying the same group password while not logged on to the account, enter the account name as well as the group name:

```
:ALTGROUP RCVBLS.ACCOUNTS;PASS=LEDGERS
```

Removing Group Level Passwords

To remove a group password, log on to the account and enter :ALTGROUP, followed by the required parameter *groupname* and PASS=.

For example, enter:

```
:ALTGROUP RCVBLES;PASS=
```

As a System Manager removing the group password while not logged on to the account, enter the account name as well as the group name:

```
:ALTGROUP RCVBLS.ACCOUNTS;PASS=
```

When password prompts are required, password prompting will occur in the following situations:

- When streaming jobs from a session.
- When streaming jobs programmatically from a session.
- When issuing :STARTSESS from a session.
- When issuing :STARTSESS programmatically from a session.

Creating a New User With User Passwords

User passwords are created and modified using the MPE/iX commands :NEWUSER and :ALTUSER.

An Account Manager must be logged on to his or her account to execute the following commands. System Managers need not be logged on to the account to execute these commands, but must enter the account name when doing so.

To create a password for a new user, log on to the account and enter :NEWUSER, followed by the required parameter *username*; PASS=*password*.

For example, enter:

```
:NEWUSER MANFRED;PASS=REDBARON
```

As a System Manager creating the same user while not logged on to the account, enter the account name as well as the group name:

```
:NEWUSER MANFRED.JASTA11;PASS=REDBARON
```

User passwords assigned by Account or System Managers can be changed by the user with the :PASSWORD command.

Modifying a User Password

To modify a user password, log on to the account and enter :ALTUSER, followed by the required parameter *username*; PASS=*password*.

For example, enter:

```
:ALTUSER MANFRED;PASS=EIGHTY
```

As a System Manager modifying a user password while not logged onto the account, enter the account name as well as the group name:

```
:ALTUSER MANFRED.JASTA11;PASS=EIGHTY
```

Modifying User Passwords with :PASSWORD

Users can change their own passwords with the :PASSWORD command. To change a password, enter:

```
:PASSWORD
```

The system prompts for the required information. When using :PASSWORD, a user may not replace an existing password with exactly the same password.

Removing User Passwords

To remove a user password, log on to the account and enter :ALTUSER, followed by the required parameter *username* ;PASS=.

For example, enter:

```
:ALTUSER MANFRED;PASS=
```

As a System Manager removing the same user password while not logged on to the account, enter the account name as well as the user name:

```
:ALTUSER MANFRED.JASTA11;PASS=
```

Revising Old Passwords

Passwords that never change present a security risk to the system. Several facilities are provided which force passwords to be revised either for individual users or for all users on the system.

This section describes additional password features that are provided by the HP Security Monitor package. These features include password expiration, password aging, password encryption and enforcing of minimum length passwords for additional security.

Expiring User Passwords

System and Account Managers can cause individual user passwords to expire using standard system commands. These facilities are the `USERPASS=REQ,EXPIRED` options of the `:NEWUSER` and `:ALTUSER` commands.

The syntax for the expiration parameter is as follows:

```
:NEWUSER username [;USERPASS=(REQ or OPT)[,EXPIRED]]

:ALTUSER username [;USERPASS=(REQ or OPT)[,EXPIRED]]
```

Once a password has been expired, the user is prompted to enter a new password the next time they log onto the sytem. After the user supplies the new password, they are prompted to enter the password a second time to ensure that the intended password was entered. If the user makes a mistake when entering the new password the second time, the system prints the message `NEW PASSWORD NOT VERIFIED`, and asks the user to enter the new password again. If the user is not successful after three tries, the logon process terminates, and the user must re-start the logon process. A user will not be allowed to log on until a new password is successfully entered.

The amount of time alloted for specifying a new user password is governed by the logon timer which is configured during system startup.

Global Password Expiration

This feature allows the System Manager to activate automatic password expiration for all users who are required to have a password. To enable this option, the System Manager specifies a number of days (from 1 to 365) which determines how long all passwords will be valid.

The System Manager can specify a date (the current day is the default) for the expiration cycle to begin. The System Manager can also specify the number of days prior to the expiration on which the user is notified of the pending expiration.

If this feature is enabled, this absolute expiration date takes precedence over the password aging values described later.

Effects of Expired User Passwords

Expiration of a password has the following effects on users:

- The global expired user password function causes the expiration only of **required** user passwords, regardless of whether required at the user or account level.
- Required user passwords are marked for expiration at the beginning of the warning period. Thus, if a new user establishes a required password after the start of the warning period, that password is not affected by the forced expiration. Of course, it will be affected by the next forced expiration.
- If a user's password has expired and the user is forced to enter a new one, it cannot be the same as the one that just expired.
- When a required password expires, the new password must meet the same requirements as defined for the previous password. It must satisfy the password minimum length function, and the user password required function. (A blank password is not allowed, the password must be of a minimum length, and the password must be different from the previous one.)
- Users can replace expired passwords only during interactive logon attempts. Other types of logon attempts will fail. Users should check that UDCs programs, and job streams that include logon commands, can recover from such failures.

Password Aging

This feature allows the system manager to implement an additional level of security by requiring users to periodically change their passwords. Prompting users for new passwords after a specified period of time helps safeguard passwords against unintended disclosure and also prevents stolen passwords from remaining valid for an indefinite period of time. There are two levels of password aging:

- One is a system wide policy that establishes aging values for all users.
- The other establishes password aging values for individual users.

Under this scheme, each password has a pre-defined maximum life-span which progresses through three stages:

Valid Allows users to log on to the system.

Expired Requires user to define a new password.

Invalid At this stage, it is too late for the user to specify a new password; only the System Manager can change the password.

Graphically, the password aging for both system wide and individual user level can be shown as:

```
|----------------- maximum lifetime ------------------|

|--------- valid --------- | --------- expired ---------| -- invalid --

|<-- minimum -> <- warn -->|     <-- expiration -->     |
```

Figure 3-1. Password Aging Life Cycle

Aging values for individual users can be established only after the system wide policy is established. Once this is done, aging values for individual users can be specified as long as they don't fall outside the range established by the system wide policy. If the system wide policy is changed, aging values for any individuals which exceed the system wide range are modified to reflect the new values the next time the user logs on. Here are the password aging values which can be set:

Maximum Lifetime The maximum lifetime range is 1 to 365 days. During this period, the password is available for authentication and may be replaced.

Minimum Lifetime The minimum number of days a password must be kept before a user can replace it. This is also the minimum time a password can spend in the valid state unless the system or account manager intervenes. This value can be zero.

Expiration period The maximum number of days a user password can remain in the expired state during which the user can replace the password. This value can be zero.

Warning Period The number of days warning given to a user before their password is expired. This value can be zero.

Start date The date on which a password life cycle begins. This field is updated whenever the password is changed. During the logon sequence, the life cycle start date and the cycle time periods are compared to determine the password's current state.

Note

If password aging is enabled, all existing users on the system enter the expired state so they can choose a new password. The start date is updated at the logon time when users change their password. When a new user is created after the password aging is enabled, the start date for the user is the creation date by default.

Encrypting Passwords

To enable password encryption, select Option 1 in the Global Security Options Menu. With the feature enabled, new passwords are automatically encrypted the first time they are entered in the system. This applies to all passwords: account, group, and user. Device passwords are always encrypted, whether encryption is enabled or not.

Discussion

With password encryption turned on, a new password is automatically encrypted before it is stored in the system directory. In that way, only the person entering the password ever sees its unencrypted form.

The encryption facility is strictly one way. Even if you know the encryption algorithm, you cannot reconstruct a password in plain language from its encrypted version.

Note

The MPE/iX commands that display passwords (`:LISTUSER`, `:LISTGROUP`, and `:LISTACCT`) will not display passwords when they are in encrypted form.

MPE/iX lets you gradually convert from unencrypted to encrypted passwords by allowing both to exist side by side. The system keeps track of which passwords are encrypted and which are not. Users do not see a difference between using an encrypted or unencrypted password. As new passwords are added or old ones changed, the system encrypts them automatically.

Effects of Password Encryption

Password encryption may produce the following effects:

- Some job scheduler programs that obtain passwords directly from the directory will not work when passwords are encrypted.
- Any utility that gets passwords from the directory will not function properly if passwords are encrypted.
- Since encrypted passwords are not compatible with MPE releases prior to 5.0, you will have to remove them in order to move back to a previous release. The HP Security Monitor provides a reset facility that will remove all encrypted passwords on a system.
- When the System Manager turns on password encryption, existing passwords are not automatically encrypted. Turning password encryption on means that the next time a password is created or changed, it will be stored in an encrypted form.
- When using the STORE/RESTORE facility to backup a directory that has an associated encrypted password, only systems which are using 5.0 or later releases will retain the password information. If the backup restores to a earlier release, the Security Monitor information will not be restored and the PASSWORD field will be left blank.

Enforcing Minimum Password Lengths

MPE/iX permits passwords of from one to eight characters. The longer the password, the more difficult it is for it to be discovered by trial and error. As a security precaution, set a minimum length for all passwords in your system. The minimum length set affects all account, group, and user passwords.

To set password length, select Option 2 in the Global Security Options Menu. The default is 0 (no minimum length). When you set a new minimum length, it applies only to passwords entered after the new minimum is set. Existing passwords are not affected, and users need not change existing passwords to comply with it. If a user does enter a new password that is too short, the following message is displayed:

```
MINIMUM PASSWORD LENGTH IS X CHARACTERS LONG.  (CIERR 763)
```

where `X` is the minimum password length, and has a range of 1 to 8.

Requirements for minimum password length have the following effects:

1. The requirement for a minimum password length does not affect existing passwords, but will affect all new passwords entered or changed subsequently.
2. New or changed user passwords must satisfy all requirements for required user passwords and for required minimum password length.
3. A user who is affected by the minimum password length requirement, and who attempts to enter too short a password, will receive an error message. The user must enter a password that meets or exceeds the minimum length requirement, as specified in the error message.

Displaying Security Information

The following command is used to display the status of account attributes and security provisions for files and devices.

You can display account attributes by entering the MPE/iX command `:LISTACCT`.

```
>LISTACCT [acctset] [,listfile] [;PASS]
```

where *acctset* specifies the name of an account, *,listfile* specifies a device that will receive the output listing, and `;PASS` specifies that the password will be displayed. If an unauthorized user enters `;PASS`, asterisks (*) are displayed in place of sensitive information. The listing will include information on the password aging values if appropriate and all of the relevant details.

To list all of the attributes, including the password, of an account named `MARKETS`, enter:

```
:LISTACCT MARKETS;PASS
```

Note The MPE/iX commands that display passwords (`:LISTUSER, :LISTGROUP,` and `:LISTACCT`) will not display passwords when they are in encrypted form.

Discussion

The three listing commands are:

1. `:LISTACCT` lists account attributes.
2. `:LISTGROUP` lists group attributes.
3. `:LISTUSER` lists user attributes.

- A System Manager (SM capability) can specify any account, group, and user on the system.
- An Account Manager (AM capability) can specify any group or user in his or her logon account.
- A general user (one without SM or AM capabilities) can specify only his or her own logon account, group, and user name.
- Information about passwords can be examined according to the following rules:
  - □ The password is displayed when ;PASS is specified by the System or Account Manager in the following way:

    ```
    LISTUSER MGR;PASS
    ```

  - □ The password state (*ENCRYPTED*, REQUIRED and EXPIRED) are displayed when ; PASS and ;FORMAT=DETAIL are specified by the System or Account Manager in the following way:

    ```
    LISTUSER MGR;PASS;FORMAT=DETAIL
    ```

  - □ The password aging value is displayed when ;FORMAT=DETAIL is specified by all users in the following way:

    ```
    LISTUSER MGR;FORMAT=DETAIL
    ```

- Only System and Account Managers can use *wildcard* characters (`#, ?,` and `@`) when specifying group names. Any user can use the character `@` when specifying file names.
- Only System Managers can use wildcards when specifying account names.

Managing System Access With Account and Group Attributes

Account and group attributes that relate to system access include:

- Limiting the amount of CPU time available to users. This can be set at the account and group levels.
- Limiting the amount of session connect time available. This can be set at the account and group levels.

Limiting CPU or session connect time provides some degree of control over system utilization and, therefore, system performance.

Controlling Account and Group CPU Time Limits

The amount of time, in seconds, users can access the CPU is set with the CPU= parameter of the :NEWACCT, :ALTACCT, :NEWGROUP, and :ALTGROUP commands. The default is unlimited time.

For example, a System Manager can set the CPU time for all users in an account to one hour by entering:

```
:ALTACCT ACCOUNTS;CPU=3600
```

Controlling Account and Group Connect Time

The amount of time a session can remain connected is set with the CONNECT= parameter of the :NEWACCT, :ALTACCT, :NEWGROUP, and :ALTGROUP commands. The time is set in minutes. Default is unlimited time.

For example, an Account Manager can set the connect time for all users in an existing account to two hours by logging on to the account and entering:

```
:ALTGROUP RCVBLS;CONNECT=120
```

Managing System Access With Logon Restrictions

With one exception, the following procedures control system access by placing limitations on users when they attempt to log on to the system. The exception automatically logs an inactive session off the system.

Controlling Access With Logon UDCs

Logon UDCs (a type of User Defined Command) can be used to confine the system access capabilities of individual users within rigidly defined limits.

A logon UDC is one that executes whenever a user logs on. Each level in the system can have a logon UDC that executes at that level. System Managers create and control system logon UDCs, Account Managers create and control account logon UDCs, and users create and control their own logon UDCs. System and Account Managers also can control user UDCs.

You might use the system level logon UDC to prevent users from accessing MPE/iX commands. For example, a logon UDC with NOBREAK runs an application program then automatically logs users off the system as soon as they exit the program. In this case, users have access only to the application program, but not to the MPE/iX command interpreter or other system facilities.

Creating a UDC

To create a UDC, type the commands you wish to use in a text file, then catalog the file with the `:SETCATALOG` command. If a UDC is to be a logon UDC, declare it as such when you create it. Set the UDC level (system, account, or user) at the time you catalog it. The *MPE/iX Commands Reference Manual* (32033-90006) describes in detail how to create and catalog UDCs.

4

Protecting Your System with Access Control Definitions (ACDs)

Access Control Definitions (ACDs)

MPE/iX file system access can be controlled by using access control definitions (ACDs). You can use an ACD to specify permissions and restrictions for access to a file. In addition, ACDs allow you to secure logical devices, device names, and device classes. ACD security replaces all standard file system security that may be in effect for that file or device.

Note

ACDs are the preferred method of controlling access to files, hierarchical directories, and devices in systems that maintain a C2 level of trust. ACDs are automatically assigned to directories and to files existing in directories.

What is an ACD?

ACDs are ordered lists of pairs that define security for a user or group of users. The pairs are made up of access permissions and user specifications that control access to **objects**. Objects are passive entities that contain or receive information, such as files, directories, and devices. Each entry in the ACD specifies object access permissions granted to a specific user or group of users. In addition to being granted access to an object protected by an ACD, users can also be granted access to read the ACD itself.

ACDs can be applied to any MPE/iX files or hierarchical directories using the `ALTSEC` command. This command was enhanced to support directories. If a file has an ACD, this method of specifying access to the file takes precedence over other security features.

How do ACDs work

When you attempt to access a file, or other object protected by the file system security facilities, the system checks access permissions in the following order of precedence:

1. Do you have SM capability? If so, you are granted all access to the file.
2. Do you have AM capability and does your GID match the GID of the file? If so, you are granted all access to the file.
3. Are you the owner of the file (your UID matches the UID of the file)?
   a. If there is no ACD associated with the file, you are given all access permissions to the file and the checking ends.
   b. If there is an ACD associated with the file and there is no `$OWNER` entry, you are given all access permissions to the file and the checking ends.

c. If there is an ACD associated with the file and that ACD contains the $OWNER entry, you are restricted to the access permissions assigned to $OWNER. (Since you are the file owner, you can always modify the ACD if you need more access permissions than provided by the $OWNER entry.)

If you are not the owner of the file, the system performs the check described in step 4.

4. Is there an ACD assigned to the file? If there is no ACD assigned to the file, the system performs the checking described in step 5. If there is an ACD, the system performs the checking in the following order (from more specific to less specific):

   a. Does your UID match a specific user name entry (for example, `ALEX.TECHNLGY`). If so, you are granted the access permissions assigned to that entry unless a `$GROUP_MASK` entry exists. If the `$GROUP_MASK` entry exists, the matching entry is combined with `$GROUP_MASK` to determine the actual access permissions. No further checking is performed.
   b. Does your GID match the GID of the file? If so, and a `$GROUP` entry exists, you are granted the access permissions assigned to that entry unless a `$GROUP_MASK` entry exists. If the `$GROUP_MASK` entry exists, the resulting access permissions are only those that are in both the `$GROUP` and the `$GROUP_MASK` entries. No further checking is performed.

      If you match the `$GROUP` entry and your GID matches the account portion of an *@.account* entry, you are granted the access permissions assigned to either ACD entry prior to `$GROUP_MASK` evaluation.
   c. Does your GID match the *account* portion of an *@.account* entry? If so, you are granted the access permissions assigned to that entry unless a `$GROUP_MASK` entry exists. If the `$GROUP_MASK` entry exists, the resulting access permissions are only those that are in both the `$GROUP` and the `$GROUP_MASK` entries. No further checking is performed.
   d. Does an `@.@` entry exist? If so, you are granted the access permissions assigned to that entry. No further checking is performed.
   e. If your name is not found (or if the access mode assigned to you is NONE), you are granted no access to the file, and no further checking is performed.

5. If there is no ACD, the system uses the file access matrix to check for access permissions.

Access modes

ACD pairs control the ability to access and change MPE files, hierarchical directories, and the files within them. MPE/iX has enhanced the `ALTSEC` command to support access to directories. The available ACD access modes are as follows:

FILES AND DEVICES

R	Read a file.
W	Write to a file.
L	Lock a file.
A	Append to a file.
X	Execute a file.

DIRECTORIES

CD	Create directory entries.
DD	Delete directory entries.
RD	Read directory entries.
TD	Traverse directory entries.
RACD	Copy or read the ACD permission.
NONE	Deny access.

The NONE and RACD access modes are available only through an ACD.

Users need appropriate access attributes to access a directory and its contents. For example, the owner of a directory can grant *create directory entries (CD)* access to other users. Users can only create files or other directories within a directory if they have CD access to the directory.

RD entries access and TD entries access differ as follows. If a user wants to use `LISTFILE` to list the files in a directory, the user needs RD entries permission for that directory. But, if a user wants to access a file such as `/users/jeff/address`, the user needs to have TD entries permission for all the directories in the path; that is, `/`, `users`, and `jeff` in this case.

By default, all users can read the contents of and traverse the root directory, all MPE accounts, and all MPE groups. However, to read or write the contents of a file, you must have the appropriate access permission to open the file itself.

Because the root, accounts, and MPE groups are special types of directories on MPE/iX, you cannot control access to them using ACDs. You cannot apply TD, DD, CD, or RD to MPE groups or accounts. You need to use existing mechanisms. For example, use the ALTGROUP command to change save access permissions for MPE groups.

The *userspecs* part of an ACD pair specifies one user or a group of users assigned the access modes specified in *modes* part of the same pair. A user is specified as a fully qualified user name in the form *username.accountname*. For example, JOAN.FINANCE specifies the user JOAN in the account FINANCE.

A wildcard character (@) can be used in place of the user name or both the user name and the account name to specify a group of users. For example, @.FINANCE specifies all users in the account FINANCE, and @.@ specifies all users in all the accounts on the system.

A user who is not specified in any ACD pairs or whose assigned access mode is NONE has no access to the file.

For example, you could define an ACD as follows:

```
ACD = (R,W:MGR.ACCTING, PETE.TECHNLGY; R:@.PAYROLL; A:@.@)
```

If this ACD were assigned to a file, it would be interpreted in the following manner:

- The users MGR.ACCTING and PETE.TECHNLGY have READ and WRITE access to the file but do not have APPEND, EXECUTE, or RACD access.
- All users in the PAYROLL account have READ access to the file but do not have WRITE, APPEND, EXECUTE, or RACD access.
- All users on the system have APPEND access to the file but do not have READ, WRITE, EXECUTE, or RACD access.
- A file owner has full access to the file.

You use the ALTSEC command to alter access modes for files, hierarchical directories, logical devices, or device classes. For more information about ACD access modes, refer to the ALTSEC command in Chapter 2 of the *MPE/iX Reference Supplement* (32650-90353).

User specifications

Beginning with MPE/iX Release 4.5, the MPE/iX access control definition (ACD) facility provides three new user specifications. In place of specifying a user (*user.account*) or set of users (@*.account*) in a file or directory ACD, you can also use the following designators:

$OWNER Specifies the file owner. The file owner is granted the access permissions specified by $OWNER. A user is a file owner if the user's UID (in the form *user.account*) matches the UID of the file. The owner can be changed from the initial creator programmatically.

$GROUP Specifies the file group members of the file or directory. If the user's GID (in the form *account*) matches the GID of the file, the user is granted the access permission assigned to `$GROUP`.

`$GROUP_MASK` Restricts all ACD entries except for `$OWNER` and @.@. In this case, if a user matches a *user.account* entry, `$GROUP` entry, or *@.account* entry, the matching entry is granted the access if it appears in both `$GROUP` and `$GROUP_MASK`. An ACD with a `$GROUP_MASK` entry must also have a `$GROUP` entry. `$GROUP_MASK` is provided to integrate the POSIX definition of security with the more robust security provided by MPE/iX ACDs.

These new user specifications modify the manner in which the file system checks access permissions when an ACD is associated with a file.

Required ACDs

Prior to release 4.5, the MPE/iX ACD facility provided an optional security facility to replace MPE/iX standard file system security features. Beginning with release 4.5, ACDs are required on the following system objects:

- All hierarchical directories
- All files under hierarchical directories
- All files directly under MPE/iX groups where the file GID does not match the GID of the account and group in which the file is located.

Because ACDs are now required in some cases, it becomes increasingly important that you understand the MPE/iX ACD facility. This section provides a summary of the enhancements made to the MPE/iX ACD facility. This section either supplements or replaces the descriptions of ACDs found the *Controlling System Activity* (32650-90155).

HFS Object creation

Creating an object, which is creating an entry for a file or directory within a directory, requires that a process have traverse directory (TD) and create directory (CD) access to the object's parent directory and SF capability. For an MPE group, SAVE access is equivalent to create directory access (see "SAVE access in MPE groups").

Users with SM capability can create files and directories anywhere on the system. Users without SM capability can create files and directories outside their logon account in any directory that they can traverse and to which they have been granted create directory access.

HFS Object deletion

To delete a file or subdirectory from a directory, you must have DD access to the directory. For files in MPE groups, you only need WRITE access to the file. For directories in MPE groups, you only need SAVE access to the MPE group.

HFS File renaming

Any user with the proper access can rename a file. To rename a file, you must have both CD and DD access. DD is required to delete the old entry from the directory where the file resides, and CD is required to create the new directory entry.

You can rename a file from one directory to another if you have DD access to the directory in which the file is located and CD access to the directory where you want the renamed file to reside.

Users with SM capability can rename files anywhere on the system. To rename a file from an MPE group in one account to an MPE group in another account, you must have SM capability.

If you rename a file that does not have an ACD from an MPE group to a directory that is not an MPE group, an ACD is automatically generated for it. Otherwise, the file would no longer be protected by the file access matrix.

If you rename a file (that does not have an ACD) from an MPE group to another MPE group outside the original account, an ACD is automatically generated for it. The file's GID would no longer match the parent group's GID and would not be protected by the file access matrix.

File owner

A file (or directory) owner has complete access to the file unless the user is restricted by a `$OWNER` ACD entry. Now that there is a `$OWNER` ACD entry, you can restrict the file access of the file owner.

For example, `MGR.PAYROLL` is the creator (owner) of the file `MYFILE`. On Releases 3.0 and 4.0, the owner's access cannot be restricted by an ACD or the file access matrix. So on Release 3.0 and 4.0 systems, `MGR.PAYROLL` still has all the access permissions on this file even if an ACD pair specifies only read permission (`R:MGR.PAYROLL`). As of Release 4.5, the access of the owner can be restricted by using the `$OWNER` ACD entry. Assigning `R:$OWNER` restricts the owner to having read permission only.

Appropriate Privilege

Appropriate privilege means that the user has sufficient capabilities to perform an operation even if the user is not explicitly granted the necessary access. The user's capabilities grant the correct access to the directory or file.

Appropriate privilege does not override file lockwords, privileged files, privileged file codes, or write-protected files.

System manager capability

Having SM capability provides appropriate privilege and allows the system manager (or those having SM) to override the file access matrix or ACD on any file or directory.

Users with SM capability can create files and directories anywhere on the system. Users with SM capability can also rename files anywhere on the system. To rename a file from an MPE group in one account to an MPE group in another account, you must have SM capability.

Account manager capability

If all objects in an account have the same GID, the traditional MPE model remains in effect. A user having AM capability for the account can access all of the files and directories within the account.

It is possible for objects within an account to have different GIDs if, for example, files are renamed or if the GID is changed programmatically. In this case, having AM capability will not be sufficient privilege to gain access to those files. The GID of the user with AM has to match the GID of the file or directory to allow access to it.

Execute (X) Access

The MPE/iX shell does not provide a way to distinguish files containing executable scripts from other files. However, the POSIX standard requires that file permission bits should be checked to verify that execute access has been granted to at least one of the file classes.

When ALL access would normally be granted to a user, X access is handled as a special case. Users with appropriate privilege are granted X access only if the file has an executable file code (PROG, SL, NMPRG, or NMXL), if the file access matrix assigns X access to the user, or if the file has an ACD that assigns X access to at least one user.

The file creator is granted X access only if the `$OWNER` ACD entry grants X access. If the `$OWNER` entry does not exist, the file creator is granted X access if the file has an executable file code or at least one user is granted X access by the file access matrix or an ACD.

A RELEASEd file grants X access only if it has an executable file code.

Users with appropriate privilege still get X access to files with executable file codes. X is also used to grant `STREAM` access to `JOB` files. Users with appropriate privilege can still stream these files because they have R access to the files.

User Identification

Users on MPE/iX are now identified by a user ID (UID). The UID is a string (in the form *user.account*) with a corresponding integer value. Each MPE account has a group ID (GID) associated with it. The GID is a string (in the form *account*) and also has a numerical value assigned to it. UIDs and GIDs were added to file and process structures to more easily identify object owners and file sharing groups, respectively.

In addition to the UIDs and GIDs, users are identified as follows:

Table 4-1. User Categories

Category	Conditions
File Owner	The user whose UID matches the object's UID (also called *user.account* or `$OWNER` in ACDs). By default, when a user creates a file or directory it is assigned the same UID as that user.
File Group Member	Any user whose GID matches the GID of the object (also called *@.account* or `$GROUP` in ACDs). By default, all members of an account are assigned the same GID. This *group* is a new file sharing concept that should be distinguished from MPE groups (that is, group directories). By default, when a user creates a file or directory, it is assigned the parent directory's GID.

SAVE access in MPE groups

Create directory entries (CD) access and delete directory entries (DD) access to all MPE groups is governed by appropriate privileges or SAVE access. (A complete definition of appropriate privilege appears later in this chapter.) SAVE access for an MPE group implies CD and DD permission for directory entries. That is, a user can create or delete a directory in an MPE group if the group grants SAVE access to the user. However, you still need write access to a file to be able to delete it from an MPE group.

CWD and File Security

You can now change the current working directory (CWD) to any directory (including an MPE account, an MPE group, the root directory, or an HFS directory) as long as you have TD access to the directories in the path to the directory. This means that you can change your CWD to any MPE group on the system because all users have RD and TD access to the root directory, all accounts, and all groups, by default.

It is important to note that changing your CWD to a new MPE group (using the `CHDIR` command) does not make you a GU user of the new group. GU is based on your logon group and account; this can only be changed using `CHGROUP`. If you attempt to access a file in the new group, you may not be able to access it. If the new group is in your logon account, you are allowed account level privileges in the new group. If the new group is not in your logon account, you are allowed the access privileges given to any user. No password check is done when you change your CWD. This is unlike `CHGROUP` which does a password check.

The Maximum File Protection Option

This Security Monitor feature provides security protection for objects at the time they are created.

This can be accomplished:

1. By enforcing a restrictive default access control on newly created files.
2. By requiring the user to explicitly specify the desired access controls on the file when requesting its creation.

In either case, absolutely no unauthorized access to newly created files is allowed.

When set, the Maximum Protection feature enforces restrictive access to newly created files. The standards for access to newly created files are:

1. If the feature is enabled and there is no ACD attached to the file, the default ACD is set to (RACD:@.@).
2. If the feature is enabled and there is an ACD present, the ACD is used to mediate access.
3. If the feature is not enabled and there is no ACD present, the normal file access matrix is used in the default fashion.

Note

(RACD:@.@)means the CREATOR of the file and processes with the appropriate privilege (either AM or SM) will be able to access the file. All other processes will only be able to read the ACD. The CREATOR can always modify the ACD afterwards.

ACD examples

You assign ACDs using the ALTSEC command. In addition, files created in hierarchical directories and hierarchical directories themselves are automatically assigned ACDs.

Following is an example of an ACD that could be assigned to a text file:

```
NONE:JIM.DOE,@.ACCTING;R,W,X,L:@.PAYROLL;R:@.@
```

The ACD pairs in this example set up the following access controls on the text file:

- Deny JIM.DOE and all users in the ACCTING account access to the file.
- Allow read, write, execute, and lock access to users in the PAYROLL account.
- Allow read access to everyone else.

Notice that in cases of contradictions, the most specific ACD pair is assigned. So even though all users are assigned read access (R:@.@), JIM.DOE cannot access the file because he is specifically assigned no access (NONE:JIM.DOE).

If the ACD in the above example had a $GROUP_MASK entry (for example, rx:$GROUP_MASK), then the users in the PAYROLL account would only have read and execute access. The entire ACD would read as follows:

```
NONE:JIM.DOE,@.ACCTING;R,W,X,L:@.PAYROLL;R:@.@;rx:$GROUP_MASK
```

An example of an ACD for an HFS directory (dir1) follows:

```
CD,DD,RD,TD,RACD:@.ACCT;TD:@.@
```

The ACD pairs in this example set up the following access controls on dir1:

- Allow all users in the ACCT account the ability to create, delete, and read directory entries in dir1, to traverse dir1, and to read the ACDs
- Allow everyone else the ability to traverse dir1 only.

Tasks Involving System Security

The following sections describe tasks relating to system security such as listing ACDs, assigning ACDs, changing ACDs, and copying ACDs.

Listing ACDs

Use the -2 *listfile* option of the LISTFILE or LISTF commands to list ACD information associated with a file. Any user on a system can use these commands to determine if a file has an ACD. In order to view the contents of an ACD, you must be either an owner of the file or be a user granted RACD access to that file.

Use the SHOWDEV command to list ACD information associated with a logical device, device name, or device class. Only a system manager and users granted RACD access can view the contents of a device ACD.

If you are the user DENNIS.ADMIN and you want to view the contents of ACDs for all files in group and account DEV.ENGR, enter:

```
LISTFILE @.DEV.ENGR,-2
```

The screen displays:

```
ACCOUNT =  ENGR      GROUP=DEV

FILENAME        ------------ACD ENTRIES-----------

RLDSPR               NO ACDS
QUEXINK              TEST.ENGR      : X,A,L
                     DENNIS.ADMIN   : RACD
                     HENRY.MFG      : NONE
                     THO.ENGR       : W
                     TOM.ENGR       : R,W
BFDFILE              NO ACD ACCESS
```

In the previous example, you (DENNIS.ADMIN) have permission to view the ACD associated with QEXINK. While the file BFDFILE has an ACD associated with it, you do not have permission to view its ACD contents.

The file RLDSPR has no ACD, so access to this file is determined through standard file system security features. Enter LISTFILE RLDSPR, -3 to obtain security provisions in effect for RLDSPR.

Listing ACDs for directories and files in directories

Because ACDs supersede other security mechanisms, it is useful to be able to determine whether or not an HFS directory or file has an ACD assigned to it and, if so, what it is. Any directories or files residing outside of traditional MPE groups are automatically assigned ACDs when they are created. You can list ACDs by using the LISTFILE command with the -2 (also called ACD) option.

The following example shows how to list the ACD associated with the directory called letters. Notice that the user named JONES in the OFFICE account has RD (read directory entries) access to the letters directory. All other users on the system have both RD and TD (traverse directory entries) access to letters.

```
listfile /dir0/letters,-2

 PATH=/dir0/

 ------------ACD ENTRIES-------------- FILENAME

     JONES.OFFICE       : RD           letters/
     @.@                : RD,TD
```

In the next example, the directory GRP is assigned the default ACD. All users can read the ACD assigned to the directory. Only the creator and the system manager can change it. Also, note that -2 is replaced with the textual equivalent ACD.

```
listfile /OFFICE/GRP,ACD

 PATH=/OFFICE/

 ------------ACD ENTRIES-------------- FILENAME

 @.@                : RACD             GRP/
```

In the next example, the file assets has an ACD assigned to it. The ACD is listed from the most specific (such as a particular user in a particular account) to the least specific (all other users in all other accounts). User ZONIS in the OFFICE account has R (read) access to the file assets. Other users in the OFFICE account have both R and W (write) access to the file. And all other users in other accounts have R, W, and X (execute) access to the file.

```
listfile /OFFICE/GRP/assets,-2

 PATH=/OFFICE/GRP/

 ------------ACD ENTRIES-------------- FILENAME

ZONIS.OFFICE      : R            assets
@.OFFICE          : R,W
@.@               : R,W,X
```

The next example shows how you can list the ACDs for all of the files in the `GRP` directory. It shows the ACDs on the file `assets` as in the previous example and lists the ACDs on the other two files in the directory.

```
listfile /OFFICE/GRP/@,-2

 PATH=/OFFICE/GRP/

------------ACD ENTRIES------------ FILENAME

ZONIS.OFFICE      : R               assets
@.OFFICE          : R,W
@.@               : R,W,X
ZONIS.OFFICE      : R               bills
WILKE.OFFICE      : R,W
@.@               : R,W,X
SMITH.OFFICE      : R               goods
@.OFFICE          : R,W,X
```

Changing access to HFS files and directories

Because access to MPE/iX files and hierarchical directories is controlled by ACDs, system users may want to change the defaults assigned when files or directories are created.

For the purpose of selectively restricting access to files with ACDs, users can be classified into three groups:

- Individual users
- Specific groups of users
- All other users

Creating ACDs

Use the NEWACD option of the ALTSEC command to create an ACD and assign it to a file or device. You must be an owner of a file to create and assign an ACD to that file. Only a system manager can assign ACDs to logical devices, device names, and device classes.

You can assign ACD pairs to the new ACD either from within the command line or by referencing a file that contains one or more ACD pairs.

To create an ACD and assign it to the file PROGNAME, enter:

```
ALTSEC PROGNAME;NEWACD=(X:@.@;W:@.ACCT)
```

This ACD grants all users on the system EXECUTE access to PROGNAME, but only users in account ACCT can write to it.

The following example performs the same action as the last example by referencing a file that contains ACD pairs:

```
ALTSEC PROGNAME;NEWACD=^ACDFILE
```

In the previous example, the ACD pairs X:@.@ and W:@.ACCT are located in the text file ACDFILE. ACD pairs are separated by semicolons.

To create an ACD that prevents any user except OPERATOR.SYS and the system manager from accessing LDEV 7 (a tape drive), enter:

```
ALTSEC 7,LDEV;NEWACD=(R,W:OPERATOR.SYS)
```

Some access modes are not applicable to certain devices. For example, it makes no sense to execute or append a tape drive. Access modes not applicable to a device can be assigned but are ignored.

Refer to the *MPE/iX Commands Reference Manual Volumes 1 and 2* (32650-90003 and 32650-90364) for further information about the ALTSEC command.

Assigning ACDs

For example, you may want to assign ACD permissions to restrict access to a sensitive file so that only you and your manager can read it. You may also want to restrict access to a sensitive directory so that only certain members of a group can create files in it.

Use the ALTSEC command to change access permissions to a file or hierarchical directory. System managers can assign ACDs on any file or directory in the system. They must supply the lockword for any lockword-protected files before they can assign an ACD, however. Once the file has an ACD, the ACD supersedes the lockword.

You can use the ADDPAIR option with the ALTSEC command to add ACD pairs to an object that already has an ACD. (You must use the NEWACD option to assign ACDs to files having no ACDs.)

For example, to assign a new ACD that gives all users on the system total access to the file NUMBERS:

```
:ALTSEC NUMBERS;NEWACD=(R,W,L,A,X,RACD:@.@)
```

The file SUMMARY has an ACD (RACD:@.@). You want to grant read and write access to users in your account:

```
:ALTSEC SUMMARY;ADDPAIR=(W,R:@.ACCT)
```

Adding an ACD Pair

Use the ADDPAIR parameter of the ALTSEC command to add an ACD pair to an ACD.

To add a new ACD pair that grants the user ENGR.LAB the access modes READ, WRITE, LOCK, APPEND, EXECUTE, and RACD to the file PROGNAME, enter:

```
ALTSEC PROGNAME;ADDPAIR=(R,W,L,A,X,RACD:ENGR.LAB)
```

Note

ACDs cannot be used to protect Image SQL files because they have their own protection.

Replacing an ACD Pair

Use the REPPAIR parameter of the ALTSEC command to replace an existing ACD pair with a new ACD pair.

To replace the access permissions previously assigned to the user ENGR.LAB with READ access to the file PROGNAME, enter:

```
ALTSEC PROGNAME;REPPAIR=(R:ENGR.LAB)
```

Replacing ACDs

You can replace the current ACD by using the REPACD option with the ALTSEC command.

All users in the MKTG account currently have RD and TD access to the directory van. The users can only move through van and read the names of files in it. Instead, you want to grant all users in MKTG greater access to the contents of the directory. You want them to be able to create directory entries, delete directory entries, read directory entries, traverse directory entries, and to be able to read the ACD.

For example,

```
:ALTSEC ./van;REPACD=(CD,DD,RD,TD,RACD:@.MKTG)
```

This option is useful when you want to change the default ACDs assigned to HFS directories and to files outside of MPE groups.

Modifying ACDs

Once an ACD is assigned to a file or device, you can modify the contents of the ACD by adding, deleting, or replacing ACD pairs. You must be an owner of a file in order to modify its ACD. Only a system manager can modify ACDs assigned to logical devices, device names, and device classes.

Deleting ACDs

Use the DELACD parameter of ALTSEC to delete an ACD assigned to a file or device. You must be an owner of a file in order to delete an ACD from that file. Only a system manager can delete ACDs from logical devices, device names, and device classes.

To eliminate any ACD that may be in effect for device class LP, enter:

```
ALTSEC LP,DEVCLASS;DELACD
```

Deleting an ACD Pair

Use the DELPAIR parameter of the ALTSEC command to delete a user name from an ACD. All other user names are unaffected.

To delete from the ACD assigned to PROGNAME only the ACD pair where the *userspecs* part exactly matches @.@, enter:

```
ALTSEC PROGNAME;DELPAIR=(@.@)
```

Deleting Optional ACDs

You can only delete optional ACDs on files in MPE groups that can be protected by the file access matrix.

Users in the ACCT account have read access to the file /ACCT/PUB/dir1/summary and all other users have read ACD access to the file (R:@.ACCT;RACD:@.@). If you decide that the users in ACCT should no longer have read access to the file, you can delete previously assigned ACD pairs (but you cannot delete the entire ACD):

```
:ALTSEC /ACCT/PUB/dir1/summary;DELPAIR=(@.ACCT)
```

The above example deletes read access to file summary for all users in ACCT but still allows all users (including those in ACCT) RACD access to the file.

You try to specify the following command to delete the ACD pair that matches @.@, which is the only ACD pair left on the file:

```
:ALTSEC /ACCT/PUB/dir1/summary;DELPAIR=(@.@)
```

Because this file is located in an HFS directory, it is required to have ACDs and cannot be protected by the file access matrix. You receive an error message and the ACD will not be deleted:

```
Cannot delete ACDs from objects where file matrix security
does not apply. (CIERR 7330)
```

If the file REPORT is a file in an MPE group, its GID matches the GID of its parent group, and its ACD is not required, you can use the following command to delete all ACD pairs:

```
:ALTSEC REPORT;DELACD
```

Copying ACDs

Use the COPYACD parameter of the ALTSEC command to copy an ACD from a source file to a target file or device. In order to copy an ACD, you must be an owner of the source file or a user granted RACD access to the source file. In addition, you must be an owner of the target file.

To copy the ACD from the file PROGNAME to the file NEWFILE, enter:

```
ALTSEC NEWFILE;COPYACD=PROGNAME
```

Copying ACD Pairs

You can copy ACD pairs from one file to another or from one directory to another. This is particularly useful if you assign a complex set of ACDs to one file or directory and you want to assign the same set to another file or directory.

Note

You can only copy an ACD from one file to another or from one directory to another. You can't copy an ACD from a directory to a file or vice versa.

For example, you can copy the ACD from directory dir1 to another directory dir2:

```
:ALTSEC ./dir2/;COPYACD=./dir1/
```

You can also copy ACDs between devices. The following example copies the ACD associated with ldev 5 to all devices in the device class TERM:

```
:ALTSEC TERM,DEVCLASS;COPYACD=5,LDEV
```

Copying Files That Have ACDs

In order to use the COPY command to copy a file that has an ACD, you must be either an owner of the source file or have READ access and RACD to the source file. In order to use the FCOPY command to copy a file, you must either be an owner of the source file or have both READ and RACD access to the source file or use the ;NOACD option of FCOPY.

The ACD of the source file is also copied to the target file. The user who copies the source file becomes the creator of the target file (and, therefore, an owner of the ACD).

In order to use the STORE or RESTORE commands to back up or restore a file that has an ACD, you must be either:

- An owner of the file
- A user who has both READ and RACD access to the file
- A user who has operator (OP) capability

If you are none of these, any attempt to either store or restore a file that has an ACD results in an error unless you specify ;NOACD.

The STORE, RESTORE, and FCOPY commands each have an optional parameter (;NOACD) that enables you to remove the ACD from a target file, removing all security restrictions in effect for the target file. When an ACD is removed from a file, standard file system security restrictions are imposed.

5

Protecting Files with File Access Restrictions and Lockwords.

File System Security Features

The account structure contains two important, standard file system security features: file access restrictions, and lockwords.

Restricting File Access

Associated with each account, group, and individual file is a list of file access restrictions. Access restrictions apply to disk files only. Their restrictions are based on the following:

- File access modes, such as reading, writing, saving, executing, locking, and appending.
- User types, such as account librarians, group librarians, and account members for whom certain access modes are allowed.

The access restrictions for any file describe who can access it and in what manner.

Access Modes

Table 5-1 lists file access modes, the codes used to reference them, and their meanings.

Table 5-1. File Access Modes

Access Modes	Mnemonic Code	Meaning
READ	R	Allows users to read files.
LOCK	L	Permits a user to prevent concurrent access to a file. Specifically, it permits the use of the `FLOCK` and `FUNLOCK` intrinsics, and the exclusive-access option of the `HPFOPEN` and `FOPEN` intrinsics, all described in the *MPE/iX Intrinsics Reference Manual* (32650-90028).
APPEND	A	Allows users to add information and disk extents to files, but prohibits them from altering or deleting information already written. This access mode implicitly allows the LOCK (L) access modes described above.
WRITE	W	Allows users general writing access, permitting them to add, delete, or change any information in files. This includes removing entire files from the system with the `PURGE` command. WRITE (W) access also implicitly allows the LOCK (L) and APPEND (A) access modes described previously.
SAVE	S	Allows users to declare files within a group as permanent, and to rename such files. This includes the ability to create new permanent files with the `BUILD` command.
EXECUTE	X	Allows users to run programs stated in files with the `RUN` command or the `CREATE` and `CREATEPROCESS` intrinsics.

User Types

Table 5-2 lists user types, the codes used to reference them, and their complete descriptions.

Table 5-2. User Types

User Type	Mnemonic Code	Meaning
Any user	ANY	Any user defined in the system. This includes all categories defined below.
Account librarian user	AL	User with account librarian capability, who can manage files within the account which may include more than one group.
Group librarian user	GL	User with group librarian capability, who can manage certain files within a home group only.
Creating user	CR	The user who created this file.
Group user	GU	Any user allowed to access this group as the logon or home group, including all GL users applicable to this group.
Account member	AC	Any user authorized access to the system under this account. This includes all AL, GU, and CR users under this account.

Users with system manager or account manager capability bypass the standard file access restrictions. A system manager has unlimited access to any file in the system, but can save files only in the system manager's own account. An account manager has unlimited access to any file in the account, except one with a negative file code. The account manager must have privileged mode (PM) capability to access a file with a negative file code.

A file's group and account as well as your capabilities determine whether you have access to the file. For example, group librarian capability gives you special access to files in your home group. You do not have special access to files in other groups.

Specifying File Access Restrictions

When a user tries to access a file, the system checks the account-level, group-level, and file-level file access restrictions. Those restrictions must give the user access rights at all three levels. If the user fails to pass the security check at any level, the system denies the user access to the file.

You set account file access restrictions when you create an account. You set group file access restrictions when you create a group. As the creator of a file, you can change its file-level access restrictions with the ALTSEC command.

When you specify file access restrictions at a certain level, you list the file access modes available to each type of user. This listing has a special format. For example, at the account level, you might assign

READ and EXECUTE access to any user and APPEND, WRITE, and LOCK access only to account users. These sample file security provisions have the following format:

```
(R,X:ANY;A,W,L:AC)
```

In this example, READ and EXECUTE access are permitted to any user. APPEND, WRITE, and LOCK access are permitted to account members only.

Account-Level File Security

The system manager sets the access restrictions that apply to all files within a given account when creating the account. A system manager can change the initial restrictions at any time (with the `ALTACCT` command). For more information, refer to "System Manager Tasks," described later in this chapter.

At the account level, the system recognizes two user types and five access modes. You can assign the access modes to the user types in any way you choose. The account-level user types are:

- Any user (ANY)
- Account member (AC)

The five account level access modes are:

- READ (R)
- LOCK (L)
- APPEND (A)
- WRITE (W)
- EXECUTE (X)

Refer to Table 5-1 for access mode descriptions and to Table 5-2 for user type descriptions.

If you do not explicitly state file access restrictions for an account, the system assigns the following default restrictions:

- For the `SYS` account, READ and EXECUTE access are permitted to all users. APPEND, WRITE, and LOCK access are limited to account members. Symbolically, these access restrictions are expressed as follows: (R,X:ANY;A,W,L:AC).
- For all other accounts, READ, APPEND, WRITE, LOCK, and EXECUTE access are limited to account members (R,A,W,L,X:AC).

Group-Level Security

The account manager sets the file access restrictions that apply to all files within a group when creating the group. They can be equal to or more restrictive than the provisions specified at the account level. The group's file access restrictions can also be less restrictive than those of the account; such provisions effectively equate the group restrictions with the account restrictions, because a user who fails a security check at the account level is denied access at that point. The account manager can change initial group file access restrictions at any time.

At the group level, the system recognizes five user types and six access modes. You can assign the access modes to the user types in any combination.

The five group-level user types are:

- Any user (ANY)
- Account librarian (AL)
- Group librarian (GL)
- Group user (GU)
- Account member (AC)

The group level file access modes are:

- READ (R)
- LOCK (L)
- APPEND (A)
- WRITE (W)
- SAVE (S)
- EXECUTE (X)

Refer to Table 5-1 for access mode descriptions and to Table 5-2 for user type descriptions.

If you do not specify group file access restrictions, the following default restrictions apply:

- For a public group (named `PUB`) whose files are normally accessible in some way by all users within the account, READ and EXECUTE access are permitted to any user; APPEND, WRITE, SAVE, and LOCK access are limited to account librarian users and group users (including group librarians) (R,X:ANY;A,W,S,L:AL,GU).
- For a public group (named `PUB`) of an account (named `SYS`), the following default restrictions apply: (R,X,L:ANY;W,A,S:AL,GU).
- For all other groups in the account, READ, APPEND, WRITE, SAVE, LOCK, and EXECUTE access are limited to group users (R,A,W,S,L,X:GU).

File-Level Security

When you create a file, it has the default file-level security provisions assigned by MPE and the provisions assigned by the account and the group to which it belongs. Only the creator of a file may use the ACCESS= option of ALTSEC on a file. An Account Manager or System Manager can change the file-level security provision with the ALTSEC command by adding an ACD or changing and ACD. All access modes and all user types apply at the file level. Refer to Table 5-1 and Table 5-2 for their descriptions.

If no security provisions are explicitly specified by the creating user, READ, APPEND, WRITE, LOCK, and EXECUTE access are permitted to all users (R,A,W,L,X:ANY), for all files, by default.

Default File Access Restrictions

Because the total security for a file always depends on security at all three levels, a file not explicitly protected from a certain access mode may benefit from the default protection at a higher level. For example, the default access restrictions at the file level allow the file to be read by any user, but the restrictions at the group level allow access only to group users. Thus, the file can be read only by a group user. In summary, the default file access restrictions at the account, group, and file levels combine to result in overall default file access restrictions as shown in Table 5-3.

Table 5-3. Default File Access Restrictions

File	File Reference	Access Permitted	Save Access To Group
Any file in public group of system account	*filename.* PUB.SYS	(R,X:ANY; W:AL, GU)	AL, GU
Any file in any group in system account	*filename. groupname*.SYS	(R,W,X:GU)	GU
Any file in public group of any account	*filename.* PUB *accountname*	(R, X:AC; W:AL, GU)	AL, GU
Any file in any group in any account	*filename.groupname. accountname*	(R,W,X:GU)	GU

In other words, when the default security provisions are in force at all levels, the standard user with default user attributes, has:

- Unlimited access (in all modes) to all files in the logon group and the home group.
- READ and EXECUTE access (only) to all files in the PUB group of the individual's account, and in the SYS account's PUB group.

Lockwords

Lockwords act as passwords for files, providing additional security beyond those provided by capabilities and file access restrictions. The creator of a file can assign a lockword with the `FILE`, `BUILD`, or `RENAME` command or with the `FOPEN` intrinsic.

If a file has a lockword, you must supply it before you can access that file. If you are a system manager (with SM capability) or account manager (with AM capability), you can displaylity), you can display file lockwords with the `LISTFILE` or `LISTF` comman file lockwords with the `LISTFILE` or `LISTF` commands, documented in *MPE/iX Commands Reference Manual Volumes 1 and 2* (32650-90003 and 32650-90364).

Figure 5-1 illustrates how lockwords and passwords work at different levels.

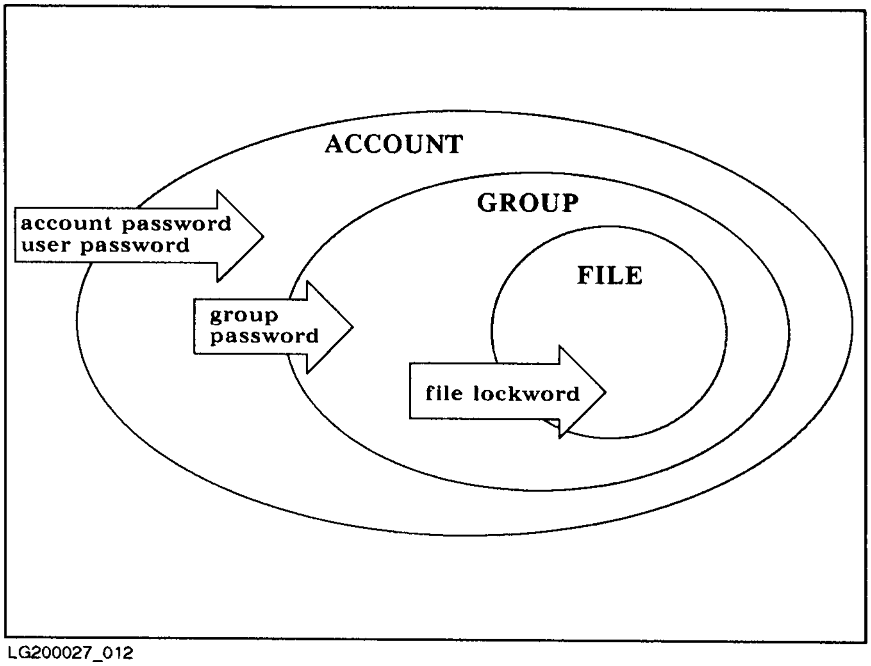


Figure 5-1. Lockwords and Passwords

Note

File lockwords and creator names can be listed only by system managers and account managers.

Releasing and Securing File Security

Sometimes other users need temporary access to your files. For example, individual members of a project team might keep their own records of the hours they worked on different aspects of the project. At the end of the month, the project manager compiles the individual reports into a team report. To compile the team report, the manager might copy the team members' time record files into a single file. To do so, the manager needs temporary access to the team members' time record files.

Give all users temporary access to a file by releasing that file. Releasing a file removes all access restrictions from it. Releasing and securing a file can be executed only by the creator of that file.

Release a file with the RELEASE command. For example:

```
RELEASE MYHOURS.SMITH.PROJECTX
```

The file remains released until it is secured with the SECURE command. For example:

```
SECURE MYHOURS.SMITH.PROJECTX
```

When default file access restrictions are in effect, general users can release and secure files only in their logon group and account.

Summary

Here is a summary of some important file system security rules:

- General users can create files only in their own accounts.
- Only the creator can modify a file's security or rename the file.
- If a file has a lockword, that lockword is required to open the file.
- An account manager has unlimited access to every file within an account. When accessing a protected file created by any other user of the account, the manager must supply the lockword, but can use the LISTFILE or LISTF commands to discover it. For example, the following command lists the lockword for a file named SECRET:

  ```
  LISTFILE SECRET
  ```

- The system manager has unlimited access to any file in the system, if able to supply the lockword (which can be discovered with the LISTFILE or LISTF commands).
- The RELEASE command allows unlimited file access, and the SECURE command secures a file that has been released. To release all security provisions on a file called FREEME, enter:

  ```
  RELEASE FREEME
  ```

  To restore security provisions that were previously in effect for FREEME, enter:

  ```
  SECURE FREEME
  ```

- The ALTSEC command restricts access to specific files in a group to which access is normally not restricted. This command can only be used by the creator of the file.

Refer to the *MPE/iX Commands Reference Manual Volumes 1 and 2* (32650-90003 and 32650-90364) for further information about the ALTSEC, LISTFILE, LISTF, RELEASE, and SECURE commands.

6

Controlling User's Special Abilities with Capabilities.

Capabilities

A variety of people use HP 3000 Computer Systems. They range from those who use the system only to run simple application programs to system programmers who modify MPE/iX. The user who runs application programs, for example, needs only to be able to log on, run a particular program or set of programs, and log off. A system programmer, on the other hand, needs access to special system functions.

Capabilities can help you control who has access to what parts of the system. In order to create permanent files, for example, a user must have Save Files Permanently (SF) capability. To create a session on another terminal from within a session, a user must have Programmatic Sessions (PS) capability. Refer to Table 6-1 for a list of all capabilities and their standard abbreviations, later in this chapter. Refer to appendix A for a complete description of each capability.

You assign capabilities at the account, group, and user level. Account capabilities are the capabilities available to account users and groups. Group capabilities are the subset of account capabilities available to users logged on to a group and to files within the group. Notice, in Table 6-1, that only a subset of the capabilities applies to groups. User capabilities are the subset of account capabilities available to a particular user. When a user issues an MPE command or an intrinsic call, the system checks the user's account, group, and user capabilities against those required for the command or intrinsic.

Files also have capabilities, especially program files. For example, a user does not need privileged mode (PM) capability to run a privileged mode program, but the program itself must have PM capability and the group in which the program file resides must have PM capability.

Listing Capabilities

Three commands allow the system manager to list capabilities of accounts, groups, and users: LISTACCT, LISTGROUP, and LISTUSER.

Listing Capabilities Assigned to an Account

Use the LISTACCT command to check the capabilities of an account. To check the capabilities for the SMITH account, including the password, enter

```
LISTACCT SMITH;PASS
```

The following account information appears on the screen:

```
***************
ACCOUNT: SMITH

DISC SPACE:   754115 (SECTORS)   PASSWORD: ACCTPASS
CPU TIME:     33330 (SECONDS)    LOC ATTR: $00000000
CONNECT TIME:    102 (MINUTES)   SECURITY-- READ    :ANY
DISC LIMIT:   UNLIMITED                     WRITE  : AC
CPU LIMIT:   UNLIMITED                      APPEND   :AC
CONNECT TIME:    UNLIMITED                  LOCK     :ANY
MAX PRI: 150                                EXECUTE  :ANY
GROUP UFID: $0000001 $800001050 $00138A20 $00000008 $000001FA
USER UFID : $0004001 $800001050 $00138C20 $00000008 $000001FB
CAP: AM,AL,GL,DI,CV,UV,LG,CS,ND,SF,IA,BA,PH,DS,MR,PM
```

Refer to appendix A for definitions of the capabilities.

Users with system manager (SM) capability can list any account on the system; all other users can list only their own accounts .

Refer to the *MPE/iX Commands Reference Manual Volumes 1 and 2* (32650-90003 and 32650-90364) for more information on the LISTACCT command.

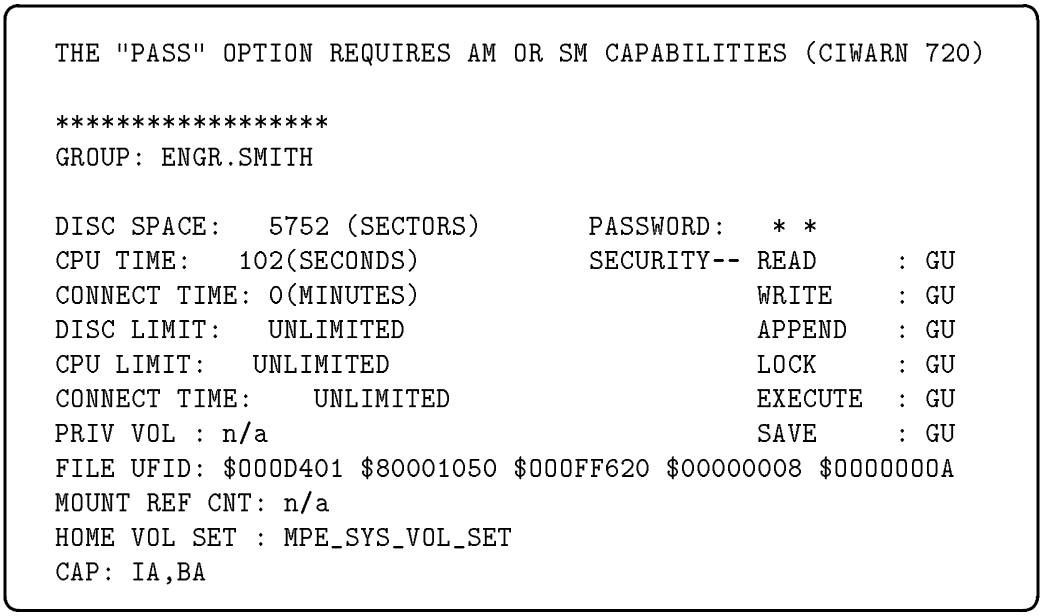
Listing Capabilities Assigned to a Group

Use the LISTGROUP command to display capabilities for one or more groups. For account managers (AM) and system managers (SM), the default is all (@) groups within the user's logon account; for general users, the default is the logon group. Use wildcard characters to specify more than one group.

To check group capabilities and the password of the group ENGR in the account to which you are logged on, enter:

```
LISTGROUP ENGR;PASS
```

The screen displays:



```
THE "PASS" OPTION REQUIRES AM OR SM CAPABILITIES (CIWARN 720)

******************
GROUP: ENGR.SMITH

DISC SPACE:   5752 (SECTORS)       PASSWORD:   * *
CPU TIME:   102(SECONDS)           SECURITY-- READ     : GU
CONNECT TIME: 0(MINUTES)                      WRITE    : GU
DISC LIMIT:   UNLIMITED                       APPEND   : GU
CPU LIMIT:   UNLIMITED                        LOCK     : GU
CONNECT TIME:    UNLIMITED                    EXECUTE  : GU
PRIV VOL : n/a                                SAVE     : GU
FILE UFID: $000D401 $80001050 $000FF620 $00000008 $0000000A
MOUNT REF CNT: n/a
HOME VOL SET : MPE_SYS_VOL_SET
CAP: IA,BA
```

Refer to appendix A for definitions of the capabilities.

Note

If the password is encrypted, the commands `LISTUSER`, `LISTGROUP`, and `LISTACCT` will only display the password as “*ENCRIPTED*”, making a password truley private to its owner.

In this example, the user does not have AM or SM capability, so the password does not appear on the screen.

Refer to the *MPE/iX Commands Reference Manual Volumes 1 and 2* (32650-90003 and 32650-90364) for more information on the `LISTGROUP` command.

Listing Capabilities Assigned to Users

Use the LISTUSER command to check the capabilities of a user. The default is all (@) users and accounts within the user's capabilities (AM or SM). For example, to review the capabilities of the user BORIS in the JONES account, enter:

LISTUSER BORIS;PASS

The screen displays:

```
********************
USER: BORIS.JONES
HOME GROUP:   DEVELOP             PASSWORD:    MYPASS
MAX PRI   :   150                 LOC ATTR:    $00000000
CONNECT TIME:   0(MINUTES)        WRITE   : GU
LOGON CNT : 1
CAP: AM,AL,GL,DI,DV,UV,LG,CS,ND,SF,IA,BA,PH,DS,MR,PM
```

Refer to appendix A for definitions of the capabilities.

Users with system manager (SM) capability can list any user in the system. Users with account manager (AM) capability can list any user in their account. Other users can list only their logon user.

For more information on the LISTUSER command, refer to the *MPE/iX Commands Reference Manual Volumes 1 and 2* (32650-90003 and 32650-90364).

Table 6-1. Capabilities

Capability	Abbreviation	Account	Group	User
System manager	SM	X		X
System supervisor	OP	X		X
Account manager	AM	X		X
Account librarian	AL	X		X
Batch access	BA	X	X	X
Use communications software	CS	X		X
Diagnostician attribute	DI	X		X
Extra data segments	DS	X	X	X
Group librarian	GL	X		X
Interactive access	IA	X	X	X
Multiple RIN	MR	X	X	X
Network administrator	NA	X		X
Node manager	NM	X		X
Use nonshareable devices	ND	X		X
Use private disk volumes	UV	X		X
Privileged mode	PM	X	X	X
Process handling	PH	X	X	X
Programmatic sessions	PS	X		X
Save user files permanently	SF	X		X
Use user logging facility	LG	X		X
Create volume sets	CV	X		X

When the system manager assigns and creates accounts, groups, and users, they each receive certain default capabilities. These capabilties are listed in the following table.

Table 6-2. Default Capabilities

Entity	Default Capabilities
Account	AL, AM, BA, GL, IA, ND, SF
Group	BA, IA
User	BA, IA, ND, SF
Program	BA, IA

Accounts and users may have all 21 of the capabilities, but groups and programs may only have BA, DS, IA, MR, PH, and PM.

Assigning Capabilities

To assign capabilities to accounts, groups, users, and programs

To assign capabilities to accounts, groups, users, and programs, use the NEWACCT, NEWGROUP, and NEWUSER commands. For example, if you are the system manager or the account manager of the PAYROLL account, enter the following to assign capabilities to a new user named GEORGE:

```
NEWUSER GEORGE.PAYROLL;CAP=IA,BA,ND,SF,
```

To alter capabilities

Alter capabilities for existing accounts, groups, and users with ALTACCT, ALTGROUP, and ALTUSER.

For example, if you are the system manager or the account manager of the PAYROLL account, enter the following to alter the capabilities of your new user named GEORGE from the default values:

```
ALTUSER GEORGE.PAYROLL;CAP=IA,BA,ND,SF,GL,AM,OP,PM,DI
```

Now, in addition to the standard default user capabilities, GEORGE has the additional capabilities of group librarian (GL) and account manager (AM).

Capabilities Table

Table 6-3 lists MPE/iX capabilities and their standard abbreviations. It also shows the types of users that require each capability. Use the information in Table 6-3 to establish capabilities for your system.

Table 6-3. Capability Assignments

Capability		Default User	Program	Account Manager	System Supervisor	System Manager
System manager	SM					X
System supervisor	OP				X	X
Account manager	AM			X	X	X
Account librarian	AL			X	X	X
Batch access	BA	X	X	X	X	X
Use Communications Software	CS				X	X
Diagnostician	DI					X
Extra Data Segments	DS		X	X	X	X
Group librarian	GL			X	X	X
Interactive access	IA	X	X	X	X	X
Multiple RIN	MR		X	X	X	X
Network administrator	NA				X	X
Node manager	NM				X	X
Use nonshareable devices	ND	X		X	X	X
Use mounted volume sets	UV					X
Privileged mode	PM		X			X
Process handling	PH		X	X	X	X
Programmatic sessions	PS				X	X
Save user files permanently	SF	X		X	X	X
Use user logging facility	LG				X	X
Create volume sets	CV				X	X

Account Librarian (AL)

A librarian has special file access modes for maintaining files within the account. An account librarian can purge files within the account, although not create or alter them. This attribute is assigned by an account manager.

Account Manager (AM)

An account manager manages all users and groups in that account. The system manager designates the initial manager for each account when creating the account. The account manager can, in turn, assign the attribute to other users in the account.

Batch Access (BA)

This capability allows access to MPE/iX in a batch processing (job) mode.

Use Communications Software (CS)

This capability allows users exclusive access to a communications device such as a DSN/RJE line or a DSN/DS line. It is a requirement in order to use the DSN/RJE subsystem.

Diagnostician (DI)

This capability permits users to run certain device and CPU verification programs. Normally only a Hewlett-Packard service representative (customer engineer) needs this capability.

Extra Data Segments (DS)

This capability lets users and programs create and manage extra data segments. Normally, a program uses these data segments for temporarily storing large amounts of data. Thus, the program's global data area stays relatively small. The extra data segment is purged at the end of the program. Programmers manage extra data segments through the `GETDSEG`, `FREEDSEG`, `DMOVIN`, `DMOVOUT`, and `ALTDSEG` intrinsics. For further information, refer to the *MPE/iX Intrinsics Reference Manual* (32650-90028).

Group Librarian (GL)

A group librarian has special file access modes for maintaining files within the home group. An account manager assigns this attribute. An account manager might, for example, assign group librarian capability to a user with the ability to create and purge files, while assigning only the ability to read and execute files to other users within the group.

Interactive Access (IA)

This capability allows access to MPE/iX in an interactive (session) mode.

Multiple RIN (MR)

This capability lets a user or program acquire more than one resource identification number (RIN) for a single process. It allows exclusive use of more than one resource number simultaneously.

Caution

If you assign MR capability, be sure that the multiple resources are correctly managed. If they are not, resource deadlocking can stop the system.

RINs are managed through the `FREELOCRIN`, `GETLOCRIN`, `LOCGLORIN`, `LOCKLOCRIN`, `LOCRINOWNER`, `UNLOCKGLORIN`, and `UNLOCKLOCRIN` intrinsics. For more information refer to the *MPE/iX Intrinsics Reference Manual* (32650-90028).

Network Administrator (NA)

This capability allows the use of NMMGR . PUB . SYS (the node management services configuration program) to configure NS and LAN and administer the resulting network.

Node Manager (NM)

This capability allows the use of NMMGR.PUB.SYS (the node management services configuration program) to configure and manage nodes in a local area network (LAN).

Use Nonshareable Devices (ND)

This capability allows the use of devices other than terminals and discs including spooled devices. If the device is not spooled, the user has complete control of it. Examples of nonshareable devices are card readers, line printers, magnetic tape units, and plotters. This capability is not needed to use the standard job or session input and list devices.

Use Mountable Volume Sets (UV)

This capability allows access to files residing on mountable volume sets.

Privileged Mode (PM)

Privileged mode gives a user or a program access to all MPE/iX resources, including intrinsics, privileged procedure calls, main memory, system tables and privileged CPU instructions. A program with this capability can run in a permanently privileged mode, or a temporarily privileged mode through the GETPRIVMODE, GETUSERRMODE, and SWITCHDB intrinsics. For further information, refer to the *MPE/iX Intrinsics Reference Manual* (32650-90028).

Caution

Privileged mode bypasses the normal checks and limitations that apply to standard MPE/iX users. A privileged mode program can actually destroy file integrity, including the MPE/iX operating system software itself. Upon request, Hewlett-Packard will investigate and attempt to resolve problems resulting from the use of privileged mode code. This service is not available under the standard service contract, but is available on a time and materials billing basis. However, Hewlett-Packard will not support, correct, or attend to any modification of the MPE/iX operating system software.

Process Handling (PH)

This capability allows the direct creation of other processes by executing the user process. It also allows process suspension, interprocess communication, and process deletion.

With process handling capability, a program can use any of the following intrinsics: ACTIVATE, CREATE, FATHER, GETORIGIN, GETPRIORITY, GETPROCID, GETPROCINFO, KILL, MAIL, RECEIVEMAIL, SENDMAIL, SUSPEND, and TERMINATE. For further information, refer to the *MPE/iX Intrinsics Reference Manual* (32650-90028).

Programmatic Sessions (PS)

This capability permits the use of the STARTSESS command and STARTSESS intrinsic. You can assign this capability to any MPE/iX user. Usually applications programmers use it when creating turnkey systems.

Save User Files Permanently (SF)

This capability allows the use of the BUILD, SAVE, and RESTORE commands, and the SAVE option of the FILE command, described in the *MPE/iX Commands Reference Manual Volumes 1 and 2* (32650-90003 and 32650-90364). Users without SF capability can create job or session temporary files that MPE/iX automatically deletes when the job or session ends.

System Manager (SM)

This capability gives its possessor the capability to manage the overall system, and create accounts within it. The initial person with system manager attribute is designated on the system tape furnished with the HP 3000 Computer System. The original system manager can create other users with SM capability.

System Supervisor (OP)

Users with system supervisor capability have day-to-day external control of the system. An account manager with OP capability can assign it to other users within the account.

Use User Logging Facility (LG)

This capability allows its owner to use user logging commands.

Create Mountable Volume Sets (CV)

This capability is needed to create, alter, and delete mountable volume sets. A user given CV capability automatically has UV capability.

7

Auditing System Use

This chapter describes methods for creating audit trails, by which system usage can be determined. Well defined audit trails tell you:

- Who is and who has been using the system.
- When.
- For how long.
- Which files were accessed.
- Which commands and system facilities were used.

MPE/iX provides you with three separate logging facilities: system logging, memory logging, and user logging. Each operates separately and has no effect on the others. The purpose of each is as follows:

- System Logging: Records the use of system resources by accounts, groups, and users on a job or session basis. Logs errors and other events detected by various system modules.
- Memory Logging: Records errors that occurred in memory. This function is useful primarily to system administrators.
- User Logging: Allows users and subsystems to record additions and modifications to databases and other files used in applications programs.

System file information is also recorded in the system log file and that is why the System Managers need to know about system logging and `LOGTOOL` to audit system security. With the system logging facility, you can keep track of the following:

- File open.*
- File close.
- Command access*
- Job initiation.
- Process termination.
- Line disconnection.
- Line close.
- Console messages.
- Stream initiation.
- User logging.

- Process initiation.
- Security configuration changes.*
- ACD creation and modification.
- Printer access refusals.
- System logging configuration.
- Restore.
- System shutdown.
- Logging errors.
- System startup.
- System shutdown.
- Power failures.
- Spoolfile completions.
- Physical volume mounts/dismounts.
- Logical volume mounts/dismounts
- Tape label reads.
- System console activity.

Items marked with an asterisk (*) are provided in the HP Security Monitor/iX.

Two additional security facilities are included in the HP Security Monitor but do not create log records. The two facilities are:

- Auditability by named user.
- Assurance of auditability.

System logging is discussed in detail in the rest of this chapter. User logging is covered in the *User Logging Programmer's Guide* (32650-90027).

Using System Logging

System logging records the use of certain resources by accounts, groups, and users. Like the administrative functions, system logging can be used for billing purposes or for obtaining an overview of system use. System logging is also used to detect security attacks or breaches after the fact.

Unlike these administrative functions, system logging describes system use by creating a running log of events, correlated with the job or session that caused each event. System logging is the only means of recording system use on a job/session basis.

The majority of logging events are optional; when the system is configured, the system manager can select whether they are recorded or not. In addition to the `LOGGING ENABLED` event, the following events are always enabled when the system is started:

- Type 100, log failure record
- Type 101, system up record
- Type 111, I/O errors
- Type 150, diagnostic information
- Type 151, high-priority machine check record
- Type 152, low-priority machine check record

The events that the system manager chooses to monitor are recorded on log records contained in a disk file. Each event is recorded in one logical record.

The LOG configurator

The LOG configurator enables the system manager to change the attributes of user and system logging processes.

System logging records the use of certain system resources by accounts, groups, and users on a job or session basis. The system manager determines which events are logged.

User logging allows users and subsystems to record additions and modifications to databases and other files used in applications programs. The system manager determines the maximum number of logging processes and the maximum number of users per logging process.

Entering the LOG Configurator

You can use the LOG configurator which is accessed through SYSGEN to change the attributes of user and system logging processes. To access the LOG configurator, enter the `LOG` command (abbreviated `LO`) at the SYSGEN prompt as shown in the following example:

```
sysgen>LOG

        ** LOG configurator commands **

        show (sh)        slog (sl)       ulog (ul)

        clear (cl)(c)    exit (ex)(e)    help (he)(h)    hold (ho)

        oclose (oc)      redo
     log>
```

Example 6-1. Activating the LOG Configurator

Using the LOG Configurator Help Facility

The help facility enables you to quickly identify the function and syntax of those LOG configurator commands and options for performing the multiple operations that define or change logging processes.

To obtain a list of the commands available for use in the LOG configurator, enter `HELP` at the LOG configurator prompt as shown in the following example:

```
log>HELP

    ** LOG configurator commands **

    show (sh)        slog (sl)       ulog (ul)

    clear (cl)(c)    exit (ex)(e)    help (he)(h)    hold (ho)

    oclose (oc)      redo

log>
```

Example 6-2. LOG Configurator Help

To display the syntax for each available command, enter `HELP ALL` as shown in the following example:

```
log>HELP ALL

   command (abb)  parameter=value
   -------------  ---------------

   show (sh)      [command      = SLOG|ULOG|ALL]
                  [dest         = OFFLINE]

   slog (sl)      [on           = event#,...]
                  [off          = event#,...]

   ulog           [nlogprocs    = number processes allowed]
                  [usersperproc = users per logging process]

log>
```

Example 6-3. LOG Configurator HELP ALL

Entering `HELP` *commandname* provides help for a specific command:

```
log>HELP SHOW


show (sh)      [command      = SLOG|ULOG|ALL]
               [dest         = OFFLINE]
```

Showing Current LOG Values

The SHOW command displays the LOG values as currently set.

SHOW has the following parameters:

```
SHOW    [COMMAND =  SLOG    ]
                    ULOG    ]
                    ALL     ]
        [DEST    =  OFFLINE ]
```

SLOG lists the state of the system logging events.

ULOG lists the number of user logging processes and users per logging process currently configured.

ALL lists all the information associated with the LOG configurator.

OFFLINE redirects the output of the SHOW command to the SYSGEN listing file, SYSGLIST. Using OFFLINE does not immediately generate a printout. The information is sent to SYSGLIST until you either enter the OCLOSE command or exit the configurator. Exiting the configurator or using OCLOSE closes SYSGLIST and prints the file.

Using SHOW without using any parameters, is the same as specifying SHOW ALL. In addition, the value entered for the ULOG parameter includes the minimum, maximum, current, and default values.

To show the current user logging process, enter SHOW ULOG:

```
log>SHOW ULOG
    configurable item             max     min    current
    -----------------           ------- -------  -------
    # of user logging processes    128       2   64
    # users per logging process   1024       1   128
```

Example 6-4. Showing User Logging Processes

To view all currently configured values, enter SHOW ALL:

Logging System Events

System logging records the use of certain resources by accounts, groups, and users. System logging can be used for several purposes, such as billing or obtaining an overview of system use. System logging describes system use by creating a running log of actual events, correlating the event with a job and session. The system manager chooses which events to enable or disable by setting an event number to ON or OFF. (Refer to the preceding example for a list of event numbers and their definitions.)

The SLOG command enables and disables the selected system logging events. SLOG has the following parameters:

```
SLOG (SL)   [ON  = event#, ...  ]

            [OFF = event#, ...  ]
```

Enable the logging of an event by entering `SLOG` *event#*, . . . :

```
log>SLOG 100      (Event 100 enabled)
```

or

```
log>SLOG ON=100   (Event 100 enabled)
```

Disable the logging of an event by entering `SLOG OFF=`*event#*, . . . :

```
log>SLOG OFF=100  (Event 100 disabled)
```

Entering `SLOG` without `ON` enables logging. Entering `SLOG` without an event number causes an error:

```
log>SLOG

(error - no parameters are specified)''
```

Logging event 100 is a special case. If 100 is off, no logging (except that forced on by MPE/iX) takes place.

Note

Some events are permanently set to ON. Currently, events 101, 111, and 150 are forced on by MPE/iX.

Logging User Events

User logging provides a means for system users and subsystems to record additions and modifications to databases and other files using application programs. The system manager determines the maximum number of logging processes and the maximum number of users per logging process.

The `ULOG` command configures the user logging process parameters. `ULOG` has the following parameters:

`ULOG (UL)  [NLOGPROCS =` *numberprocesses allowed*]

`[USERSPERPROC =` *usersperloggingprocess*]

`NLOGPROCS` controls the user logging ID (LID) table size. Lowering `NLOGPROCS` loses all current logging ID information from the tape created by SYSGEN. If `NLOGPROCS` remains unchanged or increases, the current logging ID information is copied to tape. The minimum and maximum number of processes allowed are 2 and 128, respectively.

`USERSPERPROC` specifies the maximum number of users assigned to each configured logging process. The minimum and maximum number of users per logging process are 1 and 1024, respectively.

Note

Changing `NLOGPROCS` takes effect when a tape is created and an `UPDATE CONFIG` or `INSTALL` is conducted.

To set the number of processes or users per process, enter `ULOG` followed by the number of processes or users:

```
log>ULOG 40  ** Number of Processes **
```

or

```
log>ULOG USERSPERPROC=40  ** Number of Users per Process **
```

Clearing Log Configuration Changes

If you desire to clear all LOG configuration changes made, enter the CLEAR command at the LOG configurator prompt.

```
log>CLEAR
```

Once a SYSGEN> KEEP is done, the changes kept become permanent and CLEAR does not remove them.

Holding and Saving Configuration Changes

Using the system logging and user logging commands described in the following sections changes the LOG configuration specified in the SYSGEN command line or global BASEGROUP command. These changes are temporary and are easily lost if not properly saved.

Saving configuration changes is a two-step procedure. After you alter a configuration, you must, first, hold the changes before exiting the configurator. Second, use the global module KEEP command to save the changes.

To hold changes, enter the HOLD command at the LOG configurator prompt:

```
log>HOLD
```

You can work in a SYSGEN configurator, hold the changes, and continue working in other SYSGEN configurators before saving the changes.

To save the changes, hold all desired changes, exit to SYSGEN's global module, and issue the KEEP command:

```
sysgen>KEEP newgroupname
```

Entering an MPE Command from the LOG Configurator

Use the colon (:) to introduce an MPE command from the LOG configurator. To issue an MPE command, enter the command along with the colon. For example,

```
      log>:SHOWTIME
THU, APR 20, 1989,  2:55PM
      log>
```

Exiting the LOG Configurator

Use the EXIT command to terminate the LOG configurator and return to the SYSGEN global module. Exit may be abbreviated EX or E. To end working in the LOG configurator, enter EXIT at the LOG configurator prompt:

```
log>EXIT

sysgen>
```

Printing a Log File

To analyze your logs and to read what you are logging, you must print your log files. To do this, use the `LOGTOOL` utility program. The `LOGTOOL` utility runs under the online diagnostic system, and can be invoked by entering SYSDIAG. When the diagnostic user interface prompt (`DUI>`) appears, enter `RUN LOGTOOL`.

In order to print a log, issue the following:

1. `:SYSDIAG`
2. `DUI>RUN LOGTOOL`
3. `LOGTOOL>LIST LOG=`*log#* `OUTFILE=LP`
4. `LOGTOOL>EXIT`
5. `DUI>EXIT`

Enter `HELP` after the `LOGTOOL` prompt for more information. The `STATUS` command reports on the status of all system log files.

The following example shows the use of the `STATUS` command in the sequence of printing a log.

```
:SYSDIAG
DUI >RUN LOGTOOL
LOGTOOL>status
LOGTOOL>log=0027 outfile=LP
DUI >EXIT
```

If you do not specify the `OUTFILE` parameter, the log prints on your terminal screen. Typically this report is very long and ties up your terminal for quite some time. If this does happen, you can enter CTRL Y to break the process.

Printing a subset of a log file

If you like, you can filter the output of LOGTOOL utility to show you information about only a specific user or users. The syntax for this is shown below.

$$\texttt{LIST}\ \left\{\texttt{LOG=}\mathit{log\_name}\right\} \begin{bmatrix} \texttt{;JSNAME=}\mathit{job/session\_name} \\ \texttt{;USER=}\mathit{user\_name} \\ \texttt{;ACCOUNT=}\mathit{account\_name} \end{bmatrix} [\ \ldots\ ]$$

The input for these commands should be no longer than 80 characters. Default for all parameters is the wildcard @.

For example, to select log records from log files 1 through 5, with log information about password changes (log type 134), and user identification JTEST,MARIA.PAYROLL, you would enter the following.

```
>LIST LOG=1/5;TYPE=134;JSNAME=JTEST;USER=MARIA;ACCOUNT=PAYROLL
```

Accessing Log Files from Programs

The following sections include information that you need to access log files programmatically.

Creating and naming log files

When system logging is first enabled, MPE/iX creates and opens the first log file and begins recording events as they occur. When this log file is full, or when the system is shut down and restarted, MPE/iX creates and opens a new log file.

Log file names always take the form `LOG`*xxxx*`.PUB.SYS`, where *xxxx* is the log file number, ranging from 0000 to 9999. The first log file is `LOG000`; when it is closed and a new log file opened, MPE/iX increments the file number by one. Each time a new log file is created, a console message, similar to the following, displays the new log file number:

```
LOG FILE NUMBER xxxx ON
```

Log file commands

Three MPE/iX commands, `SHOWLOG, SWITCHLOG,` and `RESUMELOG`, are available to control system logging. The `SHOWLOG` command displays the number of the current log files and the percentage of file space already used to record logging events. For example:

```
LOG FILE LOG9675 IS 16% FULL
```

The `SWITCHLOG` command closes the current log file, and creates and opens a new one.

The third command, `RESUMELOG`, restarts the logging process after it is suspended because of an error.

Note

You must have system supervisor (OP) capability to use these three commands. For more information, refer to the *MPE/iX Commands Reference Manual Volumes 1 and 2* (32650-90003 and 32650-90364).

File security

Log files are created by, and therefore belong to, the system logging process. By default, their creator is `MANAGER.SYS`. They are assigned the MPE/iX default security provisions typically assigned to files within the `PUB` group of the `SYS` account. The current log file can be modified only by users assigned Account Librarian (AL) capability for the `SYS` account, or by `PUB` group users (GU capability) of the `SYS` account.

Once the log file is closed, MPE/iX changes the file access restrictions on the file from ANY to CR (the file creator) only. The result is that only the system manager controls access to current and closed log files.

Log file structure

All log files are created as files containing variable-length records. They should always be treated as files containing variable-length records, accessed sequentially.

For a log file, the end-of-file pointer can point at the last record (block) written to the file (if the file is closed normally), or at any point beyond the last record written (if the file has not been closed). In the latter case, all space following the last record is padded with zeros.

The general format of a log file is shown in Figure 7-1. The log file record size is 2048 bytes with a maximum of 1024 records per file.

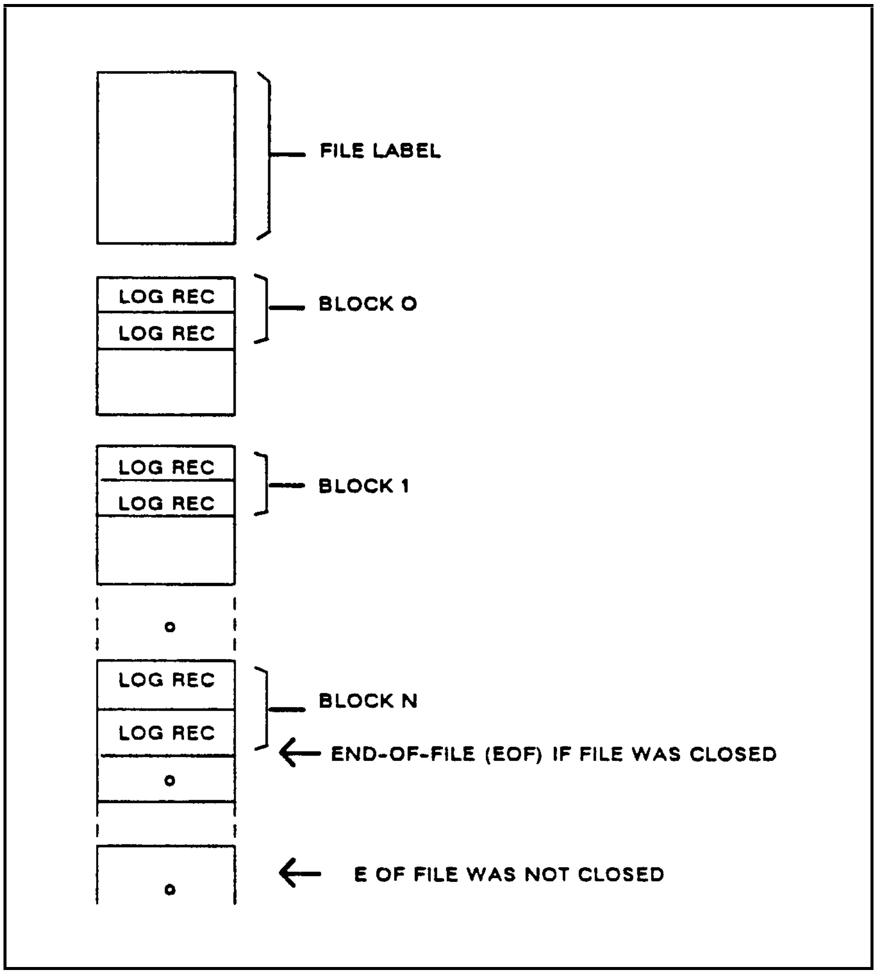


Figure 7-1. Log File Format

Console messages for log files

Log file status and error messages are reported to the system console. They conform to the format *hh/mm/PIN/message*, where:

hh	= the hour of the day
mm	= the minute of the hour
PIN	= the process identification number
message	= the message text

The log file status message text may consist of any of the following:

- `LOG FILE NUMBER` *xxxx* `ON` indicates that a new log file has been created. This message always appears prior to the welcome message after a restart. If displayed while the system is running, it indicates that the previous current log file has been closed.
- `LOG FILE NUMBER` *xxxx* `IS 50% FULL` indicates that logging data fills up half of the log file's allotted file space.
- `LOG FILE NUMBER` *xxxx* `IS 75% FULL` indicates that logging data fills up three-quarters of the log file's allotted file space.
- `LOG FILE NUMBER` *xxxx* `LOGGING RESUMED` indicates that a `RESUMELOG` command was successfully executed.

Log file errors are reported in one of the following messages. Refer to Table 7-1 for a summary of log file error numbers, their meaning, and whether they are recoverable or irrecoverable errors.

- `LOG FILE NUMBER` *xxxx* `ERROR #`*nn,* `LOGGING STOPPED` indicates that an irrecoverable error occurred; system logging is disabled until the next system startup.
- `LOG FILE NUMBER` *xxxx* `ERROR #`*nn.* `LOGGING SUSPENDED` indicates that a recoverable error occurred. A recoverable error temporarily suspends logging until the system supervisor issues the `RESUMELOG` command, discussed previously.

File error handling

Two types of errors prevent the system logging facility from maintaining the log file:

- **Catastrophic errors.** Caused by physical I/O errors or unit failures. These errors are not recoverable and will disable logging until the next restart.
- **Managerial errors.** Encountered during creation and management of the log file. These are usually recoverable, and they cause logging to be temporarily suspended. Logging resumes when the problem is resolved and a `RESUMELOG` command is issued.

When logging resumes, a special log record is created, denoting the total number of records missed, the number of job/session initiation records missed, the number of job/session termination records missed, and the number of I/O records missed. To analyze this or any other entry in the log file, run the `LOGTOOL` utility program.The following table shows the various file errors that are logged:

Table 7-1. Log File Errors

Error #	Error	Recover?
1	Input/output error in accessing the system disk.	No
2	Input/output error in accessing disk log file.	No
21	Data parity error.	No
26	Transmission error.	No
27	Input/output timeout.	No
28	Data overrun.	No
29	SIO failure.	No
30	Unit failure.	No
46	Insufficient disk space to create log file.	Yes
47	Input/output error on file label.	No
57	Virtual memory not sufficient.	No
61	Group (PUB) disk space exceeded in creating log file.	Yes
62	Account (SYS) disk space exceeded in creating log file.	Yes
63	Group disk, space exceeded in allocating new extent to the log file.	
64	Account disk space exceeded in creating log file.	Yes
100	A file of the same name as the current log file already exists in the system file directory.	Yes
102	Directory input/output error.	No
103	System directory overflow.	No
105	Illegal variable block structure.	No

LOGTOOL

In addition to the LOG configurator, the System and Memory Log Analysis Tool (`LOGTOOL`) enables you to display and manage system log files and the memory log file. System log files contain information generated by the operating system. The memory error log file contains memory error information gathered by the memory error logging process `MEMLOGP`.

Using the LOGTOOL Utility

To invoke `LOGTOOL` enter:

```
SYSDIAG
DUI > RUN LOGTOOL
```

For detailed information on any command enter `HELP` followed by the command. For example:

```
LOGTOOL> HELP LIST
```

The following is a sample of commands you would use to display data from a set of system log files.

1. Log on as `MANAGER.SYS` or with `SM`, `OP` or `DI` capability.

2. List the names of log files currently on your system (*before* invoking `LOGTOOL`):

```
LISTFILE LOG@.PUB.SYS
```

3. Invoke `LOGTOOL`:

```
SYSDIAG
DUI > RUN LOGTOOL
```

4. If you wish to obtain data from your *current* logfile enter the following to close it and open a new one:

```
LOGTOOL> SWITCHLOG
```

5. Display logfile record *types* (you may skip this step if you already are familiar with *types*) :

```
LOGTOOL> TYPES
```

6. Display the logfile analysis. The following command accomplishes three things: 1 - Specifies the numbers of the logfiles you wish to examine (see results of above `LISTFILE` command), 2 - Specifies the logging event types you wish to examine (see results of above `TYPES` command) and 3 - Produces a formatted listing of information from the logfiles.

```
SYSDIAG>LIST LOG=9/14,17,20,22;TYPE=111,146
```

The `LOG` parameter restricts analysis to logfiles `LOG0009` through `LOG0014` and `LOG0017` and `LOG0020` and `LOG0022`. The `LOG` parameter may be entered as a *range* of numbers such as 9/14, as a *string* of numbers such as 17,20,22 (or a single number), or as a combination *range* and *string* (as in the example).

The `TYPE` parameter says to select data only for event types 111 and 146 (I/O errors and maintenance requests).

7. If you do *not* wish to see the analysis on your terminal screen, but would rather write the records to an *output* file, use the `;OUTFILE` parameter as follows (Otherwise, proceed to step 8):

```
SYSDIAG>LIST LOG=9/14,17,20,22;OUTFILE=MYFILE;TYPE=111,146
```

The output file to which the analysis will be written in this example is MYFILE. You may choose any name but it must begin with an alphabetic character.

Please remember that the output file will be written to the DIAG group of the SYS account.

You may use any HP3000 text editor to examine the output file. You may also copy it with the COPY command or the FCOPY utility.

8. Terminate LOGTOOL:

```
LOGTOOL> EXIT
DUI > EXIT
```

COMMAND SUMMARY

Three categories of LOGTOOL commands are:

- System Log File Commands (SLF)
- Memory Log File Commands (MLF)
- Miscellaneous Commands (MC)

The following is a list and brief description of commands available in LOGTOOL.

Table 7-2. LOGTOOL Commands

Name	Category	Description
DISPLAYLOG	(SLF)	Displays I/O entries as information is logged.
EXIT	(MC)	Exits LOGTOOL and returns user to DUI.
HELP	(MC)	Gives help on running LOGTOOL.
LAYOUT	(SLF)	Reads in a layout file.
LIST	(SLF)	Lists contents of a system log file.
MEMCLR	(MLF)	Clears the memory logging process log files.
MEMRPT	(MLF)	Displays the contents of the memory log file.
MEMTIMER	(MLF)	Alters the timer value of the memory error logging process.
PURGESYSLOG	(SLF)	Deletes the specified system log files from the disc.
PURGEWORK	(SLF)	Deletes the specified work files from the disc.
REDO	(MC)	Edits any of the last four lines of text entered.
SELECT	(SLF)	Selects specified records from the system log files.
STATUS	(SLF)	Reports on the status of all system log files.
SUSPEND	(MC)	Suspends LOGTOOL and returns control to the DUI.
SWITCHLOG	(SLF)	Causes the system to start a new system log file.
TYPES	(SLF)	Describes the system log file “types”.

Logging Formats

MPE/iX writes log records to records in a log file. The log records can be accessed and displayed by using the system log analysis utility (LOGTOOL) or through a user-supplied analysis program.

There are two types of log files used to record system information. There is the original 100 series format and a newer 200 series format which has been adopted to acommodate POSIX specifications. Following is a description of the formats used by each format:

Format 1## system log record header

Table 7-3 illustrates the system log record header.

Table 7-3. System Log Record Header

Length, in 16-bit Words	Field Content
1	Record type
1	Record length
1	PIN
1	Time stamp date
2	Time stamp time
2	Job type/Job number

Format 1## system log audit trailer

Table 7-4 illustrates the system log audit trailer.

Table 7-4. System Log Audit Trailer

Length, in 16-bit Words	Field Content
8	User name
8	Logon group
8	Account name
8	Job/Session name

The system log audit trailer is currently appended to the following events:

- job termination
- process termination
- physical mount/dismount
- tape labels record

- console log
- program file event
- new commercial spooling
- password changes
- system logging configuration
- RESTORE logging
- printer access failure
- ACD changes
- stream initiation
- user logging
- process creation
- CHGROUP logging
- FOPEN logging

Format 2## system log record header

Format 2## log records have the information normally contained within the audit trailer incorporated into the main event record.

Table 7-5. Format 2## System Log Record Content

Length, in 16-bit Words	Field Content
1	Record type
1	Event version
1	Record length
1	PIN
2	Job type/Job number
2	Time stamp time
1	Time stamp date
1	Login type (If 1, logon name is in the format 16-byte user 16-byte acct)
18	Logon name
8	Job/Session Name
4	Reserved

The default configuration is determined at SYSGEN. Records that are not initially used to log information are OFF; records that are enabled by default are ON. Typically, the system manager sets up and maintains the system logging configuration (this determines which records will be logged). The logging configuration can be altered using the SYSGEN utility.

Log Record Types

Here are the various log record types:

Table 7-6. Log Record Types

Log Type	Record Description	State
100	System Logging	ON
101	System Up	ON
102	Job Initiation	OFF
103	Job Termination	OFF
104	Process Termination	OFF
105	NM File Close	OFF
106	System Shutdown	ON
107	Power Failure	ON
111	I/O Error	ON
112	Physical Mount/Dismount	OFF
113	Logical Mount/Dismount	OFF
114	Tape Label	OFF
115	Console Log	ON
116	Program File Event	ON
120	Native Mode Spooling	ON
127	Chdir	OFF
128	Process Adoption	OFF
129	File Owner Change	
130	Architected Interface	OFF
131	Additional Processor Launch	OFF
134	Password Change	OFF
135	System Logging Configuration	ON
136	Restore	OFF
137	Printer Access Failure	OFF
138	ACD Change	OFF
139	Stream Initiation	OFF
140	User Logging	OFF
141	Process Creation	OFF
142	Security Configuration Changes	OFF
143	Chgroup	OFF
144	File open	OFF
145	CI Command Logging	OFF
146	Maintenance Request	OFF
148	UPS Monitor Event Logging	OFF
150	Diagnostic Information	ON
151	High Priority Machine Check	ON
152	Low-priority Machine Check	ON
155	Directory Open/Close Logging	OFF
160	CM File Close	OFF

System Log Record Formats

The rest of this chapter includes the format of the log records. Notes following the log records describe the significant fields in the records.

Log failure record, type 100

Table 7-7. Log Record Heading Format

Length, in 16-bit Words	Record Content
1	Record type (100)
1	Record length
1	Process identification number
3	Time stamp
2	Job type/job number
2	Missing log records*
1	Missing job initiations*
1	Missing job terminations*
1	Missing I/O records*

* Lost when system logging is suspended or disabled.

System up record, type 101

Table 7-8. System Up Record Format

Length, in 16-bit Words	Record Content
1	Record type (101)
1	Record length
1	Process identification number
3	Time stamp
2	Job type/job number
4	Version ID (v.uu.ff)/Last 8 bits unused
2	Maximum number of concurrent jobs and sessions
2	Boot code
16	Boot device
16	Configuration group used for boot
2	NL checksum
2	Config checksum
2	SL checksum
128	Operating system nonvolatile storage

NOTES:

Boot code:

0 = Start with recovery.
1 = Start with no recovery.
2 = Update with a configuration change.
3 = Update without a configuration change.
4 = Install (new operating system loaded).

Configuration group used for boot (ASCII)

Name of group containing configuration data files used when `START` was invoked.

The NL, Config, and SL checksums fields are set to 0.

Operating system nonvolatile storage contains the following:

Table 7-9. ISL Data for Last Boot (128 bytes)

Reserved	36 bytes
Boot path	32 bytes
ISL revision	4 bytes
Time stamp (# seconds since 1970)	4 bytes
LIF utility entries entered	48 bytes
Pointer to last utility	1 byte
Word alignment	3 bytes

Table 7-10.
MPE/iX Operating System System-Dependent Data (128 bytes)

Time stamp (# seconds since 1970)	4 bytes
# microseconds since last second	4 bytes
GR2 (caller's PC)	4 bytes
MPE/iX status: For system abort, this contains failure number: (1) Error (2 bytes) (2) Subsystem (2 bytes)	4 bytes
MPE/iX version (*vv.uu.ff*)	8 bytes
Message string (if any)*	64 bytes
Halt number	2 bytes
Lockup error code	2 bytes
Reserved	36 bytes

*Identifies entity logging this entry (abort or shutdown).

Boot Device

Indicates the primary boot path, as follows:

Word 1:		
Bits	(0:8)	Flags
	(8:8)	Bus Converter 0
Word 2:		
Bits	(0:8)	Bus Converter 1
	(8:8)	Bus Converter 2
Word 3:		
Bits	(0:8)	Bus Converter 3
	(8:8)	Bus Converter 4
Word 4:		
Bits	(0:8)	Bus Converter 5
	(8:8)	Physical Module

Words 5 and 6: Logical Module

Words 7-16: Device-dependent

Job initiation record, type 102

Table 7-11. Job Initiation Record Format

Length, in 16-bit Words	Record Content
1	Record type (102)
1	Record length
1	Process identification number
3	Time stamp
2	Job type/job number
8	Job name
8	User name
8	Home group name
8	Account name
8	Logon group name
1	Input logical device number
1	Output logical device number
1	Reserved (bits 0-7)/Logon queue (bits 8-15)
2	CPU time limit
1	Inpri (bits 0-7)/Outpri (bits 8-15)
8	CI program name
8	CI program group name
8	CI program account name
2	MPE/iX logon status

NOTES:

Logon queue

Execution queue the job will run on.

CPU time limit

CPU time limit (in number of seconds) given by user on `JOB` or `HELLO` command, as follows:

```
0 = not given
# = number given
```

MPE/iX status

If failure occurred during job initiation, the MPE/iX status indicates an error. This field can be any MPE/iX status from the OS. If successful, value is zero.

Unlike MPE V, unsuccessful logon attempts are logged in MPE/iX system logging.

Job termination record, type 103

Table 7-12. Job Termination Record Format

Length, in 16-bit Words	Record Content
1	Record type (103)
1	Record length
1	Process identification number
3	Time stamp
2	Job type/job number
1	Maximum priority
1	Number of creations
2	CPU time in seconds
2	Connect time

Process termination record, type 104

Table 7-13. Process Termination Record Format

Length, in 16-bit Words	Record Content
1	Record type (104)
1	Record length
1	Process identification number
3	Time stamp
2	Job type/job number
2	CPU time in milliseconds
2	Native mode stack size in bytes
2	Native mode heap size in bytes
1	CM mas stack in 16-bit words
1	Termination type
8	Reserved

NOTES:

Termination types:

0 = Normal

1 = Dependency (This process depends upon a terminated process.)

2 = Killed (This process was terminated by another using `KILL`.)

3 = Quit (This process called the `QUIT` intrinsic.)

4 = Quitprog (This process called the `QUITPROG` intrinsic.)

5 = Softfault (This process terminated due to a fault.)

NM File close record, type 105

Table 7-14. File Close Record Format

Length, in 16-bit Words	Record Content
1	Record type (105)
1	Record length
1	Process identification number
3	Time stamp
2	Job type/job number
4	Number of logical reads
4	Number of byes read
4	Number of logical writes
4	Number of bytes written
8	File name
8	Group name
8	Account name
8	Creator name
8	User name
8	User group
8	User account
10	Unique file identifier (UFID)
2	Reserved
2	Close disposition
2	Open domain
2	File size in bytes
4	File open count

NM File close record, type 205

The type 205 record has a "maximum path exceeded" flag.

The LDEV number of a file is also included in the log record. In most cases thi s is the LDEV number of the file's label and initial extent since files are not usually restricted to one volume.

Table 7-15 illustrates the format of the type 205 record.

Table 7-15. Record Type 205

Length, in 16-bit Words	Record Content
1	Record type 205
1	Event version
1	Record length
1	PIN
2	Job type/Job number
2	Time stamp time
1	Time stamp date
1	Login type (if 1, logon name is in the format 16-byte user 16-byte acct)
18	Logon name
8	Job/session name
4	Reserved
4	# logical reads
4	# bytes read
4	# logical writes
4	# bytes written
8	Creator user name (from flabel: *user.acct* for new files)
10	UFID
2	Close disposition
2	Open domain
2	File size
2	File number
4	File open count
4	Number records read
4	Number records written
1	File LDEV #
1	"Maximum path exceeded" flag (true if full path not recordable)
##	Variable-length name buffer (file name terminated with a 0)

NOTES:

Unique file identifier (UFID)

Internal file identifier. Internal data structure that uniquely identifies a file. This entity is printed in hex.

Disposition field

Close Disposition (bits 13:3)		**Open Domain**	
0	No change.	0	New file.
1	Save permanent.	1	Old permanent file.
2	Save temporary - rewound.	2	Old temporary file.
3	Save temp - not rewound.	3	Old job or sys.
4	Delete.		
5	Make temporary.		

Disk Space Disposition (bits 11:2)

0	Do not return disk space allocated beyond EOF.
1	Return disk space allocated beyond EOF to system; EOF becomes file limit.
2	Return disk space allocated beyond EOF to system; file limit remains the same (NM file types only).

The Disposition field in an NM file close record can have values from 0-9, 8-13, and 16-21. The number 255 is also a valid value for the Disposition field. Any files left open during process termination are closed by the system and given a disposition of 255.

Shutdown record, type 106

Table 7-16. Shutdown Record Format

Length, in 16-bit Words	Record Content
1	Record type (106)
1	Record length
1	Process identification number
3	Time stamp
2	Job type/job number
1	Number of jobs
1	Number of sessions

Power failure record, type 107

Power Failure Record Format

Length, in 16-bit Words	Record Content
1	Record type (107)
1	Record length
1	Process identification number
3	Time stamp
2	Job type/job number

I/O error record, type 111

Table 7-17. I/O Error Record Format

Length, in 16-bit Words	Record Content
1	Record type (111)
1	Record length
1	Process identification number
3	Time stamp
2	Job type/job number
20	Hardware product number
20	Physical path description
20	Logical device name
1	Device class identifier
1	Diagnostic message class (bits 0-7)/Not used (bits 8-15)
2	I/O manager status
1	Reserved (bits 0-6) LAR (bit 7)/\Reserved (bits 8-14) RA (bit 15)
1	Reserved (bits 0-6) IW (bit 7)/\Reserved (bits 8-14) AD (bit 15)
1	Retry count
1	Reserved
2	I/O manager port number
2	Transaction number
1	Hardware status length
1	Not used
20	Hardware status
1	Length of I/O manager data
1	ID of I/O manager
36	I/O manager-specific data

NOTES:

Hardware product number is the number of the device; for example, 7935.

Physical path description is the hardware path to the device; for example, 2/4.0.1 (2 = bus, 4 = channel, 0 = device adapter, and 1 = device). It is hardware dependent.

Logical device name is the LDEV number.

Device class identifier identifies the type of device, such as disk, tape drive, or printer.

Diagnostic message class specifies the reason for logging this error:

`0`	= Hardware event
`1`	= Software event
`2`	= Other

I/O manager status is the LLIO status from the driver. It gives the reason for the I/O error.

`LAR`	= Log all retries requested by I/O manager
`RA`	= Retry again - I/O manager attempts retry
`IW`	= I/O worked - Retry was successful
`AD`	= Auto-diagnostic requested by I/O manager

Retry count varies depending on the value of `LAR`. If `LAR` is true, then the retry count is the number for the nth retry. If `LAR` is false, it is the number of retries performed.

I/O manager port number is the IPC port number of the device manager. It is usually a negative number.

Transaction number is the number of the I/O request.

Hardware status length specifies how much of the hardware status field contains valid data.

Hardware status contains status bytes from the device. The field may contain up to 40 bytes of error information. The information is device dependent.

Length of I/O manager data specifies how much of the I/O manager specific data field contains valid data.

ID of I/O manager is the subsystem number.

I/O manager specific data is driver-specific data that relates to the I/O error. It can contain up to 72 bytes of information.

Physical mount/dismount record, type 112

Table 7-18. Physical Mount/Dismount Format

Length, in 16-bit Words	Record Content
1	Record type (112)
1	Record length
1	Process identification number
3	Time stamp
2	Job type/job number
1	Mount/Dismount type
1	Volume type
2	Logical device number
1	MV table ID (bits 0-5)/Not used (6-15)
1	Vol ID (bits 0-7)/Not used (8-15)
2	Volume identification
8	Volume name
16	Volume set name

NOTES:

Mount/Dismount Values

0 = Device- or user-initiated mount
1 = Device-initiated dismount

Log Volume Types

1 = Master volume
3 = Loner volume
6 = Scratch volume
7 = Unknown volume

Logical mount/dismount record, type 113

Table 7-19. Logical Mount/Dismount Record Format

Length, in 16-bit Words	Record Content
1	Record type (113)
1	Record length
1	Process identification number
3	Time stamp
2	Job type/job number
1	Mount/Dismount
1	Request type
8	User name
8	Group name
8	Account name
16	Volume set name
1	Number of volumes in set
1	Logical device of first volume
1	Logical device of next volume
1	Logical device of next volume
1	Logical device of next volume
1	Logical device of next volume
1	Logical device of next volume
1	Logical device of next volume
1	Logical device of next volume

NOTES:

Mount/Dismount	Request Types
0 = Logical mount	0 = User
1 = Logical dismount	1 = Operator

Tape labels record, type 114

Table 7-20. Tape Labels Record Format

Length, in 16-bit Words	Record Content
1	Record type (114)
1	Record length
1	Process identification number
3	Time stamp
2	Job type/job number
1	Logical device number
1	File sequence number
1	File number (bits 0-7)/Sq type (bits 8, 9) /Type (bits 10, 11)/Reserved (bits 12-15)
1	Not used (bits 0-7)/Volume sequence number (bits 8-15)
1	Expiration date
9	File name/Not used (last 8 bits)
4	Lockword
3	Volume set ID
3	Volume ID
1	PIN *

* PIN of the process making the tape label request.

NOTES:

Sq Type (2 bits)

0	= Search for match on file name
1	= Next or default
2	= Add file to end of volume set
3	= Specified file sequence number

Type (2 bits)

2	= ANSI standard label
3	= IBM standard label

Console log record, type 115

Table 7-21. Console Log Record Format

Length, in 16-bit Words	Record Content
1	Record type (115)
1	Record length
1	Process identification number
3	Time stamp
2	Job type/job number
1	Byte length of console line*
Up to 140	Console input or output line

* If length is less than zero, console message is input. If length is greater than zero, console message is output.

Program file event record, type 116

Table 7-22. Program File Event Record Format

Length, in 16-bit Words	Record Content
1	Record type (116)
1	Record length
1	Process identification number
3	Time stamp
2	Job type/job number
1	Event type
1	Delta P
1	Status
25	Program file name
2	Native mode offset
1	Type (bits 0-3)/Calling location (bits 4-15)

NOTES:

This record appears only during execution of unusual Compatibility Mode code to provide a trap warning, or from the Run-Time Event Monitor (compatibility mode process).

Event Type

<0 = Compatibility mode trap warning
>0 = Run-time event monitor (RTEM) error

Native Mode Offset:

Applicable only for negative event type numbers.

Calling Location

Applicable only for positive event type numbers.

Type

0 = System SL
1 = Pub SL
2 = Group SL
3 = Program file

Bits 4-12 = Segment number

NMS spoolfile done log record, type 120 (input)

Table 7-23. Spoolfile Done Log Record Format (Input)

Length, in 16-bit Words	Record Content
1	Record type (120)
1	Record length
1	Process identification number
3	Time stamp
2	Job type/job number
4	User name
4	Reserved
4	Account name
4	Reserved
4	Job name
4	Reserved
4	File name
4	Reserved
2	Job type/job number
2	Spoolid (all of word 1, bits 0-14 of word 2)/I/O (bit 15 of word 2)
4	Device name
4	Reserved
2	Number of records in spoolfile
2	Number of sectors in spoolfile
1	Device type (bits 0-7)/Device subtype (bits 8-15)
1	Reserved (bits 0=7)/All 0s (bits 8-15)
1	All 0s
1	All 0s (bits 0-7)/Reserved (bits 8-11)/File disposition (bits 12-15)
2	All 0s

NMS spoolfile done log record, type 120 (output)

Table 7-24. Spoolfile Done Log Record Format (Output)

Length, in 16-bit Words	Record Content
1	Record type (120)
1	Record length
1	Process identification number
3	Time stamp
2	Job type/job number
4	User name
4	Reserved
4	Account name
4	Reserved
4	Job name
4	Reserved
4	File name
4	Reserved
2	Job type/job number
2	Spoolid (all of word 1, bits 0-14 of word 2)/I/O (bit 15 of word 2)
4	Device name
4	Reserved
2	Number of records processed
2	Number of sectors in spoolfile
1	Device type (bits 0-7)/Device subtype (bits 8-15)
1	Reserved (bits 0=7)/Output priority (bits 8-15)
1	Current copy number
1	Logical pages per physical page (bits 0-7)/Reserved (bits 8-11)/File disposition (bits 12-15)
2	Number of physical pages

NOTES:

Input spoolfile done log records are generated for every spoolfile that is generated.

One output spoolfile done log record is generated and added to the log records for every file copy (or partial file copy) that is printed.

The top two bits of the job type/job number field are the job type, which refers to the spoolfile, as follows:

00 = Spoolfile originally created by a session on another system or another startup of this system (S')
01 = Spoolfile created by a session on a startup of this system (S)
10 = Spoolfile created by a job on a startup of this system (J)
11 = Spoolfile originally created by a job on another system or another startup of this system (J')

I/O : 0 = Input spoolfile
1 = Output spoolfile

Func : 0 = Normal completion
1 = Delete spoolfile
2 = Defer spoolfile
3 = Relink spoolfile

Number of records processed may be greater than the number of records in the file if internal looping or powerfail occurs.

For serial printers, the number of physical pages is always 0.

Processor launch information log record, type 131

Table 7-25.
Processor Launch Information Log Record Format

Length, in 16-bit Words	Record Content
1	Record type (131)
1	Record length
1	Process identification number
3	Time stamp
2	Job type/job number
8	Processor id
8	Hard physical address
8	Launch status

NOTES:

Each processor (except the one that is used to launch the system) logs a processor launch information record.

Processor id

Identifies the type of processor.

Hard physical address (HFA)

The hard physical address of the processor.

Launch status

`1`	= Can't create interrupt control stack; usually not enough memory.
`-2`	= Processor launch error; a hardware error.
`-1`	= Processor already configured.
`0`	= All OK.

Password changes log record, type 134

Table 7-26. Password Changes Log Record Format

Length, in 16-bit words	Record Content
1	Record type (134)
1	Record length
1	Process identification number
3	Time stamp
2	Job type/job number
8	Target user name
8	Target group name
8	Target account name
1	Type changed
1	Input LDEV number
25	Executed from
3	Reserved
8	User name
8	Group name
8	Account name
8	Job/session name

System logging configuration record, type 135

Table 7-27.
System Logging Configuration Record Format

Length, in 16-bit words	Record Content
1	Record type (135)
1	Record length
1	Process identification number
3	Time stamp
2	Job type/job number
1	(Reserved)
1	LDEV number
4	System logging masking words
8	User name
8	Group name
8	Account name
8	Job/session name

Restore log record, type 136

Table 7-28. Restore Log Record Format

Length, in 16-bit words	Record Content
1	Record type (136)
1	Record length
1	Process identification number
3	Time stamp
2	Job type/job number
8	File name
8	File group
8	File account
8	Creator
17	Volume identification
1	Access type
8	User name
8	Group name
8	Account name
8	Job/session name

Restore log record, type 236

The type 236 record has a “maximum path exceeded” flag.

The type 236 record is only used by native mode RESTORE since compatibility mode RESTORE uses only MPE name syntax. Compatibility mode RESTORE cont inues to use the type 136 record.

Table 7-29 illustrates the 236 record type.

Table 7-29. Record Type 236

Length, in 16-bit words	Record Content
2	Record type 236
2	Event version
2	Record length
2	PIN
4	Job type/Job number
4	Time stamp time
2	Time stamp date
2	Login type (If , logon name is in the format 16-byte user 16-byte acct)
36	Logon name
16	Job/Session name
8	Reserved
16	Creator user name (*user.account* for new file names)
34	Volume ID
2	Access type
2	“Maximum path exceeded” flag
##	Variable-length buffer (file name terminated by a 0)

NOTES:

The restore log record traces file restorations. Files can be restored from tape or serial disk. This logging record can be enabled by SYSGEN, followed by a START command.

Printer access failure log record, type 137

Table 7-30. Printer Access Failure Log Record Format

Length, in 16-bit words	Record Content
1	Record type (137)
1	Record length
1	Process identification number
3	Time stamp
2	Job type/job number
2	Creator job number
8	Creator job name
8	Creator user name
8	Creator account name
25	Spoolfile name
8	Target device name/class
1	(Reserved)
2	File size
1	Status
8	User name
8	Group name
8	Account name
8	Job/session name

NOTES:

This log keeps track of failed attempts attaching spoolfiles to printers. New spoolfiles, which are logged by `FOPEN` as event #144, are not logged here.

This log is initially disabled, but can be enabled by `SYSGEN`, followed by a `START` command.

ACD changes log record, type 138

Table 7-31. ACD Changes Log Record Format

Length, in 16-bit words	Record Content
1	Record type (138)
1	Record length
1	Process identification number
3	Time stamp
2	Job type/job number
25	Target object name
25	Source object name
4	Function
25	Executed from
2	Status
8	User name
8	Group name
8	Account name
8	Job/session name

Type 238

The format of the ACD record has been modified to handle variable-length file na mes and hierarchical pathnames. The target object name and the source object na me fields are variable in length to handle files with expanded names.

Table 7-32 illustrates the format of the type 238 record.

Table 7-32. Record Type 238

Length, in 16-bit words	Record Content
2	Record type 238
2	Event version
2	Record length
2	PIN
4	Job type/Job number
4	Time stamp time
2	Time stamp date
2	Login type (If 1, logon name is in the format 16-byte user 16-byte acct)
36	Logon name
16	Job/Session name
8	Reserved
8	ACD function
4	Status
##	Variable-length buffer

NOTES:

This log type is activated when ACDs are changed (created, deleted, copied, or modified) with MPE commands or intrinsics.

The log can be enabled by `SYSGEN`, followed by a `START` command.

Job stream initiation log record, type 139

Table 7-33. Stream Initiation Log Record Format

Length, in 16-bit words	Record Content
1	Record type (139)
1	Record length
1	Process identification number
3	Time stamp
2	Job type/job number
1	Input LDEV
25	Job file name
2	Job logon Job/session number
8	Job logon user
8	Job logon group
8	Job logon account
8	Job name
2	Input spoolfile ID
1	Scheduled date
2	Scheduled time
8	User name
8	Group name
8	Account name
8	Job/session name

User logging record, type 140

Table 7-34. User Logging Record Format

Length, in 16-bit words	Record Content
1	Record type (140)
1	Record length
1	Process identification number
3	Time stamp
2	Job type/job number
25	Program file name
4	Intrinsic
2	Index
4	LOG ID*
1	Mode
1	Status
8	User name
8	Group name
8	Account name
8	Job/session name

*The LOG ID field in the log record contains "XXXXXX" for the CLOSELOG intrinsic when the index is bad.

NOTES:

The user logging record log keeps track of all OPENLOG and CLOSELOG intrinsic calls. The system manager can use it to see who accesses, or tries to access, the user logging facility.

This log is initially disabled, but can be enabled by SYSGEN, followed by a START command.

Process creation log record, type 141

Table 7-35. Process Creation Record Format

Length, in 16-bit words	Record Content
1	Record type (141)
1	Record length
1	Process identification number
3	Time stamp
2	Job type/job number
25	File name
1	(Reserved)
2	Priority
2	Process Space ID
4	Parent PID
2	NM_Heap_Size
2	Capabilities mask*
8	(Reserved)
8	User name
8	Group name
8	Account name
8	Job/session name

*The capabilities mask is read as follows:

User		File access		Program/group	
bit	capability	bit	capability	bit	capability
0	SM	6	CV	23	BA
1	AM	7	UV	24	IA
2	AL	8	LG	25	PM
3	GL	9	SP	28	MR
4	DI	10	PS	30	DS
5	OP	11	NA	31	PH
		12	NM		
		13	CS		
		14	ND		
		15	SF		

Internal Data Structure, type 242

The data structure of log record type 242 for Security Configuration Changes is as follows:

Table 7-36. Internal Data Structure

Length, in 16-bit words	Record Content
1	Record type (242)
1	Event Version (1)
1	Record length
1	PIN
2	Job type/job number
2	Time stamp time
1	Time stamp date
1	Login type
18	Logon name format
8	Job/session name
8	Future place of the UID & GID
20	Security Configuration feature
20	Old value
20	New Value

Change group record, type 143

Change Group Record Format

Length, in 16-bit Words	Record Content
1	Record type (143)
1	Record length
1	Process identification number
3	Time stamp
2	Job type/job number
8	Old group name
8	New group name
2	CPU time
2	Connect time

File open record, type 144*

Table 7-37. File Open Record Format

Length, in 16-bit Words	Record Content
1	Record type (144)
1	Record length
1	Process identification number
3	Time stamp
2	Job type/job number
8	User name
8	User's account name
8	User's logon group name
2	User ID
8	File name
8	File group name
8	File account name
8	File creator name
10	File UFID
2	File number
1	Foptions
1	File code
1	Reserved (bits 0-5)/Access privileges (bits 6-7)/ Security mask (bits 8-15)
1	Reserved
2	HPE status
2	Object size (current)
2	File Limit
3	File descriptor
7	Reserved

File open record, type 244

The file creator name has been modified to log the "*user.account*" string instead of just the user string for files that are created.

Table 7-38 illustrates the format of record type 244.

Table 7-38. Record Type 244

Length, in 16-bit Words	Record Content
1	Record type 244
1	Event version
1	Record length
1	PIN
2	Job type/Job number
2	Time stamp time
1	Time stamp date
1	Login type (If 1, logon name is in the format 16-byte user 16-byte acct)
18	Logon name
8	Job/Session name
4	Reserved
10	UFID
2	File number
1	Foptions
1	File code
2	File domain
2	File record format
2	File type
6bit	Reserved
2bit	Access priv level
8bit	MPE security mask
2	Open status
2	Object size
2	File limit
3	File descriptor
8	File creator (user.account for new files)
2	“Maximum path exceeded” flag
##	Variable length name buffer (format: file name0)

The file open record is only logged when an error is detected during FOPEN; therefore, the values in the record are not always valid. Specifically, values in the File Limit and Object Size fields are only valid after an error is detected during FOPEN.

Configurable Command Logging

Through the Security Configuration Utility, system managers can specify which MPE commands, when executed, are to be logged by the system logging facility. Log record type 245 will be used to log command execution and its status.

The default setting for this system logging event is "ON".

Table 7-39. Record Type 245

Length, in 16-bit Words	Record Content
1	Record type 245
1	Event version
1	Record length
1	PIN
2	Job type/Job number
2	Time stamp time
1	Time stamp date
1	Login type (If 1, logon name is in the format 16-byte user 16-byte acct)
18	Logon name
8	Job/Session name
8	Reserved
1	CIERR
1	stdin Ldev
14	Program file name or CI
1	Command Length
140	Command image (variable length)

Maintenance Request Record Format, type 146

Table 7-40. Maintenance Request Record Format

Length, in 16-bit Words	Record Content
1	Record type (146)
1	Record length
1	Process identification number
3	Time stamp
2	Job type/job number
20	Device ID
20	PDEV
20	LDEV
2	Format ID
2	Log type
2	Head - reserved
948	Data

NOTES:

Field	Length (Bytes)	Description
Device ID	32	Identifies the device; for example, HP7935 (+8 bytes for Pascal string).
PDEV	32	Physical path to the device; for example, 8.0.0 (+8 bytes for Pascal string).
LDEV	32	Logical device file name (+8 bytes for Pascal string).
Format ID	4	1 = data is HP-IB format 2 = data is FLEX format 3 = data is NIO format
Logtype	4	0 = no data; no logging errors found 101 = data is a run-time error data log record 102 = data is a fault error data log record
Head/Reserved	4	101 = head number 0 or 102 = reserved
Data	1896	Up to 1896 bytes are allowed

Diagnostic information records, type 150

Table 7-41. Auto-Diagnostic Record Format

Length, in 16-bit Words	Record Content
1	Record type (150)
1	Record length
1	Process identification number
3	Time stamp
2	Job type/job number
2	Type number
16	Hardware product number
16	Physical path description
16	Logical device name
1	Device class identifier
50	Diagnostic messages

NOTES:

There are two different formats for type 150 diagnostic information records: one is the auto-diagnostic record format and the other is the diagnostic system information record format. You can determine which format is used by looking at the type number field just after the header.

Type Number

0 = Auto-diagnostic record format
3 = Diagnostic system information record format

In the diagnostic messages field, a continuation flag indicates whether another related message was placed into the log file.

Diagnostic system information record, type 150

Table 7-42. Diagnostic System Information Record Format

Length, in 16-bit Words	Record Content
1	Record type (150)
1	Record length
1	Process identification number
3	Time stamp
2	Job type/job number
2	Type number
Up to 128	Diagnostic system messages or status information

NOTES:

There are two different formats for type 150 diagnostic information records: one is the auto-diagnostic record format and the other is the diagnostic system information record format. You can determine which format is used by looking at the type number field just after the header.

Type Number

0 = Auto-diagnostic record format
3 = Diagnostic system information record format

The diagnostic system information can include information about which users requested single-user mode (SUM). It can also record internal diagnostic system errors detected when no user exists to report them to. This can occur when an I/O error is logged, for example.

High-priority machine check, type 151

Table 7-43. High-Priority Machine Check Record Format

Length, in 16-bit Words	Record Content
1	Record type (151)
1	Record length
1	Process identification number
3	Time stamp
2	Job type (bits 0-1)/job number (bits 2-15 of word 1, bits 0-15 of word 2)
2	Hardware ID
2	Number of bytes in record
2	GR 0
2	:
2	GR31
2	CR0
2	:
2	CR31
2	SR0
2	:
2	SR7
2	Interrupting instruction address space
2	Interrupting instruction address offset
2	0 Check type word (bits 0-4 of word 1)/Reserved (bits 5-15 of word 1, bits 0-15 of word 2)
2	0 CPU state word (bits 0-5 of word 1)/Reserved (bits 6-15, bits 0-12 of word 2)/Past OK (bit 13 of word 2)/Error cleared (bits 14-15 of word 1, all word 2)
2	0 Detected by word (bits 0-2 of word 1)/Reserved (bits 3-15 of word 1, all word 2)
2	0 Cache check word (bits 0-3 of word 1)/Reserved (bits 4-15 of word 1, all word 2)
2	0 TLB check word (bits 0-4 of word 1)/Reserved (bits 5-15 of word 1, all word 2)
2	0 Bus check word (bits 0-9 of word 1)/Reserved (bits 10-15, all word 2)
2	0 Assist check word (bits 0-2 of word 1)/Reserved (bits 3-15 of word 1, all word 2)
2	0 Processor check word - reserved
2	Reserved (all word 1, bits 0-12 of word 2)/Assist ID word (bits 13-15 of word 2)
Varies	System-dependent portion of PIM

NOTES:

Fields starting with GR0 to the end of the record contain processor internal memory (PIM). Record type 151 contains HPMC PIM and record type 152 contains LPMC.

The first PIM fields contain information from the registers (such as general registers, GR0-31, and control registers, CR0-31). The length of the last PIM field (system-dependent portion of PIM) is hardware dependent. However, the total length of a logging record is restricted to 2KB. Refer to the appropriate hardware manuals for more information.

The nature of a high-priority machine check is passed by setting nonzero values in the appropriate fields of the record. The layout of these error parameters is as follows:

Check type word

Bits	Definition
0:1	Cache system check
1:1	TLB check
2:1	Bus transaction check
3:1	Assists check
4:1	Processor internal check

CPU state word

Bits	Definition
0:1	Interrupting instruction address queue valid
1:1	Interrupting instruction address queue fault
2:1	IPRs valid
3:1	General registers valid
4:1	Control registers valid
5:1	Space registers valid

Detected by word

Bits	Definition
0:1	Instruction fetch
1:1	Load
2:1	Load and clear
3:1	Store
4:1	Flush I-cache
5:1	Flush D-cache
6:1	Purge D-cache
7:1	Copyout of dirty cache line
8:1	Instruction prefetch
9:1	Data prefetch
10:1	Remote cache consistency check
11:1	Local purge TLB
12:1	Remote purge TLB
13:1	Probe read access
14:1	Probe write access
15:1	Coprocessor operation
16:1	SFU operation
17:1	Insert I or D TLB protection or access

Cache check word

Bits	Definition
0:1	I-cache check
1:1	D-cache check
2:1	Tag check
3:1	Data check

TLB check word

Bits	Definition
0:1	I-TLB check
1:1	D-TLB check

Bus check word

Bits	Definition
0:1	Address error
1:1	Data slave error
2:1	Data parity error
3:1	Data protocol error
4:1	Read transaction
5:1	Write transaction
6:1	Memory space transaction
7:1	I/O space transaction
8:1	Processor was master in transaction
9:1	Processor was slave in transaction

Assists check word

Bits	Definition
0:1	Coprocessor check
1:1	SFU check
2:1	Assist ID valid

Processor check word

All fields of this word are currently *reserved*.

Assist ID word

The 3-bit unit ID field of the failing SFU or coprocessor is stored right-justified in bits 29:3 of the assist ID word.

Low-priority machine check, type 152

Table 7-44. Low-Priority Machine Check Record Format

Length, in 16-bit Words	Record Content
1	Record type (152)
1	Record length
1	Process identification number
3	Time stamp
2	Job type (bits 0-1)/job number (bits 2-15 of word 1, bits 0-15 of word 2)
2	Hardware ID
2	Number of bytes in record
2	GR 0
2	:
2	GR31
2	CR0
2	:
2	CR31
2	SR0
2	:
2	SR7
2	Interrupting instruction address space
2	Interrupting instruction address offset
2	0 Check type word (bits 0-4 of word 1)/Reserved (bits 5-15 of word 1, bits 0-15 of word 2)
2	0 CPU state word (bits 0-5 of word 1)/Reserved (bits 6-15, bits 0-12 of word 2)/Past OK (bit 13 of word 2)/Error cleared (bits 14-15 of word 1, all word 2)
2	0 Detected by word (bits 0-2 of word 1)/Reserved (bits 3-15 of word 1, all word 2)
2	0 Cache check word (bits 0-3 of word 1)/Reserved (bits 4-15 of word 1, all word 2)
2	0 TLB check word (bits 0-4 of word 1)/Reserved (bits 5-15 of word 1, all word 2)
2	0 Bus check word (bits 0-9 of word 1)/Reserved (bits 10-15, all word 2)
2	0 Assist check word (bits 0-2 of word 1)/Reserved (bits 3-15 of word 1, all word 2)
2	0 Processor check word - reserved
2	Reserved (all word 1, bits 0-12 of word 2)/Assist ID word (bits 13-15 of word 2)
Varies	System-dependent portion of PIM

NOTES:

The fields in this record are the same as the corresponding fields in the high-priority machine check record (type 151). See the notes following the record.

CM file close record, type 160

Table 7-45. CM File Close Record Format

Length, in 16-bit Words	Record Content
1	Record type (160)
1	Record length
1	Process identification number
3	Time stamp
2	Job type/job number
13	File name - format of fname.group.acct
1	Reserved
1	Disposition (bits 0-7)/Domain (bits 8-15)
2	Number of sectors allocated
1	Device type (bits 0-7)/Unused (bits 8-15
2	Number of records transferred
2	Number of blocks transferred
1	Logical device number

NOTES:

Close Disposition (bits 13:3)		Open Domain	
0	No change	0	New file
1	Save permanent	1	Old permanent file
2	Save temporary - rewound	2	Old temporary file
3	Save temp - not rewound	3	Old job or sys
4	Delete		
5	Make temporary		

Disk Space Disposition (bits 11:2)

0	Do not return disk space allocated beyond EOF.
1	Return disk space allocated beyond EOF to system; EOF becomes file limit.
2	Return disk space allocated beyond EOF to system; file limit remains the same (NM file types only).

The Disposition field in a CM file close record can have values from 0-4, 8-12, and 255. Any files left open during process termination are closed by the system and given a disposition of 255.

8

Using the Security Configurator (SECCONF)

Overview

This chapter describes the Security Configurator (SECCONF), a tool used to configure the security features provided by the HP Security Monitor.

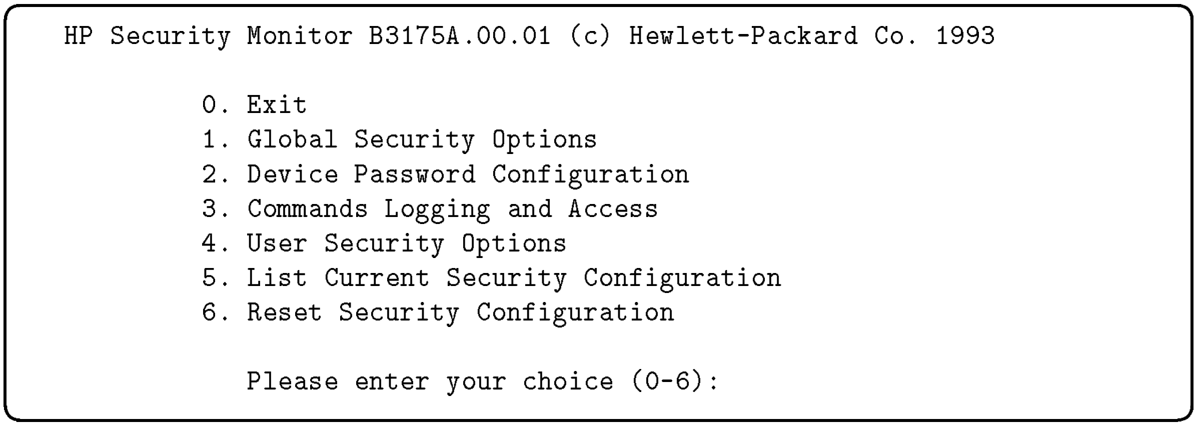
The Security Configuration Utility, SECCONF.PUB.SYS, is a program that can be run by a user with SM capability and logged on to the SYS account. SECCONF is used to establish or modify system global security information. It creates/updates the file SECDATA.PUB.SYS and copies the data from this file to the global system security information table.

Running the Security Configurator (SECCONF)

To enter the Security Configurator, at the MPE/iX prompt enter:

```
:RUN SECCONF.PUB.SYS
```

After verifying the user's capability, the program presents the user with the main menu:



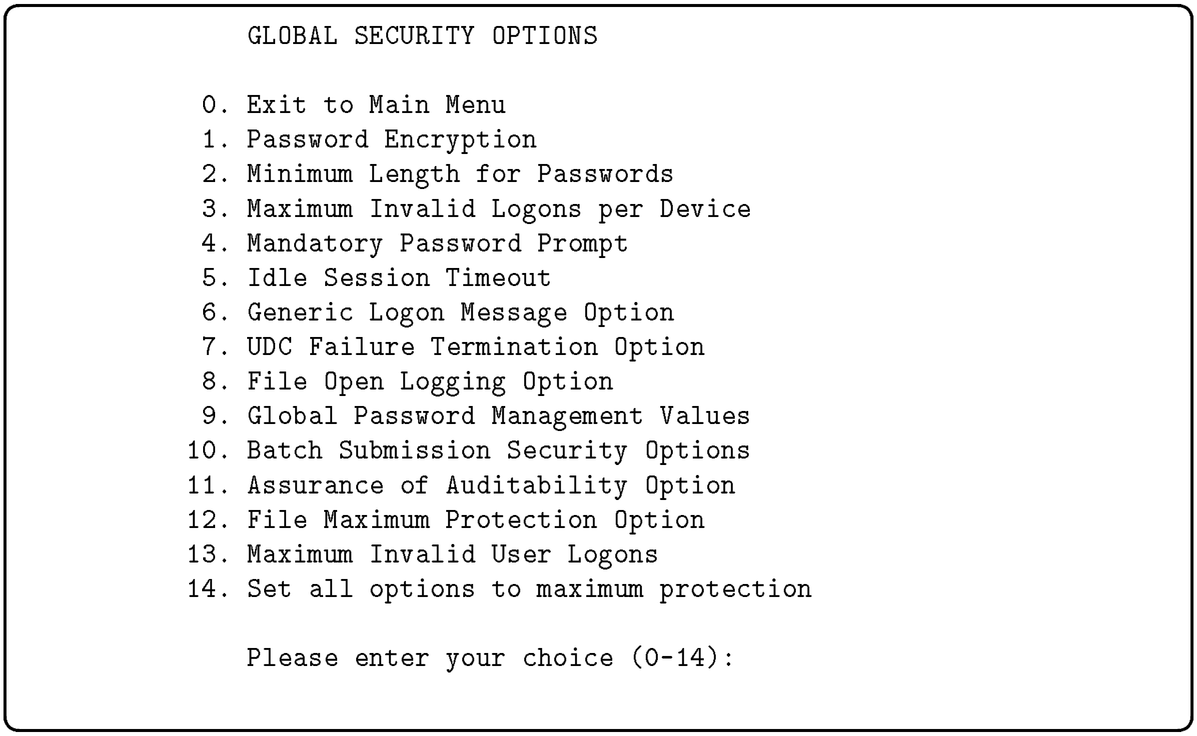
```
HP Security Monitor B3175A.00.01 (c) Hewlett-Packard Co. 1993

         0. Exit
         1. Global Security Options
         2. Device Password Configuration
         3. Commands Logging and Access
         4. User Security Options
         5. List Current Security Configuration
         6. Reset Security Configuration

            Please enter your choice (0-6):
```

After the user is finished with any function, the program always exits to a higher level menu. The user will then have a choice of exiting or going to another menu selection.

At the main menu level, if the EXIT option is specified, the program will update both the security data file and the security information table. At this time the new configuration will immediately take effect. (All changes made during this session will be logged in the system log file and generate a message on the system console).

Global Security Options

If the user selects the “Global Security Options” in the main menu, the “Global Security Options” menu will be displayed.

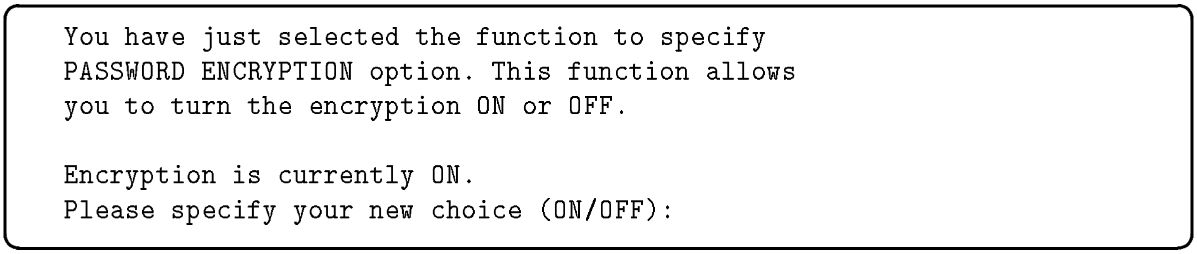


```
                GLOBAL SECURITY OPTIONS

             0. Exit to Main Menu
             1. Password Encryption
             2. Minimum Length for Passwords
             3. Maximum Invalid Logons per Device
             4. Mandatory Password Prompt
             5. Idle Session Timeout
             6. Generic Logon Message Option
             7. UDC Failure Termination Option
             8. File Open Logging Option
             9. Global Password Management Values
            10. Batch Submission Security Options
            11. Assurance of Auditability Option
            12. File Maximum Protection Option
            13. Maximum Invalid User Logons
            14. Set all options to maximum protection

                Please enter your choice (0-14):
```

For each of the choices, the program will display the current option or value (which may be the default if it was not configured), and prompts the user for the new value.

1. Password Encryption

This option produces following screen:



```
You have just selected the function to specify
PASSWORD ENCRYPTION option. This function allows
you to turn the encryption ON or OFF.

Encryption is currently ON.
Please specify your new choice (ON/OFF):
```

2. Minimum Length for Passwords

This option produces the following screen:

```
This function allows the user to specify the value for the minimum
password length. This value is applicable to USER, ACCOUNT, GROUP and
device passwords (but not lockwords).

Minimum password length currently is: 0
Please enter your choice (0-8):
```

3. Maximum Invalid Logons per Device

This option produces the following screen:

```
This function allows you to specify the maximum number
of invalid logons that a device can tolerate before it
becomes unavailable (DOWNed). You can specify a number
between 1 and 32766, or a zero (0) which is unlimited.

Maximum invalid attempts currently is: 4
Please enter your choice (0-32766):

Since you just set the maximum invalid logon count, you
may want to configure a timeout interval, during which a
violated device would be in the DOWN state. After that
interval expires, MPE will automatically UP the device.
The interval can be from 1 to 32766 seconds, or Zero (0).
A zero means there is no automatic timeout, and the
Operator must :UP the device manually.

The currently configured device timeout is: 0
Please enter your choice (0-32766):
```

4. Mandatory Password Prompt

This option produces the following screen:

```
You have just selected the function to specify whether
password prompts are mandatory for interactive logons.
When this option is ON, MPE will not accept logon
commands with embedded passwords, such as

:HELLO USER/UPASS.ACCT/APASS

Mandatory Password Prompt is currently OFF (i.e. embedded passwords OK).
Please specify your new choice (ON/OFF):ON

Mandatory Password Prompt is now ON.

Since you've just enabled the mandatory password prompt option, you may
want to consider exempting REMOTE HELLO from this rule. You might have
applications which log on remotely from within a job, or do REMOTE HELLO
programmatically. These applications will fail if the DS/NS terminals are
not exempted.

Do you want to exempt DS/NS terminals from forced prompting (Yes/No) ?yes

DS/NS terminals are now EXEMPT.


Do you want to exempt DS/NS terminals from forced prompting (YES/NO)
```

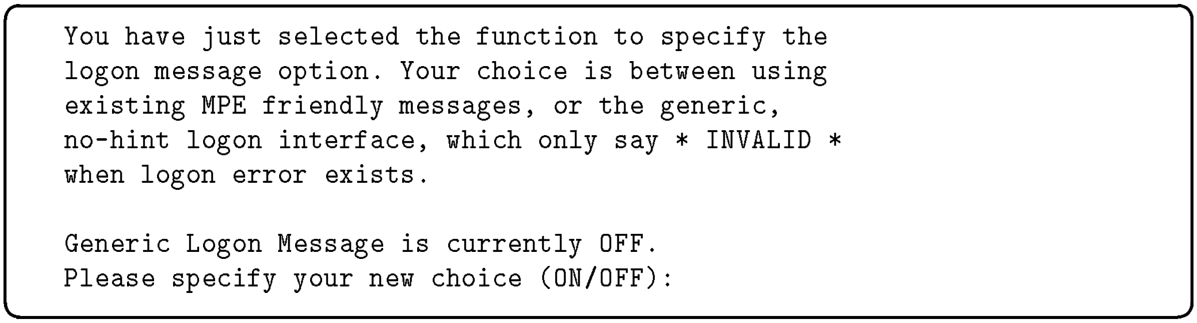
5. Idle Session Timeout

This function allows you to specify the system-wide timeout value for CI reads and application timeout. When a non-zero value is specified, all idle sessions will be terminated after that time expires without a user response. A zero (0) means no timeout.

```
The Global Timeout value currently is: 0
Please enter your choice (0 - 546):
```

6. Generic Logon Message Option

This option produces the following screen:



```
You have just selected the function to specify the
logon message option. Your choice is between using
existing MPE friendly messages, or the generic,
no-hint logon interface, which only say * INVALID *
when logon error exists.

Generic Logon Message is currently OFF.
Please specify your new choice (ON/OFF):
```

7. UDC Failure Termination Option

This option produces the following screen:

```
You have just selected the function to specify whether
a job/session is to be terminated if the initiation of
UDC at logon time fails for any reason.  When this
option is ON, it prevents users from bypassing your
system logon UDC.

UDC Failure Termination is currently OFF (disabled).
Please specify your new choice (ON/OFF):
```

8. File Open Logging Option

This option produces the following screen:

```
You have just selected the function to specify whether
to log all FOPENs or only the ones that failed.
FOPEN LOGGING OPTION is currently: FOPEN FAILURE ONLY
Do you want to keep this option: 'logging FOPEN FAILURES ONLY' (Yes/No):
```

9. Global Password Management Values

The Global Password Management Values can be configured with the Security Options menu, by selecting item 9. By entering the desired values, a system administrator may set a global password expiration date, and the time intervals for a global password aging policy.

After selecting the Global Password Management Values, the security administrator is presented with the following menu:

```
               GLOBAL PASSWORD MANAGEMENT VALUES

            0. Exit to Global Security Menu
            1. Global User Password Expiration Date
            2. Global User Password Maximum Lifetime
            3. Global User Password Minimum Time
            4. Global User Password Warning Time
            5. Global User Password Expiration Time

               Please enter your choice (0-5):
```

(1.) Global User Password Expiration Date.

This option produces the following screen:

```
The Global Password Expiration function allows you to
activate automatic password expiration for all users
who are required to have password. When this option
is enabled, MPE will expire all the REQUIRED user
passwords on the (same) global expiration date.

To enable this function, you specify the number of
days between expirations. This value can range
from 1 to 365; or a zero (0) which means no automatic
expiration.  Optionally, you can specify a date for the
expiration cycle to start (default is today); and the
number of days to warn the user prior to expiration.

Automatic global password expiration has not been configured.

Please specify your new choice (0-365 days):
Number of days between expirations is now: 365

Since you have just set the number of days for
automatic expiration, you may want to set or reset a
specific date to start the expiration cycle.

The next global expiration date is: today (by default).

To change, enter new MM/DD/YY (CR = no change):01/01/93

The global expiration date accepted is: FRI, JAN 1, 1993

Number of days to warn prior to expiration currently is: 0

Please specify your new choice (0-364 days):5
Number of days to warn prior to expiration is now: 5
```

The following options, affect system wide password aging policy for all users.

(2.) Global User Password Maximum Lifetime.

This option produces the following screen:

```
This option sets the maximum lifetime for a user password.
This value can range from 1 to 365 days, or optionally 0
for no password expiration.

The global user password maximum lifetime currently is: 0
Please enter your choice (0-365 days):90

The global user password maximum lifetime is now: 90
```

(3.) Global User Password Minimum Time.

This option produces the following screen:

```
This option sets the minimum time after setting a
password before the password can be changed.  This
value can range from 1 to 364 days, or optionally 0 for no
minimum password time.

The global user minimum lifetime currently is: 0
Please enter your choice (0-364 days):5

The global user password minimum time is now: 5
```

(4.) Global User Password Warning Time.

This option produces the following screen:

```
This option sets the time a user is warned before the
user password expires.  This value can range from 1 to 364
days, or optionally 0 to prevent warning.

The global user warning time currently is: 0
Please enter your choice (0-364 days):5

The global user password warning time is now: 5
```

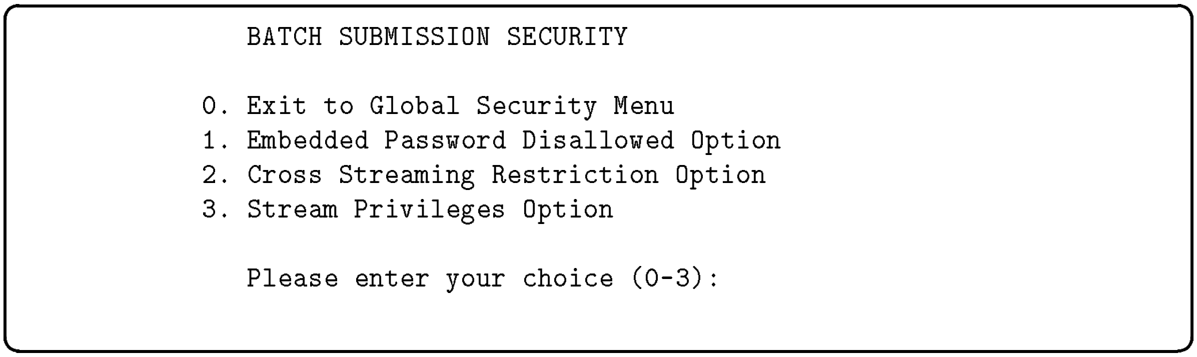
(5.) Global User Password Expiration Time.

This option produces the following screen:

```
This option sets expiration period for a user password.
When this time period expires, the user is no longer
able to log on to the system.  This value can range
from 1 to 364 days, or optionally 0 for no password
expiration.

The global user expiration time currently is: 0
Please enter your choice (0-364 days):5

The global user password expiration time is now: 15
```

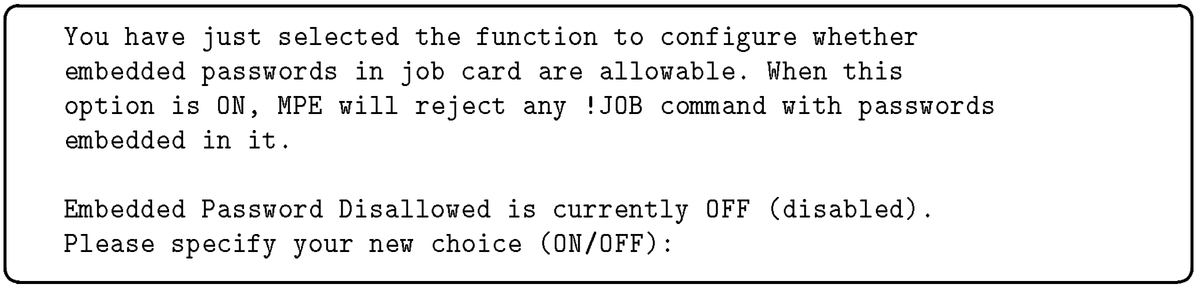
10. Batch Submission Security Options

The Batch Submission Security Options can be configured with the Global Security Option menu, by selecting item 10. From the Batch Submission Security menu, the security administrator will be able to select various options. Selecting the Batch Submission Security Options from the Global Security Options menu, the following screen is displayed:



```
               BATCH SUBMISSION SECURITY

            0. Exit to Global Security Menu
            1. Embedded Password Disallowed Option
            2. Cross Streaming Restriction Option
            3. Stream Privileges Option

               Please enter your choice (0-3):
```

Embedded Password Disallowed Option.
This option produces the following screen:



```
   You have just selected the function to configure whether
   embedded passwords in job card are allowable. When this
   option is ON, MPE will reject any !JOB command with passwords
   embedded in it.

   Embedded Password Disallowed is currently OFF (disabled).
   Please specify your new choice (ON/OFF):
```

Cross Streaming Restriction Option.

This option produces the following screen:

```
You have just selected the function to configure whether
streaming of other people's jobs is allowable.  When this
option is ON, a person will not be allowed to stream another
person's job, unless specifically authorized.

Cross Streaming Restriction is currently OFF (i.e., cross stream allowed).
Please specify your new choice (ON/OFF):ON

Cross Streaming Restriction is now ON (Enabled).

Since you have just enabled the cross streaming restriction
feature, you may want to also enable the Authorization option
to allow limited cross streaming of protected jobs.  When
enabled, this option allows those with EXECUTE access to
"protected" job files to stream them.

Cross Streaming Authorization is currently OFF (Disabled)
Please specify your new choice (ON/OFF):
```

Stream Privileges Option.

This option produces the following screen:

```
You have just selected the function to configure whether
SM, AM and a job owner is allowed to stream jobs without
the need for passwords.

Stream Privilege is currently OFF (disabled).
Please specify your new choice (ON/OFF):ON

Stream Privilege in now ON (Enabled)

Since you have just enabled the stream privilege feature,
you may want to extend this privilege to other users to
allow limited password omission in streaming of protected
jobs.  This extension allows those with EXECUTE access to
"protected" job files to stream them without passwords.

Stream Privilege Authorization is currently OFF (Disabled)
Please specify your new choice (ON/OFF):
```

11. Assurance of Auditability Option

This option produces the following screen:

```
You have just selected the function to configure whether
the auditability is to be assured when there is a system
logging error.  When this option is ON, MPE will execute
a CONTROL-A LOGOFF if a system logging error occurs.  At
that time, only users with OP or SM capability can log on
to the  system and try to correct the problem  and resume
logging or shutdown the system.

Assurance of Auditability is currently OFF
Please specify your new choice (ON/OFF):
```

12. Maximum Protection Option

This option produces the following screen:

```
You have just selected the function to configure whether
a NEWLY created object is going to be maximally protected.
When this option is ON, MPE will configure the CREATOR of
the object to be the only user who can access the object
if no ACD is attached to that object.

Maximum Protection is currently OFF
Please specify your new choice (ON/OFF):
```

13. Maximum Invalid User Logons

This option produces the following screen:

```
This function allows you to specify the maximum number
of invalid user logons before the user ID becomes invalid.
You can specify a number between 1 and 32766, or zero (0)
which is unlimited.

Maximum invalid attempts currently is: 4
Please specify your new choice (0-32766):

Since you just set the maximum invalid logon count,
you may want to configure a timeout interval, during
which a user ID will remain invalid.  After that time
interval expires, MPE will automatically change the user
ID to valid.  The interval can be from 1 to 32766
seconds, or zero (0).  A zero means there is no automatic
timeout, and the account manager or system manager must
re-activate the user.

The timeout is currently : 0

Please enter your choice (0-32766)
```

14. Set All Options To Maximum Protection

This option provides the following screen:

```
This function allows you to set the maximum protection for all
of the Global Security Options.  Use the LIST command to verify
the selections are acceptable.

Do you want to set all Global Security Options to maximum
protection (YES/NO):YES

Maximum protection has been set to maximum.
```

Device Password Configuration

For the “Device Password Configuration” selection, the program will let the user enter the device number and password for that device. Multiple groups of classes or LDEV’s can be entered on one line, separated by a comma, providing the line does not exceed 72 characters. The same password will be assigned to all LDEV’s or device classes on that line. LDEV’s and device classes can not be mixed on the same line. This option produces the following screen:

```
This function allows you to configure the device password
for terminals. Following the ">" prompt, please enter the
"LDEV;PASSWORD" that you want to configure in, for example:
                > 20; SECRET
         or     > 21,22,23; DEVPASS
         or     > TERM; TERMPASS   (TERM is a device class)

The prompt will be repeated until you end your input
with a "//" or a CR only.  If you want a list of currently
configured (passworded) devices, enter "@".  To remove
password for any LDEV, enter a blank/empty password, e.g. "21; ".

If you want your password input echo-suppressed, enter only
the LDEVs, then we will prompt you for the password with
the echo turned off.

Please enter password information in the form
LDEV [,ldev...][;PASSWORD].(up to a maximum of 72 characters)

If user enters "@" at the prompt, the format of the
display will be:

The following Ldevs have a device password:
68, 70, 71, 72, 73, 74, 75, 77, 78, 79, 200, 201, 202, 204,
205, 206, 207, 208, 209, 210, 69
```

If the user enters 71,72; the format of the display will be:

```
Removing Device Password for ldev 71
Removing Device Password for ldev 72
```

Commands Logging and Access

For the “Commands Logging and Access” category, the following display will appear:

```
                COMMANDS LOGGING & ACCESS

             0. Exit to Main Menu
             1. Configure Logging & Disabling
             2. Set Programmatic Access Level

                Please enter your choice (0-2):
```

Configure Logging & Disabling

This option produces the following display:

```
Following the "Command >" prompt, please enter the
MPE COMMAND that you want to log or disable access.
Once the command is verified to be a valid command,
you will be asked for the logging and access options
for that command.

To terminate you input, enter "//" or a carriage return in response
to the "command>" prompt.

If you want a list of all commands that are currently
configured in the security table, enter "@".

Command > CONSOLE

CONSOLE    Prog. Access ON    General Execution ON  Logging OFF

        Programmatic Access Disabled (Yes/No) ?YES
        General Execution Disabled (Yes/No) ?YES
        Logging Enabled (Yes/No) ?YES

CONSOLE     Prog. Access OFF   General Execution OFF  Logging ON
All Others:        Prog. Access ON    General Execution ON   Logging OFF
Command>
```

If the user requests the list of configured commands by entering "@", the following screen will be displayed:

```
Command > @

CONSOLE    Prog. Access OFF   General Execution OFF    Logging ON
PURGEACCT  Prog. Access OFF   General Execution OFF    Logging ON
ALTSEC     Prog. Access OFF   General Execution OFF    Logging ON
```

Set Programmatic Access Level

This option produces the following screen:

```
Since Command disabling may affect the functioning of
your present programs and subsystems, you may want to
set the Programmatic Access to WARNING level first.

At warning level, a command, when executed
programmatically, will only cause the command to be
logged via the Command Logging facility and a message to be sent to
$STDLIST.
Then, after you have examined the system log files to
ascertain that the disabled commands do not adversely
affect your applications, you can reset the WARN flag
to go to full disabling level (execution causes an error).

PROGRAMMATIC ACCESS LEVEL is currently: FULL DISABLING.

Do you want to change this to 'WARNING LEVEL' (Yes/No) ?
```

User Security Options

This option produces the following screen:

```
         USER SECURITY OPTIONS

      0. Exit
      1. Enable User ID
      2. Enable User Password
      3. User Password Aging Values
      4. Set User Passwords Required
      5. Remove User Passwords Required

         Please enter your choice (0-5):
```

Once the choice 1 or 2 is made, the system manager will then be able to manipulate the user information.

1. Enable User

This selection will allow a system to enable a user that has been disabled and will display the following screen:

```
You have selected the choice to enable a disabled user ID.
The prompt will be repeated until you enter a (user.account) name
or end your input with a "//" or carriage return.

Enter the user ID to enable (user.account):mgr.test
Enable mgr.test (YES/NO):YES
User  mgr.test  has been enabled
```

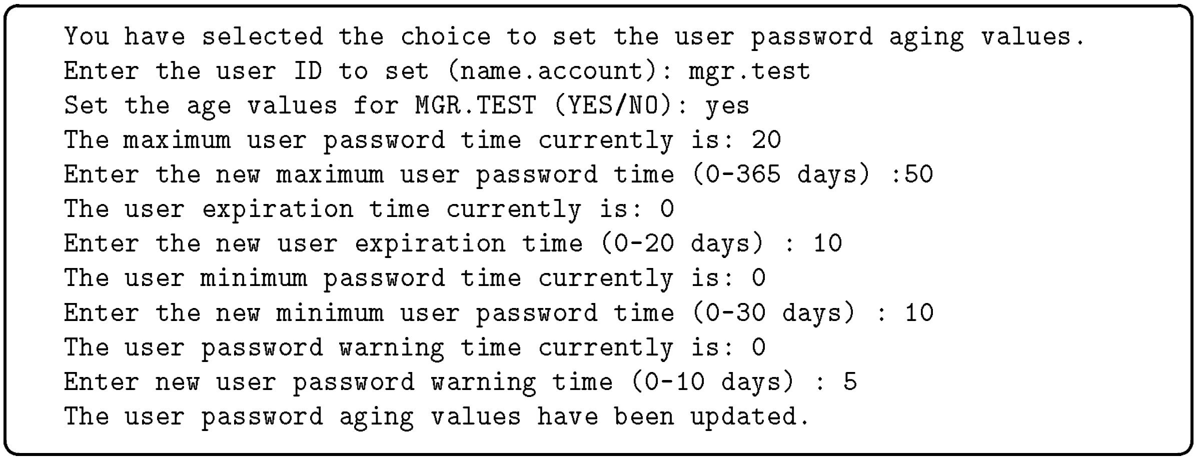
2. Enable User Password

This solution will allow a system manager to set an invalid user password to theexpired state. An invalid user password is one that went beyond the expiration time. Once the user password is in the expired state, the password can be changedby the user.

```
You have selected the choice to enable an invalid user password.
The prompt will be repeated until you enter a (user.account) name
or end your input with a "//" or a carriage return.
The password will be set expired.
Enter the invalid user ID (Name.Account):
Enable the password for Name.Account (Yes/No):
User Name.Account is now expired
```

3. User Password Aging Values

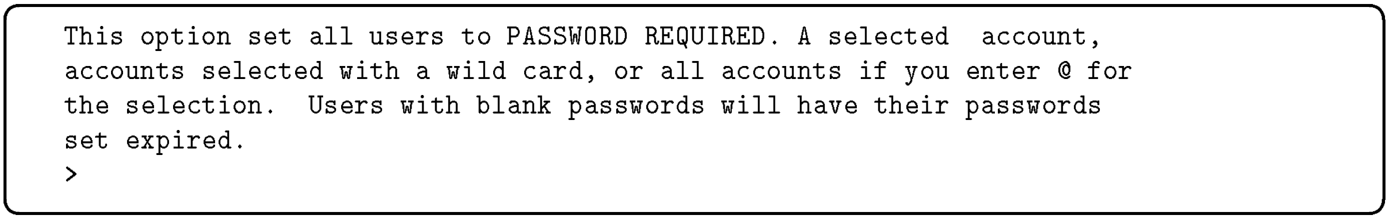
This selection describes the user level password aging values which allow System Managers to set the user password aging values for a specific user. Aging values for individual users can be established only after the system wide password aging policy is established. The dialog for the expiration, minimum and warning times will only occur if the maximum time is not set to zero. The selected times are checked in order to maintain consistency with the password aging scheme.



```
You have selected the choice to set the user password aging values.
Enter the user ID to set (name.account): mgr.test
Set the age values for MGR.TEST (YES/NO): yes
The maximum user password time currently is: 20
Enter the new maximum user password time (0-365 days) :50
The user expiration time currently is: 0
Enter the new user expiration time (0-20 days) : 10
The user minimum password time currently is: 0
Enter the new minimum user password time (0-30 days) : 10
The user password warning time currently is: 0
Enter new user password warning time (0-10 days) : 5
The user password aging values have been updated.
```

4. Set User Passwords Required

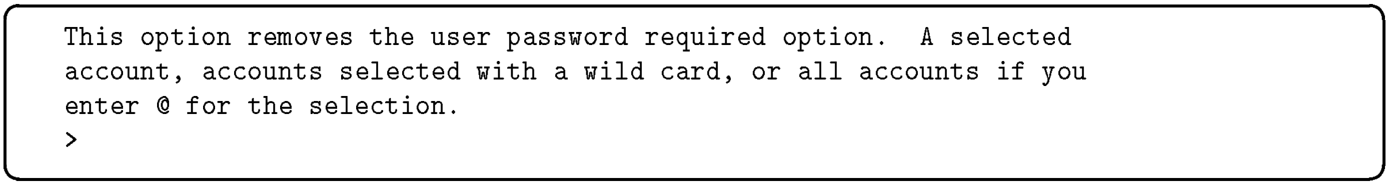
This option is used to specify that passwords are to be required for a specific account.



```
This option set all users to PASSWORD REQUIRED. A selected  account,
accounts selected with a wild card, or all accounts if you enter @ for
the selection.  Users with blank passwords will have their passwords
set expired.
>
```

5. Remove User Passwords Required

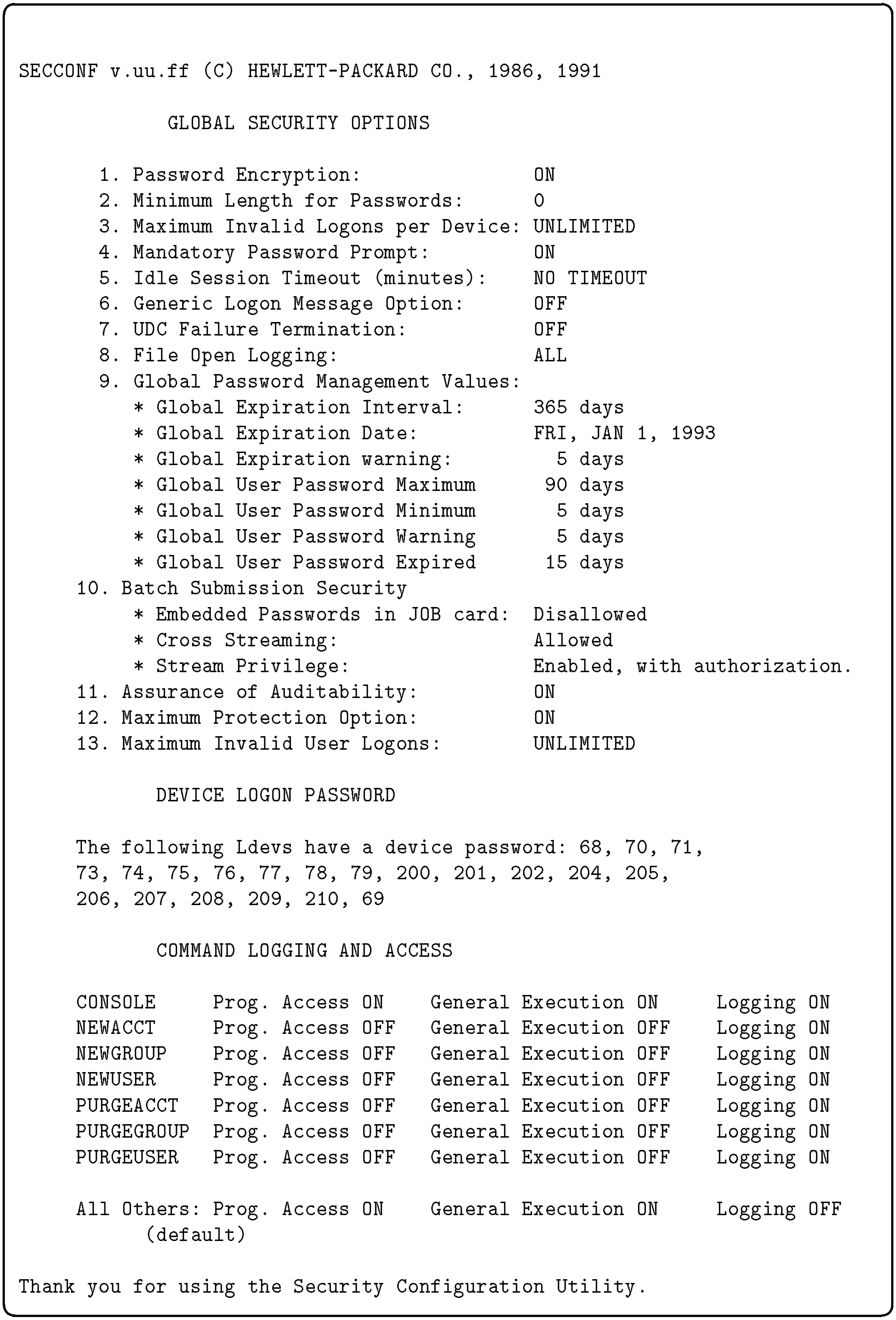
This option is used to remove required password protection from specified accounts.



```
This option removes the user password required option.  A selected
account, accounts selected with a wild card, or all accounts if you
enter @ for the selection.
>
```

List Current Security Configuration

This option produces the following screen:



```
SECCONF v.uu.ff (C) HEWLETT-PACKARD CO., 1986, 1991

             GLOBAL SECURITY OPTIONS

       1. Password Encryption:               ON
       2. Minimum Length for Passwords:      0
       3. Maximum Invalid Logons per Device: UNLIMITED
       4. Mandatory Password Prompt:         ON
       5. Idle Session Timeout (minutes):    NO TIMEOUT
       6. Generic Logon Message Option:      OFF
       7. UDC Failure Termination:           OFF
       8. File Open Logging:                 ALL
       9. Global Password Management Values:
          * Global Expiration Interval:      365 days
          * Global Expiration Date:          FRI, JAN 1, 1993
          * Global Expiration warning:         5 days
          * Global User Password Maximum      90 days
          * Global User Password Minimum       5 days
          * Global User Password Warning       5 days
          * Global User Password Expired      15 days
     10. Batch Submission Security
          * Embedded Passwords in JOB card:  Disallowed
          * Cross Streaming:                 Allowed
          * Stream Privilege:                Enabled, with authorization.
     11. Assurance of Auditability:          ON
     12. Maximum Protection Option:          ON
     13. Maximum Invalid User Logons:        UNLIMITED

            DEVICE LOGON PASSWORD

     The following Ldevs have a device password: 68, 70, 71,
     73, 74, 75, 76, 77, 78, 79, 200, 201, 202, 204, 205,
     206, 207, 208, 209, 210, 69

            COMMAND LOGGING AND ACCESS

     CONSOLE     Prog. Access ON    General Execution ON     Logging ON
     NEWACCT     Prog. Access OFF   General Execution OFF    Logging ON
     NEWGROUP    Prog. Access OFF   General Execution OFF    Logging ON
     NEWUSER     Prog. Access OFF   General Execution OFF    Logging ON
     PURGEACCT   Prog. Access OFF   General Execution OFF    Logging ON
     PURGEGROUP  Prog. Access OFF   General Execution OFF    Logging ON
     PURGEUSER   Prog. Access OFF   General Execution OFF    Logging ON

     All Others: Prog. Access ON    General Execution ON     Logging OFF
           (default)

Thank you for using the Security Configuration Utility.
```

Users can also run SECCONF with the LIST entry point to obtain security configuration information. On MPE/iX command prompt, enter:

```
:RUN SECCONF.PUB.SYS ; INFO = 'LIST'
```

Output similar to the one described on the previous page, will be displayed on the user terminal.

Reset Security Configuration

This option produces a display that allows users to reset all or part of the current security configuration:

```
SECCONF v.uu.ff (C) HEWLETT-PACKARD CO., 1986, 1992


              RESET MENU

     0. Exit
     1. Hard Reset
     2. Soft Reset - Reset Global Options
     3. Soft Reset - Reset Command Options
     4. Soft Reset - Reset Device Passwords
     5. Soft Reset - Reset User Options
     6. Suspend    - Suspend Command Disabling

        Please enter your choice (0-6):__
```

For each of the choices, the program will display a short warning describing the function of each reset. The user will then be asked if they want to continue.

Hard Reset

This option will remove all of the new security features from the system (and clear the SECDATA file). It will also replace ALL encrypted passwords with a blank password and will remove all encrypted, required, expired and warning bits from the system directory. It will essentially return the system to a pre-security state.

One thing the Hard Reset will not do is to remove the new logging bits from the system tables and from the system CONFIG file.

Soft Reset - Reset Global Options

This option will reset or turn off all of the features associated with the Global Security Options menu.

Soft Reset - Reset Command Options

This option will reset or turn off all of the features associated with the Commands Logging and Access Menu. All commands will be re-enabled and all command logging will be turned off. In addition, the warning level option will be reset to the non-warning state.

Soft Reset - Reset Device Passwords

This option will reset or turn off all of the features associated with the Device Password Configuration menu. All device passwords will be reset to blanks. Devices will no longer have passwords.

Soft Reset - Reset User Options

This option will reset or turn off all of the security features associated with the User Security Options menu.

Suspend - Suspend Command Disabling

This option will temporarily suspend the command disabling feature while saving the disabled command configuration. Upon activating this option, all command disabling will be suspended. Command disabling is re-enabled by re-running SECCONF or by re-booting the system.

Note

Users can also run SECCONF with the RESET entry point to reset the security configuration. On MPE/iX command prompt, enter:

```
:RUN SECCONF.PUB.SYS ; INFO = 'RESET'
```

The RESET MENU will be displayed on the user terminal.

A

The Security Maintenance Checklist

This checklist is provided to assist FOS security users in reviewing account and system security.

1. □ Do all accounts have passwords?
2. □ Have all default passwords been changed?
3. □ Are there procedures to ensure quarterly system password changes?
4. □ Are passwords changed when employees leave the organization?
5. □ Do special capability users (PM, SM, OP, AM, NM, and NA) have user passwords?
6. □ Are user passwords unique in accounts accessible by more than one person?
7. □ Is SM capability restricted to one person per system and AM capability to one person per account?
8. □ Do all groups with PM have restricted save access (`S=GU`)?
9. □ Are programs protected from unpriviledged users?
10. □ Is there an updated list of all released files?
11. □ Is there a logon or `NOBREAK` UDC at system and account level to restrict MPE access?
12. □ Is there `NOLIST` and `NOHELP` on data sensitive UDCs?
13. □ Are embedded passwords removed from all jobstreams?
14. □ Are system installation files removed?
15. □ Is there a procedure for positive identification from callers requesting access to the system?
16. □ Are there hard copy printouts of console messages?
17. □ Is the system console and tape drive restricted to operation personnel only?
18. □ Is the data center audited quarterly?
19. □ Are modem ports downed until required?
20. □ Are System Load Tape and System Backup Tapes protected?

B

Error Messages

General Error Messages

The first section of this appendix describes error messages returned by the CI (Command Interpreter) that relate to general security and account structure functions. Possible causes and suggestions for recovery are provided. The second section of this appendix describes ACD related error messages.

Table B-1. Error Messages

351	MESSAGE	`ACTION DISALLOWED SINCE NOT CREATOR OF FILE`
	CAUSE	You must be the creator of the file in order to to use the `:ALTSEC` command to change security restrictions.
	ACTION	For information only.
353	MESSAGE	`DISC I/O ERROR RELATED TO FILE LABEL ACCESS`
	CAUSE	An error was encountered by the input/output device when trying to get the file label.
	ACTION	Re-issue command. If error message occurs again, contact your System Manager.
410	MESSAGE	`ALTSEC REQUIRES AT LEAST A FILE NAME`
	CAUSE	You did not specify a file name. You must provide at least a file name in order to change any security restrictions.
	ACTION	Provide a file name.
411	MESSAGE	`EXTRANEOUS PARAMETER TO ALTSEC`
	CAUSE	The `:ALTSEC` command does not recognize one of the parameters that you specified on the command line.
	ACTION	Check the *MPE XL Commands Reference Manual* (3265-90003) for the valid `:ALTSEC` command parameters.

Table B-1. Error Messages (continued)

500	MESSAGE	`EXPECTED "(" TO START SECURITY SPECIFICATIONS`
	CAUSE	The left parenthesis was not included at the beginning of the security specifications.
	ACTION	Include the left parenthesis on the command line.
501	MESSAGE	`EXPECTED a ")" following the SECURITY SPECIFICATIONS`
	CAUSE	The right parenthesis was not included at the end of the security specifications.
	ACTION	Include the right parenthesis on the command line.
502	MESSAGE	`EXPECTED ONE OF R,A,W,L, or X FILE ACCESS MODES`
	CAUSE	You did not include a valid file access mode (`READ, APPEND, WRITE, LOCK,` or `EXECUTE`) on the command line.
	ACTION	Specify a valid file access mode.
503	MESSAGE	`EXPECTED ONE OF R,A,W,L,X, or S GROUP FILE ACCESS MODES`
	CAUSE	You did not include a valid group file access mode (`READ, APPEND, WRITE, LOCK,` or `EXECUTE`) on the command line.
	ACTION	Specify a valid group file access mode.
504	MESSAGE	`EXPECTED ONE OF R,A,W,L, or X ACCOUNT FILE ACCESS MODES`
	CAUSE	You did not include a valid account file access mode (`READ, APPEND, WRITE, LOCK,` or `EXECUTE,`) on the command line.
	ACTION	Specify a valid account file access mode.
505	MESSAGE	`IGNORED. SAVE ACCESS HAS NO MEANING AT FILE LEVEL`
	CAUSE	You cannot specify SAVE access at the file level.
	ACTION	This message is informational only.

Table B-1. Error Messages (continued)

506	MESSAGE	IGNORED. SAVE ACCESS NOT ALLOWED AT ACCOUNT LEVEL
	CAUSE	You cannot specify SAVE access at the account level.
	ACTION	This message is informational only.
507	MESSAGE	EXPECTED "Colon" SEPARATING MODE LIST FROM USER LIST
	CAUSE	You did not include a colon (:) between the mode list and the user list.
	ACTION	Include a colon (:) on the command line.
508	MESSAGE	EXPECTED ONE OF ANY AC, AL, GU, GL, OR CR USER TYPES
	CAUSE	You did not include an acceptable user type. Acceptable user types are Any, Account User (AC), Account Librarian (AL), Group User (GU), Group Librarian (GL), or Creator (CR).
	ACTION	Specify an acceptable user type.
509	MESSAGE	EXPECTED ONE OF ANY, AC, AL, GU, or GL USER TYPES
	CAUSE	You did not include an acceptable user type. Acceptable user types are for this command are Any, Account User (AC), Account Librarian (AL), Group User (GU), or Group Librarian (GL).
	ACTION	Specify an acceptable user type.
510	MESSAGE	EXPECTED EITHER ANY or AC USER TYPE
	CAUSE	You did not include an acceptable user types for this command. Acceptable user types are Any, or Account User (AC).
	ACTION	Specify an acceptable user type.
511	MESSAGE	USER TYPE CR NOT ALLOWED AT GROUP LEVEL
	CAUSE	The Creator (CR) user type is not allowed at the group level.
	ACTION	This message is informational only.

Table B-1. Error Messages (continued)

512	MESSAGE	`THIS USER TYPE NOT ALLOWED AT ACCOUNT LEVEL`
	CAUSE	You specified a user type that is not allowed at the account level.
	ACTION	This message is informational only.
513	MESSAGE	`READ ACCESS FOR THIS USER TYPE REDUNDANTLY SPECIFIED`
	CAUSE	You specified read access more than once on the same command line.
	ACTION	This message is informational only.
514	MESSAGE	`APPEND ACCESS FOR THIS USER TYPE REDUNDANTLY SPECIFIED`
	CAUSE	You specified append access more than once on the same command line.
	ACTION	This message is informational only.
515	MESSAGE	`WRITE ACCESS FOR THIS USER TYPE REDUNDANTLY SPECIFIED`
	CAUSE	You specified write access more than once on the same command line.
	ACTION	This message is informational only.
516	MESSAGE	`LOCK ACCESS FOR THIS USER TYPE REDUNDANTLY SPECIFIED`
	CAUSE	You specified lock access more than once on the same command line.
	ACTION	This message is informational only.
517	MESSAGE	`EXECUTE ACCESS FOR THIS USER TYPE REDUNDANTLY SPECIFIED`
	CAUSE	You specified execute access more than once on the same command line.
	ACTION	This message is informational only.

Table B-1. Error Messages (continued)

518	MESSAGE	`SAVE ACCESS FOR THIS USER TYPE REDUNDANTLY SPECIFIED`
	CAUSE	You specified save access more than once on the same command line.
	ACTION	This message is informational only.
519	MESSAGE	`THIS ACCESS MODE REDUNDANTLY SPECIFIED ON THIS ACCESS LIST`
	CAUSE	One of the access modes that you specified was repeated in the access list.
	ACTION	This message is informational only.
530	MESSAGE	`FIRST CHARACTER IN FILE NAME NOT ALPHABETIC`
	CAUSE	You specified something other than an alphabetic character at the beginning of the file name. You probably mistyped the file name.
	ACTION	Retype the command.
531	MESSAGE	`FILE NAME MISSING`
	CAUSE	You did not include a file name on the command line.
	ACTION	Specify a file name.
532	MESSAGE	`FILE NAME is more than eight CHARACTERS LONG`
	CAUSE	The file name that you specified is greater than eight characters, and file names can only be eight characters or fewer in length. You probably mistyped the file name.
	ACTION	Retype the command.
534	MESSAGE	`FILE NAME CONTAINS EMBEDDED NON-ALPHANUMERIC CHARACTERS`
	CAUSE	File names can contain both alphabetic and numeric characters. One of the characters in your file name is neither alphabetic nor numeric. You probably mistyped the file name.
	ACTION	Retype the command.

Table B-1. Error Messages (continued)

535	MESSAGE	`MISSING DELIMITER AFTER FILE NAME`
	CAUSE	You did not include a delimiter after the file name.
	ACTION	Include a delimiter (semi-colon, comma, period, or space), after the file name. See the *MPE XL Commands Reference Manual* (32650-90003) for the correct syntax.
540	MESSAGE	`FIRST CHARACTER IN GROUP NAME NOT ALPHABETIC`
	CAUSE	The first character of your group name is nonalphabetic. You probably mistyped the group name.
	ACTION	Retype the command.
541	MESSAGE	`GROUP NAME MISSING`
	CAUSE	You did not specify a group name on the command line.
	ACTION	Specify a group name on the command line.
542	MESSAGE	`GROUP NAME is more than eight CHARACTER LONG`
	CAUSE	Your group name is greater than eight characters, and group names can only be eight characters or fewer in length. You probably mistyped the group name.
	ACTION	Retype the command.
544	MESSAGE	`EMBEDDED NON-ALPHANUMERIC CHARACTER IN GROUP NAME.`
	CAUSE	Characters in group names can be both alphabetic and numeric. One of the characters in your group name is neither alphabetic nor numeric. You probably mistyped the group name.
	ACTION	Retype the command.
550	MESSAGE	`FIRST CHARACTER IN ACCOUNT NAME NOT ALPHABETIC`
	CAUSE	The first character of an account name must be alphabetic, and yours is not. You probably mistyped the account name.
	ACTION	Retype the command.

Table B-1. Error Messages (continued)

551	MESSAGE	`ACCOUNT NAME MISSING`
	CAUSE	You did not include an account name on the command line.
	ACTION	Specify an account name on the command line.
552	MESSAGE	`ACCOUNT NAME is more than eight CHARACTERS LONG`
	CAUSE	The account name that you specified is greater than eight characters. Account names can only be eight characters or fewer in length. You probably mistyped the account name.
	ACTION	Retype the command.
554	MESSAGE	`EMBEDDED NON-ALPHANUMERIC CHARACTER IN ACCOUNT NAME`
	CAUSE	Account names can consist of both alphabetic and numeric characters. One of the characters in the account name that you specified is neither alphabetic nor numeric. You probably mistyped the account name.
	ACTION	Retype the command.
590	MESSAGE	`FIRST CHARACTER IN USER NAME NOT ALPHABETIC`
	CAUSE	The first character of the user name that you specified is not alphabetic. You probably mistyped the user name.
	ACTION	Retype the command.
591	MESSAGE	`USER NAME IS MISSING`
	CAUSE	You did not include a user name on the command line.
	ACTION	Specify a user name.
592	MESSAGE	`USER NAME is more than eight CHARACTERS LONG`
	CAUSE	The user name that you specified is greater than eight characters. User names can only be eight characters or fewer in length. You probably mistyped the user name.
	ACTION	Retype the command.

Table B-1. Error Messages (continued)

594	MESSAGE	`EMBEDDED NON-ALPHANUMERIC CHARACTER IN USER NAME`
	CAUSE	User names can consist of both alphabetic and numeric characters. One of the characters in the user name that you specified is neither alphabetic nor numeric. You probably mistyped the user name.
	ACTION	Retype the command.
730	MESSAGE	`ALTACCT CAN HANDLE A MAXIMUM OF 71 PARAMETERS`
	CAUSE	You have specified too many parameters on the command line.
	ACTION	Consult the *MPE XL Commands Reference Manual* (32650-90003) for acceptable parameters.
731	MESSAGE	`ALTGROUP CAN HANDLE A MAXIMUM OF 71 PARAMETERS`
	CAUSE	You have specified too many parameters on the command line.
	ACTION	Consult the *MPE XL Commands Reference Manual* (32650-90003) for acceptable parameters.
732	MESSAGE	`ALTUSER CAN HANDLE A MAXIMUM OF 71 PARAMETERS`
	CAUSE	You have specified too many parameters on the command line.
	ACTION	Consult the *MPE XL Commands Reference Manual* (32650-90003) for acceptable parameters.
733	MESSAGE	`NEWACCT CAN HANDLE A MAXIMUM OF 71 PARAMETERS`
	CAUSE	You have specified too many parameters on the command line.
	ACTION	Consult the *MPE XL Commands Reference Manual* (32650-90003 for acceptable parameters.
734	MESSAGE	`NEWGROUP CAN HANDLE A MAXIMUM OF 71 PARAMETERS`
	CAUSE	You have specified too many parameters on the command line.
	ACTION	Consult the *MPE XL Commands Reference Manual* (32650-90003) for acceptable parameters.

Table B-1. Error Messages (continued)

735	MESSAGE	`NEWUSER CAN HANDLE A MAXIMUM OF 71 PARAMETERS`
	CAUSE	You have specified too many parameters on the command line.
	ACTION	Consult the *MPE XL Commands Reference Manual* (32650-90003) for acceptable parameters.
736	MESSAGE	`EXPECTED COMMA AFTER ACCOUNT NAME, BEFORE MANAGER'S NAME`
	CAUSE	You failed to include a comma between the account name and the manager's name.
	ACTION	Include a comma between the account name and the manager's name.
737	MESSAGE	`EXPECTED ONE OF THE FOLLOWING KEYWORDS: PASS, FILES, CPU, CONNECT, CAP, ACCESS, MAXPRI, LOCATER, VS, or HOMEVS`
	CAUSE	The command that you issued expected to see one of the parameters listed above. You specified a parameter that the command does not recognize.
	ACTION	Delete the parameter that is not specified in the list of accepted command parameters.
738	MESSAGE	`THE SYNTAX REQUIRES THAT AN EQUAL SIGN (=) FOLLOWS KEYWORD`
	CAUSE	You did not include an equal sign (=) following one of the keywords on the command line.
	ACTION	Find the keyword that is not followed by an equal sign (=) and enter one.
739	MESSAGE	`EXPECTED ONE OF: PASS, FILES, CPU, CONNECT, CAP, ACCESS, MAXPRI, LOCATER, ONVS, or HOMEVS`
	CAUSE	The command that you issued expected to see one of the parameters listed above. You specified a parameter that the command does not recognize.
	ACTION	Delete the parameter that is not specified in the list of accepted command parameters.

Table B-1. Error Messages (continued)

740	MESSAGE	`UNIDENTIFIABLE PARAMETER.  POSSIBLY A DELIMITER WAS OMITTED`
	CAUSE	The command that you issued does not recognize one of the parameters. It might be that you did not include a delimiter (semi-colon, comma, period, or space), between parameters.
	ACTION	Check the *MPE XL Commands Reference Manual* (32650-90003) and make sure that you did not omit a delimiter. If you did, enter it.
741	MESSAGE	`ACCESS INAPPROPRIATE FOR USER`
	CAUSE	One of the access modes that you specified on the command line is not allowed for users.
	ACTION	Check the allowable access modes in the *MPE XL Commands Reference Manual* (32650-90003) and change the command.
742	MESSAGE	`ACCESS REDUNDANTLY SPECIFIED.  LAST OCCURRENCE USED`
	CAUSE	One of the access modes that you specified on the command line was repeated. The last occurrence of the access mode is the one that will be used.
	ACTION	This message is informational only.
743	MESSAGE	`EXPECTED ONE OF AS, BS, CS, DS, OR ES`
	CAUSE	You did not specify an acceptable priority.
	ACTION	Specify an acceptable priority level.
744	MESSAGE	`MAXPRI REDUNDANTLY SPECIFIED. LAST OCCURRENCE USED`
	CAUSE	You specified the MAXPRI parameter twice on the same command line. The last MAXPRI value that was specified is the one implemented by the command.
	ACTION	This message is informational only.

Table B-1. Error Messages (continued)

745	MESSAGE	`MAXPRI INAPPROPRIATE FOR GROUPS. IGNORED`
	CAUSE	The MAXPRI parameter cannot be specified for groups. It was ignored.
	ACTION	This message is informational only.
746	MESSAGE	`CAPABILITY LIST REDUNDANTLY SPECIFIED. LAST OCCURRENCE USED`
	CAUSE	You specified the CAP parameter twice on the same command line. The last CAP list that was specified is the one implemented by the command.
	ACTION	This message is informational only.
747	MESSAGE	`NO CAPABILITY SPECIFIED. IGNORED`
	CAUSE	You did not specify any capabilities in your capability list.
	ACTION	This message is informational only.
748	MESSAGE	`EXPECTED ONE OF: SM, AM,  AL, GL, DI, OP, PH, DS,  MR, PM, IA, BA, CS, ND, SF, UB, CV, LG, NA, NM, or PS`
	CAUSE	You did not specify an acceptable capability.
	ACTION	See this manual for a definition of acceptable capabilities.
749	MESSAGE	`THIS CAPABILITY INAPPROPRIATE FOR GROUPS. IGNORED`
	CAUSE	One of the capabilities in your capability list cannot be specified for groups. It was ignored.
	ACTION	This message is informational only.
750	MESSAGE	`THIS CAPABILITY REDUNDANTLY SPECIFIED. IGNORED`
	CAUSE	You specified a capability twice in the capability list. There should be a caret pointing to the repeated capability.
	ACTION	This message is informational only.

Table B-1. Error Messages (continued)

751	MESSAGE	`CREATOR SPECIFIED NEITHER IA NOR BA FOR ACCOUNT, SO BOTH WERE IMPOSED`
	CAUSE	You did not specify either interactive access (IA) or batch access (BA) for the account. These must be specified.
	ACTION	This message is informational only.
752	MESSAGE	`CREATOR SPECIFIED NEITHER IA NOR BA FOR USER, SO BOTH WERE IMPOSED`
	CAUSE	You did not specify either interactive access (IA) or batch access (BA) for the user. These must be specified.
	ACTION	This message is informational only.
753	MESSAGE	`LOCAL ATTRIBUTE INAPPROPRIATE FOR GROUPS. IGNORED`
	CAUSE	The LOCAL attribute cannot be specified for groups. The attribute was ignored.
	ACTION	This message is informational only.
754	MESSAGE	`ACCOUNT MANAGER NAME MUST BE SPECIFIED IN :NEWACCT`
	CAUSE	You neglected to specify the name of the account manager. The :NEWACCT command requires the name of the account manager.
	ACTION	Specify the name of the account manager.
755	MESSAGE	`MANAGER NAME MUST START WITH ALPHABETIC CHARACTER`
	CAUSE	The first character of the manager name is not alphabetic. You probably mistyped the command.
	ACTION	Retype the command.
756	MESSAGE	`MANAGER NAME CANNOT BE MORE THAN 8 CHARACTERS LONG`
	CAUSE	The name of the manager is too long. Eight characters or fewer is the limit. You probably mistyped the command.
	ACTION	Retype the command.

Table B-1. Error Messages (continued)

758	MESSAGE	`EMBEDDED SPECIAL CHARACTER IN MANAGER'S NAME`
	CAUSE	The name of the manager can consist of both alphabetic and numeric characters. One of the characters in your manager name is neither alphabetic nor numeric. You probably mistyped the command.
	ACTION	Retype the command.
760	MESSAGE	`PASSWORD MUST START WITH ALPHABETIC CHARACTER`
	CAUSE	The password that you specified does not start with an alphabetic character. You probably mistyped the command.
	ACTION	Retype the command.
761	MESSAGE	`PASSWORD REDUNDANTLY SPECIFIED. LAST OCCURRENCE USED`
	CAUSE	You specified a password twice on the command line. The last occurrence of the password specification is the one implemented.
	ACTION	This message is informational only.
762	MESSAGE	`PASSWORD CANNOT BE MORE THAN 8 CHARACTERS LONG`
	CAUSE	You specified a password that has more than eight characters. A password can only be eight characters or fewer. You probably mistyped the command.
	ACTION	Retype the command.
764	MESSAGE	`EMBEDDED NON-ALPHANUMERIC CHARACTER IN PASSWORD`
	CAUSE	You specified a password with a character that is neither alphabetic nor numeric. You probably mistyped the command.
	ACTION	Retype the command.
765	MESSAGE	`HOME GROUP OPTION APPROPRIATE ONLY TO USERS. IGNORED`
	CAUSE	You specified the home group option for an account or a group. It may only be specified for users.
	ACTION	This message is informational only.

Table B-1. Error Messages (continued)

767	MESSAGE	`FILES OPTION INAPPROPRIATE FOR USERS.  IGNORED`
	CAUSE	You cannot specify the FILES option for a user.
	ACTION	This message is informational only.
768	MESSAGE	`EXPECTED POSITIVE INTEGER <2,147,483,647 AS SECTORS LIMIT`
	CAUSE	You specified a sectors limit with the FILES option that is greater than 2147483647.
	ACTION	Specfiy a new sectors limit that is less than 2147483647.
769	MESSAGE	`FILE SECTOR LIMIT MAY NOT BE A NEGATIVE NUMBER`
	CAUSE	You specified a negative number for the file sector limit. It must be a positive number.
	ACTION	Specfiy a new sectors limit with a positive number.
770	MESSAGE	`FILE SECTOR LIMIT REDUNDANTLY SPECIFIED.  LAST USED`
	CAUSE	You specified the file sector limit twice on the same command line. The last file sector limit specification is the one implemented.
	ACTION	This message is informational only.
771	MESSAGE	`VS OPTION INAPPROPRIATE FOR USERS. IGNORED`
	CAUSE	You cannot specify the ONVS option for a user. It was ignored.
	ACTION	This message is informational only.
773	MESSAGE	`CPU LIMIT OPTION INAPPROPRIATE FOR USERS. IGNORED`
	CAUSE	You cannot specify the CPU limit option for a user. It was ignored.
	ACTION	This message is informational only.

Table B-1. Error Messages (continued)

774	MESSAGE	`EXPECTED POSITIVE INTEGER <2,147,483,647 AS CPU SECONDS LIMIT`
	CAUSE	You specified a CPU limit that is greater than 2147483647.
	ACTION	Specfiy a new CPU limit that is less than 2147483647.
775	MESSAGE	`CPU SECONDS LIMIT MAY NOT BE A NEGATIVE NUMBER`
	CAUSE	You specified a negative number for the CPU seconds limit. Only a positive number is allowed.
	ACTION	This message is informational only.
776	MESSAGE	`CPU SECONDS LIMIT REDUNDANTLY SPECIFIED.  LAST USED`
	CAUSE	You specified a CPU seconds limit more than once on the same command line. The last CPU seconds limit specification is the one implemented.
	ACTION	This message is informational only.
779	MESSAGE	`CONNECT TIME OPTION INAPPROPRIATE FOR USERS. IGNORED`
	CAUSE	You cannot specify the connect time option for a user. It was ignored.
	ACTION	This message is informational only.
781	MESSAGE	`CONNECT TIME LIMIT MAY NOT BE A NEGATIVE NUMBER`
	CAUSE	You specified a negative number for the connect time limit option. Only a positive number is allowed.
	ACTION	Specify a new connect time limit that is a positive number.
782	MESSAGE	`CONNECT TIME LIMIT REDUNDANTLY SPECIFIED.  LAST USED`
	CAUSE	You specified a connect time limit more than once on the same command line. The last connect time limit specification is the one implemented.
	ACTION	This message is informational only.

Table B-1. Error Messages (continued)

784	MESSAGE	`"SM" CAPABILITY CANNOT BE REMOVED FROM MANAGER.SYS.  COMMAND REJECTED`
	CAUSE	You cannot remove System Manager (SM) capability from MANAGER.SYS.
	ACTION	Review account structure capabilities in this manual.
785	MESSAGE	`ATTEMPT TO REMOVE SM CAPABILITY FROM SYS ACCOUNT OVERRIDDEN`
	CAUSE	You cannot remove System Manager (SM) capability the SYS account.
	ACTION	Review account structure capabilities in this manual.
786	MESSAGE	`FILE SPACE LIMIT REQUESTED LESS THAN ACTUAL SPACE ALREADY IN USE. COMMAND REJECTED WITH NO CHANGES`
	CAUSE	You have requested a file space limit that is less than the space that is already in use.
	ACTION	This message is informational only.
787	MESSAGE	`GROUP CPU LIMIT REQUESTED EXCEEDS ACCOUNT LIMIT. GROUP LIMIT LOWERED TO ACCOUNT LIMIT`
	CAUSE	The group CPU limit cannot exceed the account CPU limit.
	ACTION	The group CPU limit that you specified has automatically been lowered to the account CPU limit.
788	MESSAGE	`GROUP CONNECT TIME LIMIT REQUESTED EXCEEDS ACCOUNT LIMIT. GROUP LIMIT LOWERED TO ACCOUNT LIMIT`
	CAUSE	The group connect time limit cannot exceed the account connect time limit.
	ACTION	The group connect time limit that you specified has automatically been lowered to the account connect time limit.
789	MESSAGE	`GROUP FILE SPACE LIMIT REQUESTED EXCEEDS ACCOUNT LIMIT. GROUP LIMIT LOWERED TO ACCOUNT LIMIT`
	CAUSE	You have requested a group file space limit that exceeds the account file space limit.
	ACTION	The group file space limit has automatically been lowered to the account file space limit.

Table B-1. Error Messages (continued)

790	MESSAGE	GROUP CAPABILITIES REQUESTED EXCEED ACCOUNT CAPABILITIES! "NOT" GRANTED
	CAUSE	The group capabilities cannot exceed the account capabilities.
	ACTION	This message is informational only.
791	MESSAGE	GROUP FILE SPACE LIMIT REQUESTED LESS THAN ACTUAL SPACE ALREADY IN USE. COMMAND REJECTED
	CAUSE	You have requested a group file space limit that is less than the space that is already in use.
	ACTION	This message is informational only.
792	MESSAGE	ACCOUNT MANAGER ATTEMPTED TO REMOVE HIS OWN ACCOUNT MANAGER CAPABILITY. COMMAND REJECTED
	CAUSE	You cannot remove account manager capability from the account manager account.
	ACTION	This message is informational only.
793	MESSAGE	USER MAXPRI REQUESTED IS GREATER THAN THE ACCOUNT MAXPRI. USER MAXPRI LOWERED TO ACCOUNT'S
	CAUSE	The group maximum priority level cannot exceed the account maximum priority level.
	ACTION	The group connect maximum priority level that you specified has automatically been lowered to the account maximum priority level.
794	MESSAGE	USER CAPABILITIES REQUESTED EXCEED ACCOUNT CAPABILITIES. "NOT" GRANTED
	CAUSE	User capabilities cannot exceed account capabilities.
	ACTION	This message is informational only.
795	MESSAGE	USER ASSIGNED LOCAL ATTRIBUTES GREATER THAN THE ACCOUNT LOCAL ATTRIBUTES. LOWERED TO ACCOUNT'S
	CAUSE	User local attributes cannot be greater than the account's local attributes.
	ACTION	The user local attributes were automatically lowered to the account's local attributes.

Table B-1. Error Messages (continued)

796	MESSAGE	HOME GROUP REDUNDANTLY SPECIFIED. LAST OCCURRENCE USED.
	CAUSE	You specified the home group more than once on the command line. The last home group specification is the one implemented.
	ACTION	This message is informational only.
797	MESSAGE	LOCAL ATTRIBUTE REDUNDANTLY SPECIFIED. LAST OCCURRENCE USED
	CAUSE	You specified the local attribute more than once on the command line. The last local attribute specification is the one implemented.
	ACTION	This message is informational only.
798	MESSAGE	EXPECTED INTEGER BETWEEN -2,147,483,647 AND 2,147,483,647
	CAUSE	You specified an integer that is not greater than -2147483647 or less than 2147483647.
	ACTION	Specfiy an integer within the accepted range.
799	MESSAGE	EXPECTED ONE OF PH, DS, MR, PM, IA or BA
	CAUSE	The command that you issued expected one of the following capabilities: Process Handling (PH), Extra Data Segments (DS), Multiple RIN (MR), Privileged Mode (PM), Interactive Access (IA), Batch Access (BA).
	ACTION	Review the account structure capabilities in this manual, and re-issue the command.
956	MESSAGE	THIS COMMAND REQUIRES SYSTEM MANAGER (SM) CAPABILITY
	CAUSE	You must have System Manager (SM) capability to execute this command.
	ACTION	See the System Manager.
957	MESSAGE	THIS COMMAND REQUIRES ACCOUNT MANAGER (AM) CAPABILITY
	CAUSE	You must have Account Manager (AM) capability to execute this command.
	ACTION	See the Account Manager.

ACD Related Error Messages

This appendix lists error messages which may be encountered when creating or modifying ACDs.

7100

MESSAGE `UNABLE TO DEALLOCATE ACD SPACE. (CIWARN 7100)`

CAUSE ACD information is kept as an MPE "pseudo extent". A pointer to this "pseudo extent" is maintained for each file or device which has an ACD. If you are attempting to delete an ACD, the pseudo extent will be deallocated by MPE. Even if the operation fails and you get this warning, the ACD will still be deleted.

If you are attempting to add additional entries to an existing ACD, then it may be necessary to create a larger ACD (and therefore allocate a larger pseudo extent). After the new ACD is created, MPE will deallocate the old pseudo extent automatically. You may get the warning if the deallocation of the old pseudo extent fails. The new ACD entries succeed regardless, and an ACD with all of the desired entries will be associated with the device or file.

ACTION No immediate action need be taken. You may wish to report the occurrence to your System Administrator so the lost disc space can be recovered at the next system re-start.

This is only a warning, the operation you performed succeeded!

7101

MESSAGE `ACD VERSION DOES NOT MATCH THE CURRENT VERSION. (CIWARN 7101)`

CAUSE There is a version number associated with the MPE software which implements ACDs. This version number is placed in the ACD itself when an ACD is created. Each time an ACD is accessed the version number in the ACD is checked against the current version number for the software running on your system.

If you are attempting to delete an ACD and these numbers do not match, then MPE will issue this warning message. Note that the version numbers here are not the same as the version update fix (V.UU.FF) numbers associated with MPE. Instead they are internal version numbers associated only with the ACD component of MPE.

ACTION You do not need to take any additional action to correct this problem. The ACD will be deleted successfully. You can create a new ACD, if you wish, without any further side effects.

7102 MESSAGE ACD WAS CORRUPTED PRIOR TO BEING DELETED. (CIWARN 7102)

CAUSE This message indicates that the ACD you deleted was corrupted. The delete operation succeeded so there is no ACD associated with the device or file in question.

ACTION No action needs to be taken. The delete operation has removed the corrupted ACD. You can create a new ACD, if you wish, without any further side effects.

7103 MESSAGE OPERATION FAILED ON SOME DEVICES SPECIFIED. (CIWARN 7103)

CAUSE The operation which you requested (;NEWACD, :DELACD, ;REPPAIR, ;DELPAIR, ;ADDPAIR, or ;COPYACD) did not succeed for all of the devices in the the device specification. If a device class was specified, the operation failed for one or more devices in the device class. If "@" was specified, indicating all devices on the system, then the operation failed on one or more devices.

ACTION Use the :SHOWDEV command with the ;ACD option to determine which devices the command failed on. Then execute the same :ALTSEC command against those devices one at a time to determine the reason for the failure.

7104 MESSAGE MISSING CLOSE PARENTHESIS ")" IN ACD INDIRECT FILE. (CIWARN 7104)

CAUSE An opening parenthesis was found in the ACD indirect file, however, the corresponding closing parenthesis was not found. This message indicates that the ACD indirect file was syntactically correct except for the missing closing parenthesis.

ACTION To avoid this message, add the closing parenthesis to your ACD indirect file. Alternatively, you could delete the opening parenthesis which is already in your ACD indirect file since it is not required.

7105 MESSAGE EXTRA CLOSE PARENTHESIS ")" ENCOUNTERED IN ACD INDIRECT FILE. (CIWARN 7105)

CAUSE A closing parenthesis was found in the ACD indirect file. However, the corresponding opening parenthesis was not found. This message indicates that the ACD indirect file was syntactically correct except for the extra closing parenthesis.

ACTION To avoid this message, add an opening parenthesis to your ACD indirect file. Alternatively, you could delete the closing parenthesis which is already in your ACD indirect file since it is not required.

7221 MESSAGE `WILDCARDS NOT ALLOWED IN FILENAME HERE. (CIERR 7221)`

CAUSE You have specified a generic file name which contains wildcards as the target file name or the source file name in the `:ALTSEC` command.

ACTION Repeat the `:ALTSEC` command for each file contained in the file set specified by the wildcard.

7223 MESSAGE `LOCKWORDS NOT ALLOWED IN GENERIC FILE SETS. (CIERR 7223)`

CAUSE A generic file specification (one which contains wildcards) should not contain a lockword.

ACTION Remove the lockword from the generic file specification.

7224 MESSAGE `LOCKWORDS NOT ALLOWED. (CIERR 7224)`

CAUSE A lockword was specified as part of a file name.

ACTION Remove the lockword from the file name.

7225 MESSAGE `INVALID CHARACTER IN DEVICE CLASS NAME. (CIERR 7225)`

CAUSE An invalid character was included in a device class name. Device class names must begin with a letter and they can contain letters or numbers after the first character. The maximum length for a device class name is 8 characters.

ACTION Correct the device class name and issue the command again.

7227 MESSAGE `NUMBER SPECIFIED IS GREATER THAN 32767. (CIERR 7227)`

CAUSE You have specified an ASCII representation of a number which is larger than 32767. 32767 is the largest number which can be stored in a 16-bit signed integer. This number is too large to be valid in this context.

ACTION Re-issue the command using a number which is valid. Notice that the valid range for the number depends on the context in which you are using it. An *ldev* number, for example, must be less than 999 on MPE/iX.

7228 MESSAGE `WILDCARD CHARACTERS, OTHER THAN "@" BY ITSELF, NOT ALLOWED IN DEVICE CLASS NAME. (CIERR 7228)`

CAUSE You have specified a device class name which contains wildcard characters. The use of wildcard characters is not supported for device class names.

ACTION Please remove any wildcards included in the device class name specified.

7229 MESSAGE `"_" (UNDERBAR) CHARACTER NOT ALLOWED IN DEVICE CLASS NAME. (CIERR 7229)`

CAUSE The “_” (underbar) character was included in a device class name. Device class names must begin with a letter and they can contain letters or numbers after the first character. The maximum length for a device class name is 8 characters.

ACTION Remove the “_” (underbar) character from the device class name and re-issue the command.

7230 MESSAGE `SINGLE QUOTE "’" CHARACTER NOT ALLOWED IN DEVICE CLASS NAME. (CIERR 7230)`

CAUSE A single quote (’) character was included in a device class name. Device class names must begin with a letter and they can contain letters or numbers after the first character. The maximum length for a device class name is 8 characters.

ACTION Remove the single quote (’) character from the device class name and re-issue the command.

7231 MESSAGE `FULLY QUALIFIED NAME NOT ALLOWED HERE. (CIERR 7231)`

CAUSE A fully qualified name is not allowed in this context. This error could apply to either file names or user names.

ACTION Please issue the command without specifying the fully qualified file or user name. If it is a file name, omit the group and account. If it is a user name, omit the account.

7250 MESSAGE `INVALID USER SPECIFICATION. (CIERR 7250)`

CAUSE You must specify a standard MPE user specification. This specification must take one of the following forms:

username.acctname
@.acctname
@.@

You must use “fully qualified” user specifications (for example, you cannot put the *username* by itself and default *acctname* to the logon account).

ACTION Correct the user specification to conform to the rules specified above.

7251 MESSAGE `DUPLICATE ACCESS MODE SPECIFIED. (CIERR 7251)`

CAUSE Your ACD specification contains a duplicated access mode in the list of access modes specified for a single ACD entry.

Examples:

```
:ALTSEC FILENAME;NEWACD=( R,W,R: FRED.SMITH )
```

The `:ALTSEC` command shown above is illegal because read access is specified twice for a single ACD entry (corresponding to user `FRED.SMITH`).

```
:ALTSEC FILENAME;NEWACD=( R,W: JOE.SMITH; R,X: BILL.SMITH )
```

In the `:ALTSEC` command above, however, it is not illegal to specify read access twice because it is for two different ACD entries (corresponding to `JOE.SMITH` and `BILL.SMITH`).

ACTION Delete the duplicate access mode from your list and issue the `:ALTSEC` command again.

7252 MESSAGE `DUPLICATE PERMISSION SPECIFIED. (CIERR 7252)`

CAUSE Your ACD specification contains a duplicated permission in the list of access modes specified for a single ACD entry.

Examples:

```
:ALTSEC FILENAME;NEWACD=( R,W,RACD,X,RACD: FRED.SMITH )
```

The `:ALTSEC` command shown above is illegal because read ACD permission is specified twice for a single ACD entry (corresponding to user `FRED.SMITH`).

```
:ALTSEC FILENAME;NEWACD=( R,W,RACD: JOE.SMITH; R,X,RACD: BILL.SMITH
)
```

In the `:ALTSEC` command above, however, it is not illegal to specify read ACD permission twice because it is for two different ACD entries (corresponding to `JOE.SMITH` and `BILL.SMITH`).

ACTION Delete the duplicate permission from your list and issue the `:ALTSEC` command again.

7253 MESSAGE `CONTRADICTORY ACCESS MODES SPECIFIED. (CIERR 7253)`

CAUSE You have specified access modes for a given entry which are contradictory. The examples below will clarify what is meant by contradictory access modes.

Examples:

```
:ALTSEC FILENAME;NEWACD=( R,W,NONE: @.@ )
```

The `:ALTSEC` command shown above is illegal because you are granting read and write access to the same user (`@.@`) you are granting no access.

```
:ALTSEC FILENAME;NEWACD=( R,W: @.@; NONE: BILL.SMITH )
```

In the `:ALTSEC` command above, however, it is not illegal because you are granting read and write access to a different user than the one to whom you are granting no access.

ACTION Change your access modes so that the modes specified for all your entries are not contradictory.

7254 MESSAGE `INVALID ACCESS MODE SPECIFIED. (CIERR 7254)`

CAUSE You have specified an invalid access mode. Only the following access modes are legal in an ACD specification:

Mode	Meaning
R	Read access allowed
W	Write access allowed
X	eXecute access allowed
L	Lock access allowed
A	Append access allowed
NONE	No access allowed
RACD	Read ACD permission

Upper or lower case is allowed. You may specify each mode only once for a given ACD entry. If NONE is specified then you may not specify any other access mode or permission for the same entry.

ACTION Correct your ACD specification to include only valid access modes.

7255 MESSAGE `MISSING OPEN PARENTHESIS "(". (CIERR 7255)`

CAUSE You have omitted the open parenthesis "(" from your ACD specification. Unless you are using an ACD indirect file, both the open and close parentheses are required.

ACTION Re-issue the command and add the missing open parenthesis.

7256 MESSAGE `MISSING CLOSE PARENTHESIS ")". (CIERR 7256)`

CAUSE You have omitted the close parenthesis ")" from your ACD specification. Unless you are using an ACD indirect file both the open and close parentheses are required.

ACTION Re-issue the command and add the missing close parenthesis.

7257 MESSAGE `MISSING COLON ":". (CIERR 7257)`

CAUSE You have omitted the colon character from your ACD specification. A colon is required after the access modes and before the user specification.

ACTION Re-issue the command and add the missing colon.

7258 MESSAGE `UNEXPECTED INPUT ENCOUNTERED AFTER ACD SPECIFICATION. (CIERR 7258)`

CAUSE At the end of your ACD specification, after the last user specification or the closing parenthesis, you have some additional input which is not recognized as be correct.

ACTION Delete the extra input and re-issue the command.

7259 MESSAGE `INVALID ACCOUNT NAME SPECIFIED. (CIERR 7259)`

CAUSE The account name you have specified is invalid for your system.

Check the account name and re-issue the command specifying the correct account name.

7260 MESSAGE `EMBEDDED "@" CHARACTER NOT ALLOWED IN USER SPECIFICATION. (CIERR 7260)`

CAUSE You must specify a standard MPE user specification. This specification must take one of the following forms:

username.acctname
@.acctname
@.@

You must use "fully qualified" user specifications (for example, you cannot put the *username* by itself and default *acctname* to the logon account).

ACTION Correct the user specification to conform to the rules specified above.

7261 MESSAGE `USER NAME MUST BE "@" IF ACCOUNT NAME IS SPECIFIED AS "@". (CIERR 7261)`

CAUSE You must specify a standard MPE user specification. This specification must take one of the following forms:

username.acctname
@.acctname
@.@

You must use "fully qualified" user specifications (for example, you cannot put the *username* by itself and default *acctname* to the logon account).

ACTION Correct the user specification to conform to the rules specified above.

7262 MESSAGE `"#" CHARACTER NOT ALLOWED IN USER SPECIFICATION. (CIERR 7262)`

CAUSE You must specify a standard MPE user specification. This specification must take one of the following forms:

username.acctname
@.acctname
@.@

You must use "fully qualified" user specifications (for example, you cannot put the *username* by itself and default *acctname* to the logon account).

ACTION Correct the user specification to conform to the rules specified above.

7263 MESSAGE `"?" CHARACTER NOT ALLOWED IN USER SPECIFICATION. (CIERR 7263)`

CAUSE You must specify a standard MPE user specification. This specification must take one of the following forms:

username.acctname
@.acctname
@.@

You must use "fully qualified" user specifications (for example, you cannot put the *username* by itself and default *acctname* to the logon account).

ACTION Correct the user specification to conform to the rules specified above.

7264 MESSAGE `MISSING ACCESS MODE IN ACD SPECIFICATION. (CIERR 7264)`

CAUSE You have either omitted an access mode in your ACD specification or you have typed an extra comma (,) in your specification.

ACTION Either delete the extra comma or provide the missing access mode when you re-issue the command.

7265 MESSAGE `USER SPECIFICATION MUST BE FULLY QUALIFIED. (CIERR 7265)`

CAUSE You must specify a standard MPE user specification. This specification must take one of the following forms:

username.acctname
@.acctname
@.@

You must use "fully qualified" user specifications (eg: you cannot put the *username* by itself and default *acctname* to the logon account).

ACTION Correct the user specification to conform to the rules specified above.

7266 MESSAGE `INVALID USER NAME SPECIFIED. (CIERR 7266)`

CAUSE The user name part of your user specification is invalid for your system. The account name is valid.

ACTION Check the user name and re-issue the command specifying the correct user name.

7267 MESSAGE `MISSING USER SPECIFICATION. (CIERR 7267)`

CAUSE You have either omitted a user specification or you have included and extra comma (,) in your ACD specification.

ACTION Either delete the extra comma or add the missing user specification to the ACD specification when you re-issue the command.

7268 MESSAGE `DUPLICATE USER SPECIFICATION ENCOUNTERED IN LIST. (CIERR 7268)`

CAUSE The ACD specification you used contains more than one reference to the same user specification.

ACTION Delete the duplicate reference from you ACD specification and re-issue the command.

7269 MESSAGE `INTERNAL ERROR NUMBER "-269". (CIERR 7269)`

CAUSE An unexpected internal error has occurred.

ACTION Try re-issuing the command. If you still get this error, contact your HP Representative and give him/her the internal error number.

7270 MESSAGE `INTERNAL ERROR NUMBER "-270". (CIERR 7270)`

CAUSE An unexpected internal error has occurred.

ACTION Try re-issuing the command. If you still get this error, contact your HP Representative and give him/her the internal error number.

7271 MESSAGE `INTERNAL ERROR NUMBER "-271". (CIERR 7271)`

CAUSE An unexpected internal error has occurred.

ACTION Try re-issuing the command. If you still get this error, contact your HP Representative and give him/her the internal error number.

7272 MESSAGE `INVALID LDEV NUMBER SPECIFIED. (CIERR 7272)`

CAUSE You have specified an *ldev* number which does not correspond to an *ldev* which is currently configured on your system.

ACTION Correct the *ldev* number and re-issue the command.

7273 MESSAGE `INVALID TARGET LDEV NUMBER SPECIFIED. (CIERR 7273)`

CAUSE You have specified an *ldev* number which does not correspond to an *ldev* which is currently configured on your system.

ACTION Correct the *ldev* number and re-issue the command.

7274 MESSAGE `INVALID SOURCE LDEV NUMBER SPECIFIED. (CIERR 7274)`

CAUSE You have specified an *ldev* number which does not correspond to an *ldev* which is currently configured on your system.

ACTION Correct the *ldev* number and re-issue the command.

7275 MESSAGE `INVALID DEVICE CLASS NAME SPECIFIED. (CIERR 7275)`

CAUSE You have specified a device class name which does not correspond to any device class currently configured on your system.

ACTION Correct the device class name and re-issue the command.

7300 MESSAGE `ACD ENTRY DOES NOT EXIST. (CIERR 7300)`

CAUSE You are attempting to access (delete or replace) an ACD entry which does not exist in the specified ACD.

ACTION You can list the content of an ACD using the `:LISTF ,-2` command (for file ACDs) or the `:SHOWDEV` command with the `;ACD` option (for device ACDs).

7301 MESSAGE `THERE IS NO ACD ASSOCIATED WITH THE SOURCE FILE. (CIERR 7301)`

CAUSE You are attempting to copy an ACD from a file which does not currently have an ACD associated with it.

ACTION Copy the ACD from a file which actually has an ACD associated with it.

7302 MESSAGE `THE ACD ASSOCIATED WITH THE SOURCE FILE IS CORRUPTED. (CIERR 7302)`

CAUSE You are attempting to copy a file ACD which is corrupted.

ACTION You cannot copy this ACD because it is corrupted. It is possible to delete the ACD using the `;DELACD` option on the `:ALTSEC` command. This will leave your file without an ACD to protect it. You can also create an ACD for that file (using the `;NEWACD` option), or you can copy an existing ACD from another file (using the `;COPYACD` option), without deleting the current ACD first. This is only allowed for corrupted ACDs (otherwise the file must not have an ACD prior to using the `;NEWACD` or `;COPYACD` options).

7303 MESSAGE `THERE IS ALREADY AN ACD ASSOCIATED WITH THE TARGET FILE. (CIERR 7303)`

CAUSE You are attempting to create a new ACD for (via the `;NEWACD` option), or copy an existing ACD to (via the `;COPYACD` option) a file which already has an ACD associated with it.

ACTION You must either delete the existing target file ACD prior to executing the `:ALTSEC` command with the `;NEWACD` or `;COPYACD` option, or you must use the `;ADDPAIR` and `;REPPAIR` options to change the existing ACD.

7304 MESSAGE `THE ACD ASSOCIATED WITH THE TARGET FILE IS CORRUPTED. (CIERR 7304)`

CAUSE You are attempting to copy a file ACD which is corrupted.

ACTION You cannot copy this ACD because it is corrupted. It is possible to delete the ACD using the `;DELACD` option on the `:ALTSEC` command. This will leave your file without an ACD to protect it. You can also create an ACD for that file (using the `;NEWACD` option), or you can copy an existing ACD from another file (using the `;COPYACD` option), without deleting the current ACD first. This is only allowed for corrupted ACDs (otherwise the file must not have an ACD prior to using the `;NEWACD` or `;COPYACD` options).

7305 MESSAGE `THERE IS NO ACD ASSOCIATED WITH TARGET FILE. (CIERR 7405)`

CAUSE You are attempting to manipulate an ACD for a file which does not have an ACD.

ACTION You must create the ACD (via the `;NEWACD` option on the `:ALTSEC` command) before you can manipulate it. You can determine if a file has an ACD by using the `:LISTF ,-2` command.

7306 MESSAGE `THERE IS NO ACD ASSOCIATED WITH THE SOURCE LDEV. (CIERR 7306)`

CAUSE You are attempting to copy an ACD from a device which does not currently have an ACD associated with it.

ACTION Copy the ACD from a device which actually has an ACD associated with it.

7307 MESSAGE `THE ACD ASSOCIATED WITH THE SOURCE LDEV IS CORRUPTED. (CIERR 7307)`

CAUSE You are attempting to copy a device ACD which is corrupted.

ACTION You cannot copy this ACD because it is corrupted. It is possible to delete the ACD using the `;DELACD` option on the `:ALTSEC` command. This will leave your device without an ACD to protect it. You can also create an ACD for that device (using the ;NEWACD option), or you can copy an existing ACD from another device (using the `;COPYACD` option), without deleting the current ACD first. This is only allowed for corrupted ACDs (otherwise the device must not have an ACD prior to using the `;NEWACD` or `;COPYACD` options).

7308 MESSAGE THERE IS ALREADY AN ACD ASSOCIATED WITH THE TARGET LDEV. (CIERR 7308)

CAUSE You are attempting to create a new ACD for (via the ;NEWACD option), or copy an existing ACD to (via the ;COPYACD option) a device which already has an ACD associated with it.

ACTION You must either delete the existing ACD prior to executing the :ALTSEC command with the ;NEWACD or ;COPYACD option, or you must use the ;ADDPAIR and ;REPPAIR options to change the existing ACD.

7309 MESSAGE THE ACD ASSOCIATED WITH THE TARGET LDEV IS CORRUPTED. (CIERR 7309)

CAUSE You are attempting to manipulate a device ACD which is corrupted.

ACTION You cannot manipulate this ACD because it is corrupted. It is possible to delete the ACD using the ;DELACD option on the :ALTSEC command. This will leave your device without an ACD to protect it. You can also create an ACD for that device (using the ;NEWACD option), or you can copy an existing ACD from another device (using the ;COPYACD option), without deleting the current ACD first. This is only allowed for corrupted ACDs (otherwise the device must not have an ACD prior to using the ;NEWACD or ;COPYACD options).

7310 MESSAGE THERE IS NO ACD ASSOCIATED WITH TARGET LDEV. (CIERR 7310)

CAUSE You are attempting to manipulate an ACD for a device which does not have an ACD.

ACTION You must create the ACD (via the ;NEWACD option on the :ALTSEC command) before you can manipulate it. You can determine which devices have ACDs using the :SHOWDEV command with the ;ACD option.

7311 MESSAGE ERROR OPENING ACD INDIRECT FILE. (CIERR 7311)

CAUSE An error occurred when opening the ACD indirect file. An additional message will be printed indicating the exact cause of the error.

ACTION Take the appropriate action to correct/avoid the error. The additional message should help you figure out what action to take.

7312 MESSAGE `INVALID ACD INDIRECT FILE CODE. FILE CODE MUST BE 0. (CIERR 7312)`

CAUSE You have specified an ACD indirect file with a non-zero file code. This should not be a problem very often because most editors create text files with a file code of zero.

ACTION You can determine if the file code for a file is zero by using the `:LISTF` command. You can use `:FCOPY` to copy the file to another file which has a file code of zero.

7313 MESSAGE `INVALID ACD INDIRECT FILE RECORD SIZE. MUST BE <= 88 BYTES. (CIERR 7313)`

CAUSE You have specified an ACD indirect file with a record length greater than 88 bytes. This should not be a problem very often because most editors create text files with record lengths less than or equal to 88 bytes. The record length is often affected by whether or not you choose to use numbered or unnumbered files. Either file type is acceptable if the total record length is less than or equal to 88 bytes.

ACTION You can determine the record length of a file by using the `:LISTF` command. You can use `:FCOPY` to copy the file to another file with an appropriate record length. Be careful not to truncate important data when copying the file.

7314 MESSAGE `ACD INDIRECT FILE MUST BE ASCII. (CIERR 7314)`

CAUSE You have specified an ACD indirect file which is not an ASCII file. This should not be a problem very often because most editors create ASCII text files.

ACTION You can determine if the file is an ASCII file by using the `:LISTF` command. You can use `:FCOPY` to copy the file to another file which is an ASCII file.

7315 MESSAGE `INVALID ACD INDIRECT FILE RECORD FORMAT. MUST BE FIXED. (CIERR 7315)`

CAUSE You have specified an ACD indirect file which does not have fixed length records. This should not be a problem very often because most editors create text files with fixed length records, or they offer some option to allow the user to select the record format.

ACTION You can determine if the file has fixed length records by using the `:LISTF` command. You can use `:FCOPY` to copy the file to another file with fixed length records to avoid this problem.

7316 MESSAGE MAXIMUM NUMBER OF ACD ENTRIES (40) WOULD BE EXCEEDED. (CIERR 7316)

CAUSE You are attempting to add some number of entries to the ACD. If you added these entries to the ACD then the total number of entries in the ACD would exceed the maximum number allowed (40).

ACTION You cannot have more than 40 entries in a given ACD. You may be able to combine some of the entries by using wildcards. For example, you could have one entry for all the FINANCE users instead of having separate entries for JOHN.FINANCE, SAM.FINANCE, TOM.FINANCE, for example. This will only work if the users are supposed to have the same access rights.

7317 MESSAGE ATTEMPTING TO MODIFY MORE ENTRIES THAN CURRENTLY EXIST IN ACD. (CIERR 7317)

CAUSE You are attempting to modify (with the :ALTSEC ;REPPAIR or ;DELPAIR option) more entries than currently exist in the ACD.

ACTION You can use either :LISTF -2 or :SHOWDEV to determine what the ACD currently looks like. Issue the :ALTSEC command again (with the appropriate ;REPPAIR or ;DELPAIR option) making sure that you are modifying only entries which actually exist in the ACD.

7318 MESSAGE ENTRY ALREADY EXISTS IN ACD. (CIERR 7318)

CAUSE You are attempting to add an entry to an ACD which already contains an entry corresponding to the same user. This error will only occur if the user name matches exactly a user name already specified in the ACD. For example, if you are attempting to add an entry for JOHN.DOE and an entry already exists for @.DOE this will not result in an error. If, however, you attempt to add an entry for @.DOE you will get this error.

ACTION You can modify an existing entry in an ACD by using the ;REPPAIR option on the :ALTSEC command. Or you can delete the entry using the ;DELPAIR option and re-issue the :ALTSEC command with the ;ADDPAIR option.

7319 MESSAGE `INCOMPATIBLE TARGET AND SOURCE FOR COPYING ACD. (CIERR 7319)`

CAUSE The target and source file/device specified on the `:ALTSEC` command must be of the same type. Either they must both be devices, or they must both be files.

ACTION If you want to grant the same explicit access rights to a file and a devices you should create an indirect file containing the ACD specification and use this indirect file on the `:ALTSEC` command with the `;NEWACD` option.

7320 MESSAGE `SOURCE AND TARGET FOR COPYING ACD ARE THE SAME. (CIERR 7320)`

CAUSE The source and target specified on the `:ALTSEC` command are the same. Either they are the same device, or they are the same file. You cannot copy an ACD onto itself.

ACTION Either the target or the source must be changed for this command to execute correctly.

7321 MESSAGE `USER DOES NOT HAVE SUFFICIENT CAPABILITIES TO MANIPULATE ACD. (CIERR 7321)`

CAUSE The user attempting to manipulate the ACD does not have sufficient capabilities, or is not the creator of the file.

The capability requirements for manipulating an ACD are as follows:

a user with SM capability can manipulate any ACD;

a user with AM capability can manipulate any ACD associated with a file in the account for which he/she has AM capability;

only a user with SM capability can manipulate device ACDs.

The creator of the file is not required to have any specific capabilities to manipulate the ACD.

Notice, however, that SM or AM capability always takes precedence over the permissions granted explicitly within the ACD. Even if you have specified that `MANAGER.SYS` has no access, he or she can still access the ACD.

ACTION The person attempting to manipulate the ACD must request the appropriate capability from either system or account manager. Alternatively, the user can ask the file creator to make the desired change to the ACD.

7322 MESSAGE `OPERATION FAILED ON ALL DEVICES SPECIFIED. (CIERR 7322)`

CAUSE The operation which you requested (`;NEWACD, :DELACD, ;REPPAIR, ;DELPAIR, ;ADDPAIR`, or `;COPYACD`) did not succeed for any of the devices in the the device specification. If a device class was specified, the operation failed for all of the devices in the device class. If "@" was specified, indicating all devices on the system, then the operation failed on all devices on the system.

ACTION Execute the same `:ALTSEC` command against those devices one at a time to determine the reason for the failure.

7323 MESSAGE `USER NOT ALLOWED TO READ THE ACD. (CIERR 7323)`

CAUSE The user attempting to read the ACD does not have sufficient capabilities, is not the creator of the file, or has not been granted explicit "read ACD" (`RACD`) permission.

The capability requirements for reading an ACD are as follows:

a user with SM capability can read any ACD;

a user with AM capability can read any ACD associated with a file in the account for which he/she has AM capability;

the creator of the file can read the ACD.

Users granted "read ACD" (`RACD`) permission can read an ACD regardless of their capabilities. Note that SM or AM capability always takes precedence over the permissions granted explicitly within the ACD. Even if you specify that `MANAGER.SYS` has no access, he or she can still do so.

ACTION The person attempting to read the ACD must request the appropriate permission/capability from either the file creator or a system or account manager.

7324 MESSAGE `USER NOT ALLOWED TO COPY THE SOURCE ACD. (CIERR 7324)`

CAUSE The user attempting to copy the ACD does not have sufficient capabilities, is not the creator of the file, or has not been granted explicit "read ACD" (`RACD`) permission.

The capability requirements for copying an ACD are as follows:

a user with SM capability can copy any ACD;

a user with AM capability can copy any ACD associated with a file in the account for which he/she has AM capability;

the creator of the file can copy the ACD.

Users granted "read ACD" (`RACD`) permission can copy an ACD regardless of their capabilities. Note that SM or AM capability always takes precedence over the permissions granted explicitly within the ACD. Even if you specify that `MANAGER.SYS` has no access, he or she can still do so.

ACTION The person attempting to copy the ACD must request the appropriate permission/capability from either the file creator or a system or account manager.

7325 MESSAGE `ERROR OPENING TARGET FILE. (CIERR 7325)`

CAUSE An error occurred when opening the target file. An additional message will be printed indicating the exact cause of the error.

Take the appropriate action to correct/avoid the error. The additional message should help you figure out what action to take.

7326 MESSAGE `ERROR OPENING SOURCE FILE. (CIERR 7326)`

CAUSE An error occurred when opening the source file. An additional message will be printed indicating the exact cause of the error.

ACTION Take the appropriate action to correct/avoid the error. The additional message should help you figure out what action to take.

7400 MESSAGE `ACD INTERNAL ERROR. (CIERR 7400)`

CAUSE This message indicated that some kind of internal error occurred while processing your command. This message will be preceded by another message indicating the internal status and subsystem number. This information will be helpful in diagnosing the cause of the problem.

ACTION Contact you HP Support Representative.

7401 MESSAGE `ERROR ENCOUNTERED WITHIN ACD INDIRECT FILE.`

CAUSE A error occurred when performing an :ALTSEC command using an indirect file. This message will be followed by additional messages to help you isolate the problem.

ACTION The message printed by the command interpreter after this message will indicate the actual error and the position where that error occurred. Refer to the descriptions of those messages for the appropriate action(s) to be taken.

7402 MESSAGE `ERROR OCCURRED IN ACD PAIR NUMBER !.`

CAUSE A syntax or semantic error occurred while parsing an ACD specification in an ACD indirect file. This message indicates the "pair number" where the error occurred. The actual syntax or semantic error will be stated in the next message issued by the command interpreter.

If the ACD specification is for any of the following `:ALTSEC` operations `;ADDPAIR, ;REPPAIR, ;NEWACD`, then a pair will consist of a modes specification followed by a list of users. If the ACD specification is for the `;DELPAIR` operation then a pair refers to the user name (the modes specification is not necessary).

Examples:

`:ALTSEC` *filename*`;NEWACD=`*indirect*

`where` *indirect* `contains:`

(*r,w,l:user1.acct1, user2.acct2; none: @.@*)

`:ALTSEC` *filename*`;DELPAIR=`*indirect*

`where` *indirect* `contains:`

(*user1.acct1, user2.acct2, @.acct3, @.@*)

ACTION Correct the syntactic or semantic error in you ACD indirect file and re-issue the `:ALTSEC` command.

7403 MESSAGE `ACD INTERNAL STATUS ! - SUBSYSTEM NUMBER !.`

CAUSE An unexpected internal error has occurred.

ACTION Try re-issuing the command. If you still get this error, call in the internal error number to your HP Representative.

Index