900 Series HP 3000 Computer Systems

HP Security Monitor/iX User's Guide

HP Part No. 32650-90454
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 four chapters, two appendixes, and an index as follows:

Chapter 1	*Introduction* provides an overview of security on the system.
Chapter 2	*Accessing the System* describes the process of selecting a password and logging on to the system.
Chapter 3	*Protecting Your Files with Access Control Definitions* explains the use of ACDs.
Chapter 4	*Protecting Your Files with Capabilities, File Access Restrictions and Lockwords* describes additional ways to protect your files.
Appendix A	*Error Messages* contains a brief explaination of each error message.

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`:

`COMMAND` { `ON` / `OFF` }

`[ ]` 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.

`COMMAND` *filename* [`OPTION` / *parameter*]

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.

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

`...`

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

The HP Security Monitor/iX User's Guide

The Hewlett-Packard Security Monitor/iX User's Guide is written for general users of HP 3000 systems. It contains an explanation of the basic security features and a discussion of security policy and concerns. For more information on the security in place on your system, see your security administrator.

- Physical Security
- Procedural Security
- System Security
- Security Policy

Physical Security

Physical security involves the prevention of physical damage to system hardware, and prevention of the corruption of software. The causes of damage to hardware and software can range from deliberate sabotage or vandalism, to inadvertent damage caused by unskilled users.

Physical access to hardware is usually effected by perimeter controls, which restrict entry into areas in which computer equipment is located.

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, using passwords, lockwords, and 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. 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, and the way 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 will 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 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 is 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.

Defining 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 operation. Job 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 UDC,s (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 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.

Security Policy

The set of laws, rules, and practices that regulate how an organization manages, protects, and distributes sensitive information. See your system manager for a copy of the current security policy.

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 1-1 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 1-1 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.

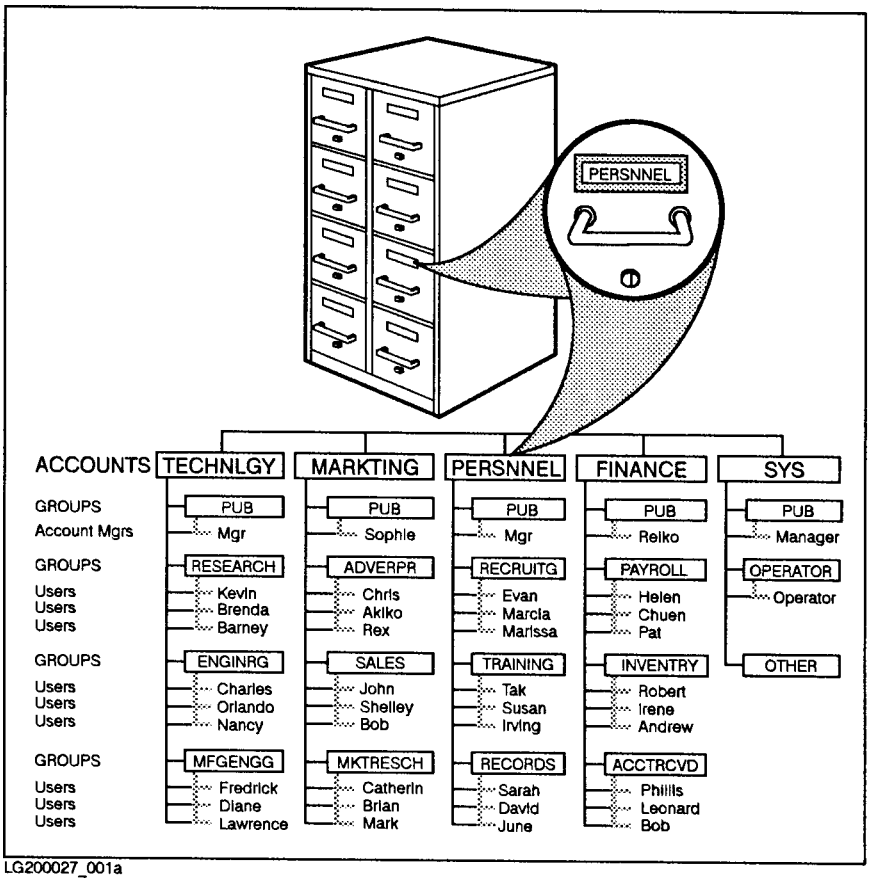


Figure 1-1. Account Relationships

Notice in Figure 1-1 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 1-2 shows the structure of an individual account. Not all accounts look like the one in Figure 1-2, 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.

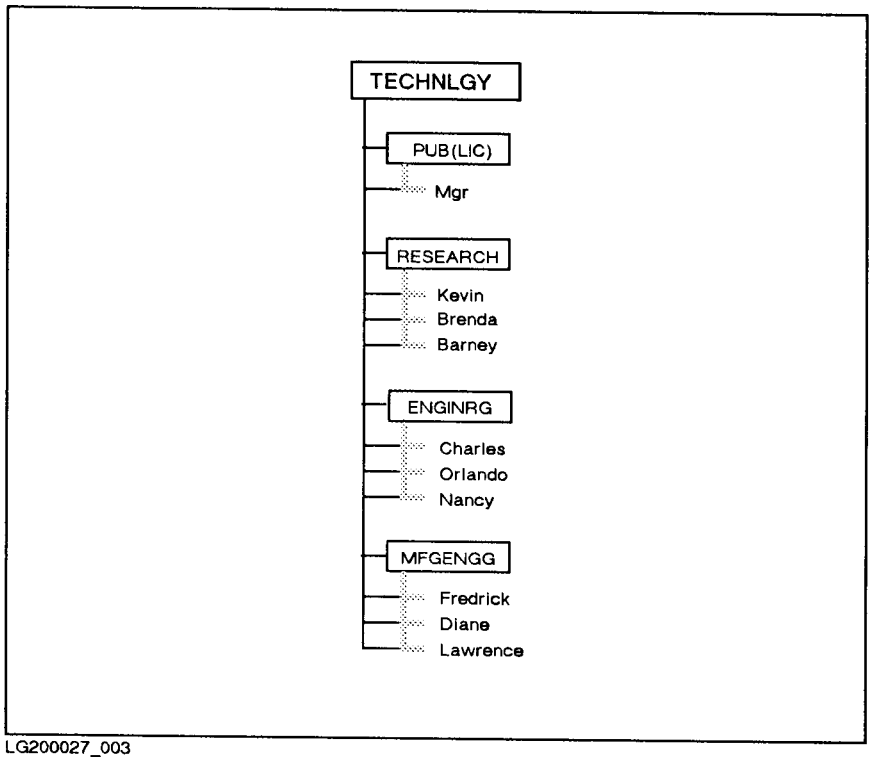


Figure 1-2. An Individual Account

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.

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 1-3.

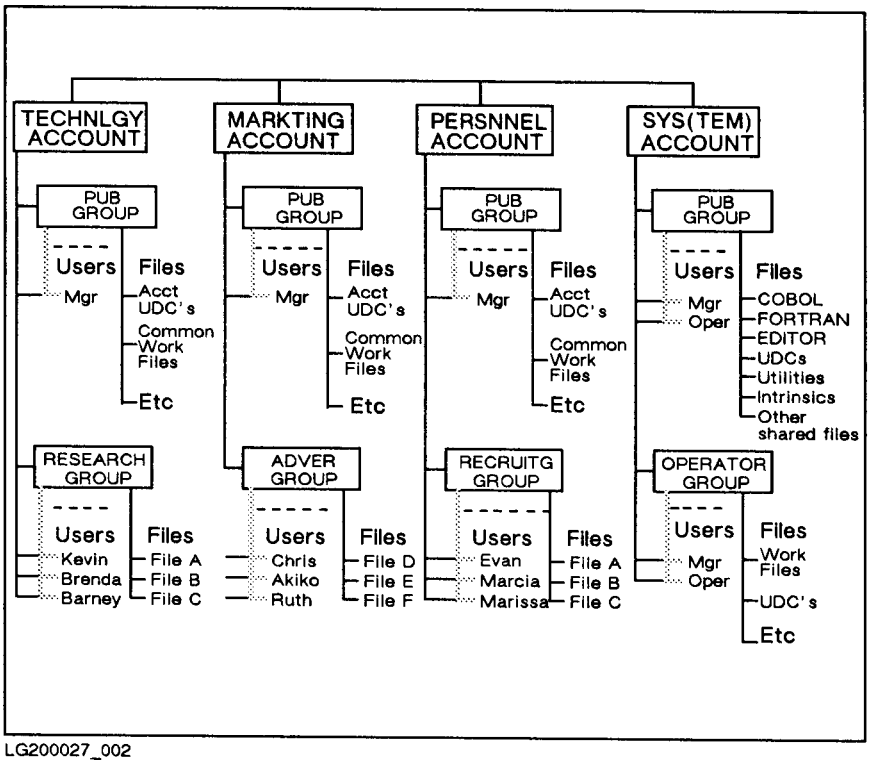


Figure 1-3. Groups, Users, and Files

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 1-3 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 1-1, 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).

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.

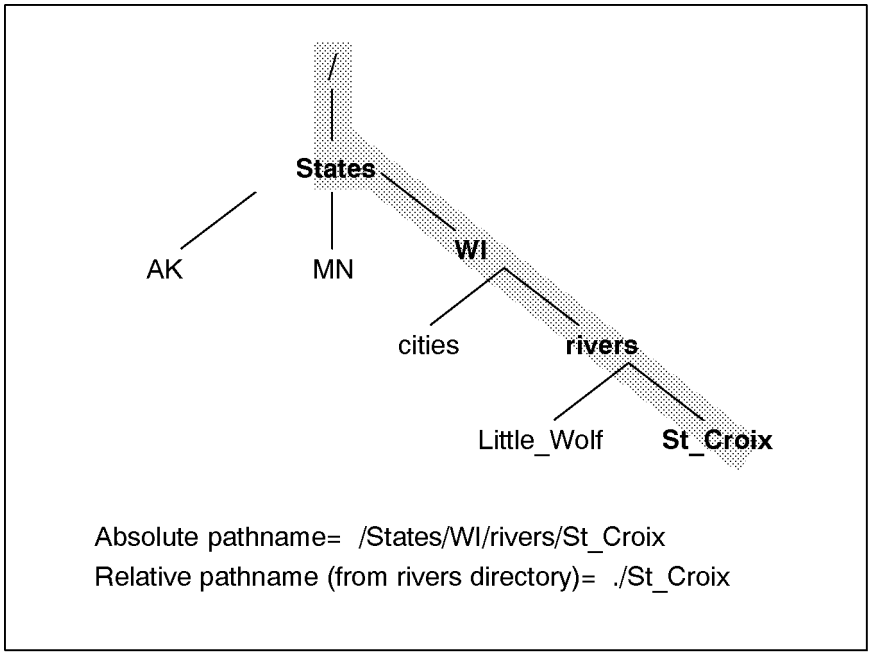
You're used to referring to files, groups, and accounts using the traditional MPE syntax: `FILE1.PUB.SYS`. You can still use MPE syntax. You can also make use of a new syntax called HFS syntax, which looks like this: `/SYS/PUB/FILE1`.

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

Table 1-1.
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.		

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*.



LG200208_007

Figure 1-4. 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 1-2 summarizes name lengths for accounts, groups, directories, and files previous to Release 4.5 and after Release 4.5.

Table 1-2.
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 1-3 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 1-3. 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)

2

Accessing the System

Getting Started

Logging on means identifying yourself to the computer. You must identify yourself as an authorized user by typing your *logon identity* (user name and account) and a password. If you do not have a logon identity, ask your system administrator to give you one. If this is the first time you have logged on to this system you will be asked to select a password.

To Log On

- After switching on the terminal, press Return one or more times until you see the system prompt.
- Type your logon identity. For example:

```
MPE XL:HELLO PAT.FINANCE
```

- Press return.
- Type any required passwords.

As a security precaution, passwords are not displayed on the screen as you type them.

- Type return.

A system prompt (:) signals the start of your session. A welcome message may also be displayed.

Note

If this is your first time to log in, you will be prompted to choose and enter your new password at this time.

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 notorious 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.

Protecting Your System with Passwords

User passwords add an additional level of security to your system. User passwords are used to authenticate individual users and to prevent unauthorized individuals from accessing the system. When an account, group, or user does not have a password, it is said to have a blank password. Without a password, an account or group is open to access by anyone who knows its name.

Note

The System Manager can specify that a password be required by a particular user or it can be left to the discretion of the user when the following option is invoked with the NEWACCT and NEWUSER commands:

```
;userpass=(req or opt)
```

This option can also be used with the ALTACCT and ALTUSER commands.

To ensure adequate password security, passwords should contain from five to eight alphanumeric characters, beginning with an alphabetic character. When assigning passwords, you should not assign names of friends or relatives. User passwords should be changed quarterly or when someone leaves the organization. All other MPE passwords should be changed every three, six, or twelve months, depending on the data sensitivity on your system.

If you are an account manager, you can assign, change, and list group or user passwords only within your own account.

Use the NEWACCT, NEWGROUP, and NEWUSER commands to create passwords for a new account, group, and user, respectively. Use the ALTACCT, ALTGROUP, and ALTUSER commands to modify existing passwords.

Changing Your Password

You can change your 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.

Do not write your password down where someone may find it. Do not use any passwords that would be easy to guess.

If Your Password Expires

Passwords that never change present a security risk to the system. System and Account Managers can cause individual user passwords to expire using standard system facilities. These facilities are the USERPASS=EXPIRED options of the :NEWUSER and :ALTUSER commands.

In addition, the system can be set so that all required passwords in the system can be made to expire simultaneously at specified intervals. When such intervals occur, users must enter new passwords or find themselves locked out of the system.

Discussion

On the expiration date, only user passwords that were not changed during the warning period expire. Users with expired passwords must select a new password the next time they log on. For example, suppose Susan has allowed her password to expire. When she logs on, she sees the following:

```
:HELLO SUSAN.MYACCT,LAPIN
ENTER ACCOUNT PASSWORD:              (Susan enters password)

ENTER USER PASSWORD:                 (Susan enters password)

ENTER GROUP PASSWORD:                (Susan enters password)

USER PASSWORD HAS EXPIRED
ENTER NEW PASSWORD:                  (Susan enters new password)
ENTER NEW PASSWORD AGAIN:            (Susan enters new password again)
PASSWORD WAS CHANGED SUCCESSFULLY
```

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 go though the procedure again. A user will not be allowed to log on until a new password is successfully entered.

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 password, 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 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 Encryption

When the password encryption feature is enabled by the system manager, 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

The MPE/iX commands that display passwords (`:LISTUSER`, `:LISTGROUP`, and `:LISTACCT`) display them in encrypted form. Passwords are encrypted in such a way that even privileged users can not recover a forgotten password. If you forget your password, you will have to pick a new one.

MPE/iX lets your system 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. Old passwords stay unencrypted. As new passwords are added or old ones changed, the system encrypts them automatically.

Caution

Your old password is not encrypted. The system leaves your old password unencrypted until you change it.

Minimum Password Lengths

MPE/iX permits passwords of up to eight characters. The longer the password, the more difficult it is for it to be discovered by trial and error. A minimum length for passwords can be set by the System Manager. This minimum length affects all account, group, and user passwords.

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).

Mandatory Password Prompts

MPE/iX allows embedded passwords in logon commands. Embedded passwords are displayed on the screen when a user logs on, and can be read by any one with a clear view of the screen. To eliminate this possibility, an HP Security Monitor facility called Mandatory Password Prompt prevents the use of embedded passwords, and prompts for passwords instead. A prompted password is not displayed on the screen when it is entered.

Discussion

Without the Mandatory Password Prompt feature in effect, MPE/iX users can log on to the system by embedding passwords within a logon command. For example, the account password `PASSWORD` is embedded in the following logon command and clearly displayed on the screen.

```
:HELLO JOE.SMITH/PASSWORD
```

With the Mandatory Password Prompt feature in effect, this is not allowed. Instead, a user enters a logon command without a password, then enters the password in answer to a prompt. The password is not displayed.

If a user tries to enter embedded passwords with Mandatory Password Prompt active, the system ignores the embedded passwords, refuses to allow the user to logon, and displays the following message:

```
PASSWORD PROMPTS ARE REQUIRED, EMBEDDED PASSWORDS ARE NOT
ALLOWED. (CIERR 1449)
```

Note

If you have attempted to log on with an embedded password and gotten an error message, be careful to clear the screen so that your password will not be discovered.

Controlling System Access with Logon Restrictions

Terminating Sessions on Initial UDC Failure

When the UDC Initiation Failure Option is enabled by the system manager, any failed UDC initiation causes the job/session to terminate and a message to this effect is sent to the system console. The only exception to this is when the MANAGER.SYS or OPERATOR.SYS logs on from the system console. Here are the types of errors which will cause a job/session to terminate in a UDC initiation:

- Failure to access COMMAND.PUB.SYS (fopen, Fwrite ... errors)
- Failure to access or read a UDC file for System or Account level UDC.

Failing to access a user-level UDC file and other non-fatal errors will only result in warnings, and the session/job will be able to continue.

Note

Should the initiation of the UDC facility (and thus the logon UDCs) fail, you may lose control over the user environment, depending on your level of capability.

Limiting the Number of Logon Attempts

The Maximum Invalid Logon Attempts security option limits password guessing by disabling user entries and terminal devices in response to repeated erroneous logon attempts.

This option works by keeping track of invalid logon attempts for specified user ID's and devices and compares them to the maximum allowable logon attempts established by the System Manager. When the maximum is exceeded, the user or device is disabled, usually for a specified period of time referred to as a "down-time interval".

Note

See your System Manager if your device fails to return after going down from too many invalid logon attempts.

Attempts to log on using a disabled user ID will be aborted with the exception of the MANAGER.SYS who can still logon from the physical system console even when this user entry has been disabled.

This exception is intended to prevent situations where a system is inaccessible because all user ID's have been disabled.

Batch data and jobs cannot be submitted using user ID's associated with disabled user entries. Attempts to submit batch data or jobs using disabled user ID's will be aborted.

User ID's with non-zero down-time intervals are automatically enabled after their downtime intervals have elapsed. When the user entry becomes available to the sytem, the invalid logon attempt count will be reset to 0. Upon successful logon to the sytem, the count is also reset to 0.

The :LISTUSER command shows which user ID's are currently disabled. User ID's with zero down-time intervals, must be manually re-enabled using the Security Configuratation Utility's User Security Options. The User Security Opions may also be used to enable a disbaled user ID with a non-zero down-time interval before it has elapsed. After the entry is re-enabled, the down-time interval count is reset to zero.

Only interactive logon devices (terminals) are affected by the maximum invalid logon attempts feature; tape drives and the STREAM device are not included. When a user exceeds the maximum number of logon attempts for a particular terminal, the system removes the terminal from use just as if you had entered the `:DOWN` command for that device at the System Console. After such an occurrence, the system displays the following message on the System Console:

```
LDEV #nn IS DOWNED, FAILED LOGON ATTEMPTS EXCEED LIMIT.
```

(where nn is the ldev number of the downed device).

The system security may be configured so that devices that have been set `:DOWN` for security reasons will automatically return to user availability after a specified amount of time. The device may also be set to stay down until it is manualy reset by the System Manager.

Note

The following is considered a single invalid logon attempt if the Maximum Invalid Logon Attempts security option is set to three:

- 3 consecutive invalid device password matches.
- 3 consecutive invalid account password matches.
- 3 consecutive invalid user password matches.
- 3 consecutive invalid group password matches.

Providing Minimal Logon Assistance

Normally, when users make a mistake while logging on, the system helps by identifying the mistake. For example, when a user enters an invalid logon command, the system displays one or more of the following messages:

```
EXPECTED [SESSION NAME,] USER.ACCT[,GROUP] (CIERR 1424)
EXPECTED ACCOUNT NAME. (CIERR 1426)
EXPECTED GROUP NAME. (CIERR 1429)
```

If your system configured for minimal logon assistance, such messages will not be displayed. This helps prevent users who are not familiar with your logon procedures from accessing the system. With minimal logon help, a user entering an invalid logon command sees only the message:

```
*INVALID*
```

Dealing with Embedded Passwords in Remote Logons

Many applications and subsystems use embedded passwords within remote logon commands. An example is the case of accessing a database residing on another system.

When your system receives such logon attempts, it treats them as if they had been issued interactively. For example, if your system is set up to require password prompts, applications or subsystems will fail if they issue remote logon commands that contain embedded passwords.

To minimize or eliminate the problem, the HP Security Monitor allows remote logons to be exempt from the password prompt requirement. After the system is set to require password prompts, the HP Security Monitor allows the option of exempting remote logons.

Rather than exempting remote logons, users may be asked to alter jobs that contain embedded passwords so they comply with the second example, below. The first example illustrates a job with a password embedded in the `REMOTE HELLO` command line. The second example illustrates a job with a password on the following line, anticipating the prompt.

"Example 1".

```
!JOB USER.ACCT
!DSLINE X
!REMOTE HELLO USERX/PASSWORD.ACCT
!COMMENT
```

"Example 2".

```
!JOB USER.ACCT
!DSLINE X
!REMOTE HELLO USERX.ACCT
```

```
PASSWORD
!COMMENT
```

The first example will be rejected if embedded passwords and exempting REMOTE HELLO are not allowed. The second example is acceptable whether or not embedded passwords are allowed.

Note

Embedded passwords in remote sessions are not recommended.

Passwords in Batch Submissions

Just as embedded passwords are a source of security exposure in sessions, so too are passwords embedded in batch submissions.

Prevention of password exposure in batch submissions is effected by:

- Rejecting embedded passwords in job cards.
- Prohibiting cross streaming (letting a user stream another user's job).
- Allowing users, AM, and SM to stream jobs without supplying passwords (stream privilege).

The first feature causes a job to fail if it contains an embedded password. The second feature places limitations on who can stream jobs of other users. The third feature permits various types of users to stream jobs without supplying passwords.

The third feature does not mean the system is now left open to the submission of unauthorized batch files. It means that a user with stream privilege can stream a job without entering a password. A user without stream privilege still must enter a valid password to stream a job.

Embedded Passwords in Job Files

This option prevents embedding passwords in job files. It operates by rejecting any "!JOB" command with an embedded password(s), regardless of the validity of the password.

When this feature is enabled, and a password is found in a job, the `:STREAM` command returns the following message:

```
PASSWORD SECURITY ENABLED. EMBEDDED PASSWORDS ARE NOT ALLOWED
(CIERR 1450).
```

This feature is applicable to all JOB cards (the ":JOB" command strings) regardless of the origin of the job. This includes jobs streamed from disk files, tapes, from within a job, or from interactive terminal input, including jobs generated by the interactive `:JOB` command. The default for this feature is "OFF".

Restricting Job Cross Streaming

To preserve accountability of each individual user, the security administrator will be able to disallow cross streaming. This prevents a person without SM or AM (of the appropriate account) from streaming jobs that log on as another person.

The Cross Streaming Authorization Option

Since these rules may be too restrictive in some situations, an exception to the rule is provided in the form of the *Cross Streaming Authorization* option. If your system manager has selected this option, it allows limited cross streaming on certain *protected* jobs.

A *protected* job is defined as a job that is streamed from a permanent file and logs on to the same user as the creator of the file. The job is protected in the sense that the job owner (user accountable for the job) also controls the file (as its creator).

A user, other than the System Manager, Account Manager, and the job owner, is authorized to stream a *protected* job when:

1. The user has EXECUTE access to the job file, and
2. The security administrator has enabled the Cross Streaming Authorization feature.

When a job submission attempt violates the cross streaming restriction, the following error message is returned to the user:

```
CROSS STREAMING IS DISALLOWED. JOB STREAM ATTEMPT FAILED
(CIERR 1440).
```

Eliminating Password Exposure with the Stream Privilege Option

This section explains how to reduce the chance of password exposure by using the Stream Privilege option.

Stream Privilege Option Features

This option provides the following features:

- Allows system processes to stream jobs without passwords.
- Allows the System Manager, Account Manager, and job owners to stream jobs without supplying passwords.
- Provides Stream Privilege Authorization to let users other than System Managers, Account Managers, and job owners stream jobs without supplying passwords.

When password verification is waived under this privilege, passwords are ignored if present. Note that if the Embedded Password Disallowed option is enabled, the stream attempt fails if an embedded password is present.

The Stream Privilege feature is independent of the Cross Streaming restriction. System Managers, Account Managers and job owners always have the right to stream jobs within their domain of control, even with the cross streaming restriction in effect. On the other hand, they do not have the right to bypass password authentication when the Stream Privilege feature is not enabled.

Stream privilege can be granted at two levels:

1. System Managers, Account Managers, and job owners only, this is the more restrictive of the two.
2. Additional authorization on protected jobs, this extends the privilege to other users when streaming protected jobs to which they have EXECUTE access.

Recommendation:. If nested jobs (jobs that are streamed from within another job) are used, Stream Privilege should be enabled. This lets System Managers, Account Managers, and job owners stream the nested job without passwords. (Make sure any passwords are removed, and ensure the outer job has proper capability to stream the nested job).

Similarly, enable the Stream Privilege when running device-direct jobs, such as those that come directly from tapes. This lets these jobs run without passwords.

When enabled, the Stream Privilege option also applies to system processes. This is the case because system processes are associated with `MANAGER.SYS` and therefore, share the same attributes and capabilities.

3

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.

Table 3-1. 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.
EXECUTE	X	Allows users to run programs stated in files with the `RUN` command or the `CREATE` and `CREATEPROCESS` intrinsics.

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 3-2. 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. When the feature is set, an file created without an associated ACD can be accessed only by its creator, SM, AM, and no one else.
2. When a file is associated with an ACD, the ACD rules, regardless of the status of the Maximum Protection feature.
3. When the feature is not enabled, and no ACD is present, file access can be controlled using the `:ALTSEC` command.

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.

4

Protecting Your Files with Capabilities, File Access Restrictions and Lockwords

File System Security Features

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

The recommended file system security feature, "Access Control Definitions," is described in a previous chapter.

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 are used to control access to 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 4-1 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.

Account, Group, and User Capabilities

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 4-1, 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

Note

If the password is encrypted, the commands LISTUSER, LISTGROUP, and LISTACCT will only display the password as "*ENCRYPTED*", making a password truly private to its owner.

Listing Account Capabilities

Use the LISTACCT command to check the capabilities of an account. To check the capabilities for the SMITH account enter:

```
LISTACCT SMITH
```

The following account information appears on the screen:

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

DISC SPACE:   754115 (SECTORS)   PASSWORD: *ENCRYPTED*
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.

The System Manager 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 Group Capabilities

Use the LISTGROUP command to display capabilities for one or more groups. For account managers (AM), 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 of the group ENGR in the account to which you are logged on, enter:

```
LISTGROUP ENGR
```

The screen displays:

```
******************
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.

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 User Capabilities

Use the LISTUSER command to check the capabilities of a user. For example, to review the capabilities of the user BORIS in the JONES account, enter:

```
LISTUSER BORIS
```

The screen displays:

```
********************
USER: BORIS.JONES
HOME GROUP:   DEVELOP             PASSWORD:    *ENCRYPTED*
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 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).

Capabilities Table

Table 4-1 lists MPE/iX capabilities and their standard abbreviations. It also shows the types of users that require each capability. Use the information in Table 4-1 to establish capabilities for your system.

Table 4-1. 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

Table 4-1. Capability Assignments (continued)

Capability		Default User	Program	Account Manager	System Supervisor	System Manager
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.

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 4-2 lists file access modes, the codes used to reference them, and their meanings.

Table 4-2. 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 4-3 lists user types, the codes used to reference them, and their complete descriptions.

Table 4-3. 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.

Note

As soon as an ACD is attached to a file all other file matrix restrictions are ignored.

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.

Account file access restrictions are set when an account is created. 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.

At the account level, the system recognizes two user types and five access modes. 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 3-1 for access mode descriptions and to Table 4-3 for user type descriptions.

If the file access restrictions for an account are not explicitly stated, 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. Access modes can be assigned to 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 3-1 for access mode descriptions and to Table 4-3 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 3-1 and Table 4-3 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 4-4.

Table 4-4. 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 the account manager (with AM capability), you can display file lockwords with the LISTFILE or LISTF command, file lockwords with the LISTFILE or LISTF commands, documented in *MPE/iX Commands Reference Manual Volumes 1 and 2* (32650-90003 and 32650-90364).

Note

Lockwords are not encrypted and a user with SM or AM capabilities can view them.

Figure 4-1 illustrates how lockwords and passwords work at different levels.

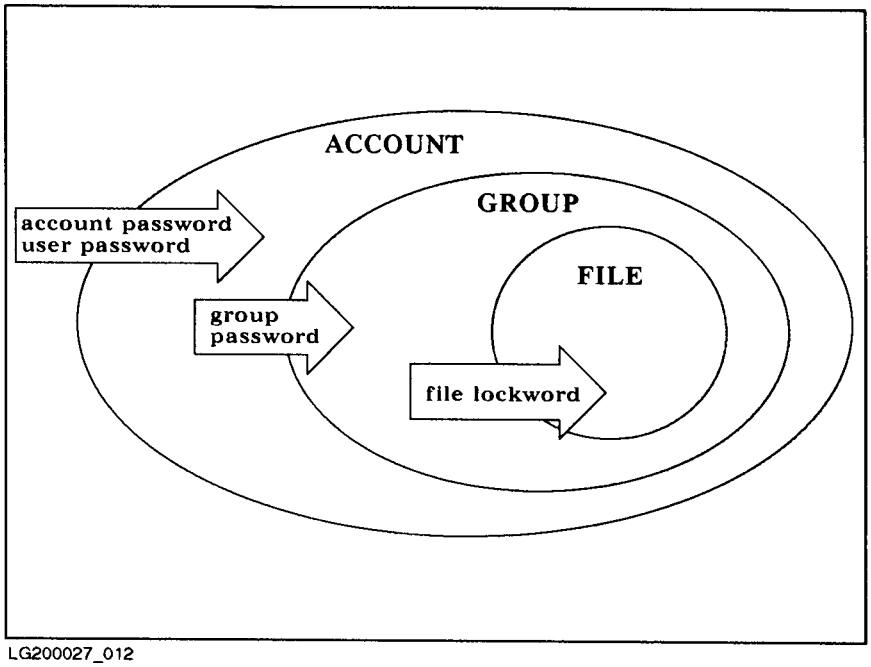


Figure 4-1. Lockwords and Passwords

Note

Lockwords should not be used on files that have ACDs attached to them.

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 and securing a file can be executed only by the creator of that file.

Note

Releasing a file removes all access restrictions to 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, including the ability to view lockwords.
- 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.

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.

A

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.

---

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.

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.

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.

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.

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.

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.

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.

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.

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, 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.

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.

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.

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.

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.

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.

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.

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 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.

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.

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 you may encounter 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 file specification containing 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

C