Safeguard Reference Manual

Abstract

This manual describes the syntax of commands needed to secure an HP NonStop™ system using the Safeguard software. The manual is intended for security administrators and general users.

Product Version

Safeguard G06.06, H05

Supported Release Version Updates (RVUs)

This publication supports J06.03 and all subsequent J-series RVUs, H06.08 and all subsequent H-series RVUs, and G06.29 and all subsequent G-series RVUs, until otherwise indicated by its replacement publications. Additionally, all considerations for H-series throughout this manual will hold true for J-series also, unless mentioned otherwise.

Part Number	Published
520618-030	February 2014

Document History

Part Number	Product Version	Published
520618-025	Safeguard G06.06, H05	August 2011
520618-026	Safeguard G06.06, H05	February 2012
520618-027	Safeguard G06.06, H05	August 2012
520618-028	Safeguard G06.06, H05	February 2013
520618-029	Safeguard G06.06, H05	August 2013
520618-030	Safeguard G06.06, H05	February 2014

Legal Notices

Confidential computer software. Valid license from HP required for possession, use or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

The information contained herein is subject to change without notice. The only warranties for HP products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. HP shall not be liable for technical or editorial errors or omissions contained herein.

Export of the information contained in this publication may require authorization from the U.S. Department of Commerce.

Microsoft, Windows, and Windows NT are U.S. registered trademarks of Microsoft Corporation.

Intel, Itanium, Pentium, and Celeron are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Java® is a U.S. trademark of Oracle and/or its affiliates

Motif, OSF/1, UNIX, X/Open, and the "X" device are registered trademarks and IT DialTone and The Open Group are trademarks of The Open Group in the U.S. and other countries.

Open Software Foundation, OSF, the OSF logo, OSF/1, OSF/Motif, and Motif are trademarks of the Open Software Foundation, Inc.

> OSF MAKES NO WARRANTY OF ANY KIND WITH REGARD TO THE OSF MATERIAL PROVIDED HEREIN, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.
>
> OSF shall not be liable for errors contained herein or for incidental consequential damages in connection with the furnishing, performance, or use of this material.
>
> © 1990, 1991, 1992, 1993 Open Software Foundation, Inc. This documentation and the software to which it relates are derived in part from materials supplied by the following:
>
> © 1987, 1988, 1989 Carnegie-Mellon University. © 1989, 1990, 1991 Digital Equipment Corporation. © 1985, 1988, 1989, 1990 Encore Computer Corporation. © 1988 Free Software Foundation, Inc. © 1987, 1988, 1989, 1990, 1991 Hewlett-Packard Company. © 1985, 1987, 1988, 1989, 1990, 1991, 1992 International Business Machines Corporation. © 1988, 1989 Massachusetts Institute of Technology. © 1988, 1989, 1990 Mentat Inc. © 1988 Microsoft Corporation. © 1987, 1988, 1989, 1990, 1991, 1992 SecureWare, Inc. © 1990, 1991 Siemens Nixdorf Informationssysteme AG. © 1986, 1989, 1996, 1997 Sun Microsystems, Inc. © 1989, 1990, 1991 Transarc Corporation.
>
> This software and documentation are based in part on the Fourth Berkeley Software Distribution under license from The Regents of the University of California. OSF acknowledges the following individuals and institutions for their role in its development: Kenneth C.R.C. Arnold, Gregory S. Couch, Conrad C. Huang, Ed James, Symmetric Computer Systems, Robert Elz. © 1980, 1981, 1982, 1983, 1985, 1986, 1987, 1988, 1989 Regents of the University of California.
>
> Printed in the US

Safeguard Reference Manual

Index Figures Tables

9. Disk Volume and Subvolume Security Commands

13. Security Group Commands

14. Terminal Security Commands

Figures

Tables

What is New in this Manual

Manual Information

Abstract

This manual describes the syntax of commands needed to secure an HP NonStop™ system using the Safeguard software. The manual is intended for security administrators and general users.

Product Version

Safeguard G07, H05

Supported Release Version Updates (RVUs)

This publication supports J06.03 and all subsequent J-series RVUs, H06.08 and all subsequent H-series RVUs, and G06.29 and all subsequent G-series RVUs, until otherwise indicated by its replacement publications. Additionally, all considerations for H-series throughout this manual will hold true for J-series also, unless mentioned otherwise.

Part Number	Published
520618-030	February 2014

Document History

Part Number	Product Version	Published
520618-025	Safeguard G06.06, H05	August 2011
520618-026	Safeguard G06.06, H05	February 2012
520618-027	Safeguard G06.06, H05	August 2012
520618-028	Safeguard G06.06, H05	February 2013
520618-029	Safeguard G06.06, H05	August 2013
520618-030	Safeguard G06.06, H05	February 2014

New and Changed Information

Changes to the 520618-030 manual

- Updated the description of IN filename on page 3-2.
- Updated the section OBEY command Considerations on page 4-21.
- Updated the description of OUT [listfile] on page 3-3.
- Added a reference to the PASSWORD-ERROR-DETAIL attribute on page 16-29.

Changes to the 520618-029 manual

- Updated the following sections:
  - Security Groups on page 1-7.
  - Security Group Commands on page 13-1.
- Updated the following commands:
  - ADD SECURITY-GROUP Command:
    - Attribute `sec-group-spec` on page 13-5.
    - Example on page 13-9.
  - ALTER SECURITY-GROUP Command:
    - Attribute `sec-group-spec` on page 13-11.
    - Example on page 13-15.
  - DELETE SECURITY-GROUP Command:
    - Attribute `sec-group-spec` on page 13-16.
    - Example on page 13-16.
  - FREEZE SECURITY-GROUP Command:
    - Attribute `sec-group-spec` on page 13-17.
    - Example on page 13-17.
  - INFO SECURITY-GROUP Command:
    - Attribute `sec-group-spec` on page 13-19.
    - Example on page 13-22.
  - SET SECURITY-GROUP Command:
    - Attribute `sec-group-spec` on page 13-25.
    - Example on page 13-30.
  - THAW SECURITY-GROUP Command:
    - Attribute `sec-group-spec` on page 13-33.
    - Example on page 13-33.
- Updated the table Logon_Data (Logon Message_Data Interactive/Programmatic) on page 15-20.
- Added a note to the section STOP SAFEGUARD Command on page 16-2.
- Added the attribute PROMPT-BEFORE-STOP to the section ALTER SAFEGUARD Command on page 16-4.

- Added the attribute PROMPT-BEFORE-STOP {ON | OFF} on page 16-29.
- Added error messages on the following pages:
    - Page A-7.
    - Page A-37.

Changes to the 520618-028 manual

- Updated the section Security Groups on page 1-7.
- Updated the STATIC FAILED LOGON COUNT = count on page 5-32.
- Added a note for the OBJECTTYPE DISKFILE/VOLUME/SUBVOLUME in the following:
    - Page 1-6
    - Page 12-3
    - Page 12-8
    - Page 12-14
    - Page 12-20
    - Page 12-27
- Updated the chapter Security Group Commands on page 13-1.
- Updated the section ADD SECURITY-GROUP Command on page 13-4.
- Updated the section Example on page 13-9.
- Updated the section Example on page 13-15.
- Updated the section Example on page 13-16.
- Updated the section FREEZE SECURITY-GROUP Command on page 13-17.
- Updated the section Example on page 13-17.
- Updated the section INFO SECURITY-GROUP Command on page 13-18.
- Updated the section Example on page 13-22.
- Updated the section SET SECURITY-GROUP Command on page 13-25.
- Updated the section Example on page 13-30.
- Updated the section ADD EVENT-EXIT-PROCESS Command on page 15-2.
- Updated the section ALTER EVENT-EXIT-PROCESS Command on page 15-6.
- Updated the section INFO EVENT-EXIT-PROCESS Command on page 15-11.
- Updated the section Timeout Policy for Authorization on page 15-27.

- Updated the section Event-Exit Design, Management, and Operation on page 15-33.
- Added error messages on the following pages:
  - Page A-6 .
  - Page A-7.
- Added warning messages on the following pages:
  - Page A-36.
  - Page A-37.

Changes to the 520618-027 manual

- Added note for the `year` on page 5-15.
- Updated the maximum value for STATIC FAILED LOGON COUNT = count on page 5-32.
- Added examples for OWNER and OWNER-LIST attributes in the Examples on page 7-7 and other relevant places in the chapter.
- Added the attribute OWNER-LIST [[-]user-list] on page 7-10 and other relevant places in the chapter.
- Added the attribute COMPARE on page 16-4.
- Added PASSWORD-ERROR-DETAIL {ON | OFF} on page 16-29.

Changes to the 520618-026 manual

- Added SECURITY-AUDITOR to Security Groups on page 1-7.
- Added a Note regarding new access authorities (WRITE and PURGE) in the following sections:
  - OBJECTTYPE Access Authorities on page 12-2.
  - ADD OBJECTTYPE Command on page 12-8.
  - ALTER OBJECTTYPE Command on page 12-14.
  - INFO OBJECTTYPE Brief Report on page 12-19.
  - SET OBJECTTYPE Command on page 12-26.
- Added information on the SECURITY-AUDITOR security group in Security Group Commands.
- Updated information under PASSWORD-MINIMUM-LENGTH n and PASSWORD-MAXIMUM-LENGTH {n}.
- Added Note under PASSWORD-MIN-QUALITY-REQUIRED on page 16-32.

- Added new error messages on pages A-5 and A-6, and a warning message on page A-36.

Changes to the 520618-025 manual

- Added a Note on page 5-2 to the User Security Commands section.
- Added a Note on page 6-2 to the User Alias Security Commands section.
- Updated the attribute CLEARONPURGE-DISKFILE { ON | OFF } on page 16-14.
- Added a Note on page 13-1 to the Security User Group section.
- Added SAVED-DISCFILE-PATTERN to the OBJECTTYPE list in the following:
    - ADD OBJECTTYPE Command on page 12-5
    - ALTER OBJECTTYPE Command on page 12-11
    - DELETE OBJECTTYPE Command on page 12-16
    - FREEZE OBJECTTYPE Command on page 12-16
    - INFO OBJECTTYPE Command on page 12-18
    - SET OBJECTTYPE Command on page 12-24
- Added the attribute PROCESS-ACCESS to the following sections:
    - ADD DISKFILE Command on page 8-7.
    - ALTER DISKFILE Command on page 8-23.
- Added examples for the PROCESS-ACCESS attribute to the following sections:
    - ADD DISKFILE Command on page 8-15.
    - ALTER DISKFILE Command on page 8-30.
    - INFO DISKFILE on page 8-50.
- Replaced U_A_ with USER_AUTHENTICATE_ on following pages:
    - 8-13.
    - 8-29.
    - 8-66.
- Replaced “PRS group” with “group 86” on page 8-15.

Changes to the H06.22/J06.11 manual

- Updated the Safeguard product version on page -1.
- Added SECURITY-PRV-ADMINISTRATOR to Security Groups on page 1-7.
- Updated the description of the following user attributes:

  - INITIAL-PROGRAM [prog-path] on page 5-18.
  - INITIAL-PROGTYPE [prog-type] on page 5-18.
- Added information on the SECURITY-PRV-ADMINISTRATOR security group in Security Group Commands.
- Corrected the order of the Subject data fields in Table 15-3, Subject_Data, on page 15-17.
- Added a note specifying that the following password attributes support the DES and HMAC256 password algorithms:
  - PASSWORD-UPPERCASE-REQUIRED on page 16-30.
  - PASSWORD-LOWERCASE-REQUIRED on page 16-30.
  - PASSWORD-NUMERIC-REQUIRED on page 16-31.
  - PASSWORD-SPECIALCHAR-REQUIRED on page 16-31.
- Added the following attributes and their descriptions to the ALTER SAFEGUARD Command:
  - PASSWORD-MIN-UPPERCASE-REQ on pages 16-8 and 16-33.
  - PASSWORD-MIN-LOWERCASE-REQ on pages 16-8 and 16-34.
  - PASSWORD-MIN-NUMERIC-REQ on pages 16-8 and 16-34.
  - PASSWORD-MIN-SPECIALCHAR-REQ on pages 16-8 and 16-35.
  - PASSWORD-APLHA-REQUIRED on pages 16-8 and 16-35.
  - PASSWORD-MIN-APLHA-REQ on pages 16-8 and 16-36.
- Updated the description of the PASSWORD-MIN-QUALITY-REQUIRED attribute on page 16-32.
- Added new error messages on pages A-5 and A-6, and a warning message on page A-36.

Changes to the H06.21/J06.10 Manual

- Added information on Safeguard Helper Process (SHP) on page 1-11.
- Added SYNC VOLUME to Table 1-1, Who Can Use SAFECOM Commands, on page 1-13.
- Updated the list of owners who can control the PROGID attribute on page 8-2.
- Added the SAFECOM Saved Diskfile Pattern Commands section, containing the following commands, on page 8-78:
  - ADD SAVED-DISKFILE-PATTERN Command on page 8-78.
  - ALTER SAVED-DISKFILE-PATTERN Command on page 8-83.

    - DELETE SAVED-DISKFILE-PATTERN Command on page 8-89.
    - FREEZE SAVED-DISKFILE-PATTERN Command on page 8-90.
    - INFO SAVED-DISKFILE-PATTERN Command on page 8-91.
    - RESET SAVED-DISKFILE-PATTERN Command on page 8-94.
    - SET SAVED-DISKFILE-PATTERN Command on page 8-95.
    - SHOW SAVED-DISKFILE-PATTERN Command on page 8-99.
    - THAW SAVED-DISKFILE-PATTERN Command on page 8-100.
- Added SHP in the description of STOP SAFEGUARD Command on page 16-2.
- Added DYNAMIC-PROC-UPDATE attribute on pages 16-7 and 16-24.
- Updated the description of PASSWORD-UPPERCASE-REQUIRED { ON | OFF } on page 16-30.
- Updated the description of PASSWORD-LOWERCASE-REQUIRED { ON | OFF } on page 16-30.
- Updated the description of PASSWORD-NUMERIC-REQUIRED {ON / OFF} on page 16-31.
- Updated the description of PASSWORD-SPECIALCHAR-REQUIRED {ON / OFF} on page 16-31.

Changes to the H06.20/J06.09 Manual

- Updated the description of PASSWORD-MIN-QUALITY-REQUIRED on page 16-32.
- Updated the output generated by the FUP INFO command when a file is placed automatically under the Safeguard control by using the DEFAULT-PROTECTION or PERSISTENT PROTECTION record, on page 8-2.
- Updated the Considerations section on page 8-14.

Changes to the 520618-020 Manual

- Updated notes in the following sections to include support for G-series RVUs:
    - CREATION-TIME on page 1-2.
    - RESET-STATIC-FAILED-LOGON-COUNT field on page 5-21.
    - OBJECT-TEXT-DESCRIPTION on pages 8-11, 8-27, 8-27, and 8-63.
    - AUDIT-PRIV-LOGON and PRIV-LOGON on pages 8-7, 8-23, 8-59, 8-67, 8-68, and 8-75.

    - OBJECT-TEXT-DESCRIPTION field and RESET-OBJECT-DESCRIPTION attribute on pages 8-27, 8-27, 9-10, 9-16, 9-31, 10-9, 10-15, 10-29, 11-11, 11-17, 11-18, 11-32, 12-9, 12-15, 12-27, 13-8, and 13-14.
    - AUDIT-OSS-FILTER attribute on pages 16-6 and 16-24.
    - AUDIT-TACL-LOGOFF on pages 16-6 and 16-24.
- Updated the INFO USER command display on page 5-54.
- Updated the ADD GROUP Command section on page 7-4.
- Updated the example depicting use of wild cards to specify group member names on page 7-8.
- Included wild-card pattern in the ALTER GROUP command on page 7-10.
- Updated the CHECK-DISKFILE-PATTERN settings in table B-3 on page B-5 and table B-4 on page B-6 respectively.

Changes to the H06.19/J06.08 Manual

- Added the following attributes to the user-authentication record:
    - CREATION-TIME on page 1-2.
    - Creators details on page 1-2.
- Added the following attributes and their descriptions to the INFO USER detailed report:
    - CREATION-TIME on pages 5-30 and 5-31.
    - CREATOR-USER-NAME on pages 5-30 and 5-32.
    - CREATOR-USER-TYPE on pages 5-30 and 5-32.
    - CREATOR-NODENUMBER on pages 5-30 and 5-32.
- Updated the description of AUDIT-USER-ACTION-PASS on pages 5-47 and 6-45.
- Updated the description of AUDIT-USER-ACTION-FAIL on pages 5-48 and 6-45.
- Added the following attributes and their descriptions to the INFO ALIAS detailed report:
    - CREATION-TIME on pages 6-28 and 6-30.
    - CREATOR-USER-NAME on pages 6-28 and 6-30.
    - CREATOR-USER-TYPE on pages 6-28 and 6-30.
    - CREATOR-NODENUMBER on pages 6-28 and 6-30.
- Added a note indicating Safeguard support to group numbers 65536 through 65567 and the SECURITY-ENCRYPTION-ADMIN group on page 7-4.

- Added a note on including wild-card pattern in the ADD GROUP command on page 7-5.
- Added examples to depict how wild cards can be used to specify group member names in the ADD GROUP command on page 7-7.
- Added a note on including wild-card pattern in the ALTER GROUP command on page 7-10.
- Added the following attributes and their descriptions to the INFO GROUP detailed report:
    - CREATION-TIME on page 7-17.
    - CREATOR-USER-NAME on page 7-17.
    - CREATOR-USER-TYPE on page 7-17 .
    - CREATOR-NODENUMBER on page 7-17.
- Added Considerations for the INFO GROUP command on page 7-19.
- Added the following attributes to the Example on page 7-20:
    - CREATION-TIME
    - CREATOR-USER-NAME
    - CREATOR-USER-TYPE
    - CREATOR-NODENUMBER
- Updated the note on OBJECT-TEXT-DESCRIPTION on page 8-11.
- Updated the syntax of RESET-OBJECT-TEXT-DESCRIPTION attribute on pages 8-23, 8-27, 9-13, 9-16, 10-12, 10-15, 11-15, 11-18, 12-12, 12-15, 13-11, and 13-14.
- Made the following changes in Table 15-4, Access Data (Access Control Message_Data):
    - Added ObjFilename and its description on page 15-19.
    - Added OssPathname and its description on page 15-19.
    - Updated the description of FEA on page 15-19.
- Added MID option to the CHECK-DISKFILE-PATTERN attribute on pages 16-7 and 16-28.
- Added AUDIT-OSS-FILTER attribute on page 16-6 and its description on page 16-23.
- Added AUDIT-TACL-LOGOFF attribute on page 16-6 and its description on page 16-24.

- Updated the description of WARNING-FALLBACK-SECURITY { GUARDIAN | GRANT } on page 16-26.
- Added new error messages on pages A-17, A-18, A-29, and A-32.
- Added Table B-3, CHECK-DISKFILE-PATTERN settings when value is MID and Direction Diskfile is Filename-First on page B-5.
- Added Table B-4, CHECK-DISKFILE-PATTERN settings when value is MID and Direction Diskfile is Volume-First on page B-5.

About This Manual

This reference manual presents the detailed syntax for the commands of SAFECOM, the command interpreter for the Safeguard subsystem. You use SAFECOM to establish Safeguard protection for users and system objects.

Readership of the Manual

This manual is intended for security administrators or other users who want to secure objects on their system or control user access to the system.

Organization of This Manual

Section	**Description** (page 1 of 3)
Section 1, Introduction	Introduces the Safeguard software and summarizes its features.
Section 2, Common SAFECOM Language Elements	Provides the syntax of common SAFECOM language elements.
Section 3, The Command to Run SAFECOM	Describes how to run SAFECOM.
Section 4, SAFECOM Session-Control Commands	Describes the SAFECOM session-control commands, which establish the working environment for your SAFECOM session.

Section	**Description** (page 2 of 3)
Section 5, User Security Commands Section 6, User Alias Security Commands Section 7, Group Commands Section 8, Disk-File Security Commands Section 9, Disk Volume and Subvolume Security Commands Section 10, Device and Subdevice Security Commands Section 11, Process and Subprocess Security Commands	Sections 5 through 11 contain syntax descriptions of the commands that manage protection for the following system elements: ● Users. The Safeguard software controls who can log on to the system, as well as various user security requirements. ● Aliases. The Safeguard software allows you to define alternate names for users. ● User groups. The Safeguard software allows you to define file-sharing groups for users. ● Disk files. The Safeguard software controls who can access protected disk files. ● Disk volumes and subfamilies. The Safeguard software controls who can create and access files on protected volumes and subvolumes. ● Devices and subdevices (other than disks). The Safeguard software controls who can open protected devices and subdevices. ● Processes and subprocesses. The Safeguard software controls who can use protected process names or open a process running with a protected name.
Section 12, OBJECTTYPE Security Commands	Describes the SAFECOM commands used to create and manage OBJECTTYPE records. These records define users or groups of users who can add subjects and objects to the Safeguard database.
Section 13, Security Group Commands	Describes the SAFECOM commands used to create and manage security groups. By creating security groups, you can restrict the use of certain commands.
Section 14, Terminal Security Commands	Describes the SAFECOM commands used to add and manage terminal definition records. A terminal definition record specifies a terminal at which the Safeguard software can automatically start a specific command interpreter.
Section 15, Event-Exit-Process Commands	Describes the SAFECOM commands used to define and manage a security event-exit.

Section	Description (page 3 of 3)
Section 16, Safeguard Subsystem Commands	Describes the Safeguard subsystem management commands, which are used to obtain information about the Safeguard subsystem, alter the Safeguard configuration, and stop the Safeguard software.
Section 17, Running Other Programs From SAFECOM	Describes the SAFECOM RUN command.
Appendix A, SAFECOM Error and Warning Messages and Appendix B, Disk-File Access Rules	Explains SAFECOM error messages and define disk file access rules.

Only users who have EXECUTE authority for the SAFECOM program object file can use SAFECOM. The system security administrator can secure the SAFECOM object file (using the SAFECOM ADD DISKFILE command) so that only certain individuals can use SAFECOM. In addition, some SAFECOM commands are restricted to super group members. The audit service commands and terminal commands are restricted to members of special security groups, but most commands are available to any user, with some restrictions determined by the Safeguard software.

Related Manuals

Before you use this reference manual, read:

- *Introduction to NonStop S-Series Servers*
- *Safeguard User's Guide*
- *Safeguard Administrator's Manual*
- *Safeguard Audit Service Manual*

The user's guide and administrator's manual contain task-oriented procedures for securing each element in a NonStop system. The audit service manual contains information about using the Safeguard auditing facility to monitor attempts to log on, access objects, or manage the Safeguard records for users and objects.

You might also want to have these manuals available for reference:

- *Security Management Guide*
- *Safeguard Management Programming Manual*
- *Guardian User's Guide*
- *Guardian Application Conversion Guide*

Notation Conventions

Hypertext Links

Blue underline is used to indicate a hypertext link within text. By clicking a passage of text with a blue underline, you are taken to the location described. For example:

> This requirement is described under Backup DAM Volumes and Physical Disk Drives on page 3-2.

General Syntax Notation

The following list summarizes the notation conventions for syntax presentation in this manual.

UPPERCASE LETTERS. Uppercase letters indicate keywords and reserved words; enter these items exactly as shown. Items not enclosed in brackets are required. For example:

```
MAXATTACH
```

lowercase italic letters. Lowercase italic letters indicate variable items that you supply. Items not enclosed in brackets are required. For example:

file-name

computer type. `Computer type` letters within text indicate C and Open System Services (OSS) keywords and reserved words; enter these items exactly as shown. Items not enclosed in brackets are required. For example:

```
myfile.c
```

italic computer type. *`Italic computer type`* letters within text indicate C and Open System Services (OSS) variable items that you supply. Items not enclosed in brackets are required. For example:

pathname

[ ] Brackets. Brackets enclose optional syntax items. For example:

TERM [\system-name.]$*terminal-name*

```
INT[ERRUPTS]
```

A group of items enclosed in brackets is a list from which you can choose one item or none. The items in the list may be arranged either vertically, with aligned brackets on each side of the list, or horizontally, enclosed in a pair of brackets and separated by vertical lines. For example:

```
FC [ num  ]
   [ -num ]
   [ text ]
```

```
K [ X | D ] address
```

{ } Braces. A group of items enclosed in braces is a list from which you are required to choose one item. The items in the list may be arranged either vertically, with aligned braces on each side of the list, or horizontally, enclosed in a pair of braces and separated by vertical lines. For example:

```
LISTOPENS PROCESS { $appl-mgr-name }
                  { $process-name  }
```

```
ALLOWSU { ON | OFF }
```

| Vertical Line. A vertical line separates alternatives in a horizontal list that is enclosed in brackets or braces. For example:

```
INSPECT { OFF | ON | SAVEABEND }
```

… Ellipsis. An ellipsis immediately following a pair of brackets or braces indicates that you can repeat the enclosed sequence of syntax items any number of times. For example:

```
M address [ , new-value ]…
```

```
[ - ] {0|1|2|3|4|5|6|7|8|9}…
```

An ellipsis immediately following a single syntax item indicates that you can repeat that syntax item any number of times. For example:

```
"s-char…"
```

Punctuation. Parentheses, commas, semicolons, and other symbols not previously described must be entered as shown. For example:

```
error := NEXTFILENAME ( file-name ) ;
```

```
LISTOPENS SU $process-name.#su-name
```

Quotation marks around a symbol such as a bracket or brace indicate the symbol is a required character that you must enter as shown. For example:

```
"[" repetition-constant-list "]"
```

Item Spacing. Spaces shown between items are required unless one of the items is a punctuation symbol such as a parenthesis or a comma. For example:

```
CALL STEPMOM ( process-id ) ;
```

If there is no space between two items, spaces are not permitted. In the following example, there are no spaces permitted between the period and any other items:

```
$process-name.#su-name
```

Line Spacing. If the syntax of a command is too long to fit on a single line, each continuation line is indented three spaces and is separated from the preceding line by a blank line. This spacing distinguishes items in a continuation line from items in a vertical list of selections. For example:

```
ALTER [ / OUT file-spec / ] LINE

   [ , attribute-spec ]…
```

!i and !o. In procedure calls, the !i notation follows an input parameter (one that passes data to the called procedure); the !o notation follows an output parameter (one that returns data to the calling program). For example:

```
CALL CHECKRESIZESEGMENT (  segment-id                    !i
                         , error       ) ;               !o
```

!i,o. In procedure calls, the !i,o notation follows an input/output parameter (one that both passes data to the called procedure and returns data to the calling program). For example:

```
error := COMPRESSEDIT ( filenum ) ;                      !i,o
```

!i:i. In procedure calls, the !i:i notation follows an input string parameter that has a corresponding parameter specifying the length of the string in bytes. For example:

```
error := FILENAME_COMPARE_ (  filename1:length           !i:i
                            , filename2:length ) ;       !i:i
```

!o:i. In procedure calls, the !o:i notation follows an output buffer parameter that has a corresponding input parameter specifying the maximum length of the output buffer in bytes. For example:

```
error := FILE_GETINFO_ (  filenum                        !i
                        , [ filename:maxlen ] ) ;        !o:i
```

Notation for Messages

The following list summarizes the notation conventions for the presentation of displayed messages in this manual.

Bold Text. Bold text in an example indicates user input entered at the terminal. For example:

```
ENTER RUN CODE

?123

CODE RECEIVED:      123.00
```

The user must press the Return key after typing the input.

Nonitalic text. Nonitalic letters, numbers, and punctuation indicate text that is displayed or returned exactly as shown. For example:

```
Backup Up.
```

lowercase italic letters. Lowercase italic letters indicate variable items whose values are displayed or returned. For example:

p-register

process-name

[ ] Brackets. Brackets enclose items that are sometimes, but not always, displayed. For example:

Event number = *number* [Subject = *first-subject-value*]

A group of items enclosed in brackets is a list of all possible items that can be displayed, of which one or none might actually be displayed. The items in the list might be arranged either vertically, with aligned brackets on each side of the list, or horizontally, enclosed in a pair of brackets and separated by vertical lines. For example:

proc-name trapped [in SQL | in SQL file system]

{ } Braces. A group of items enclosed in braces is a list of all possible items that can be displayed, of which one is actually displayed. The items in the list might be arranged either vertically, with aligned braces on each side of the list, or horizontally, enclosed in a pair of braces and separated by vertical lines. For example:

obj-type obj-name state changed to *state*, caused by
{ Object | Operator | Service }

```
process-name State changed from old-objstate to objstate
{ Operator Request. }
{ Unknown.          }
```

| Vertical Line. A vertical line separates alternatives in a horizontal list that is enclosed in brackets or braces. For example:

```
Transfer status: { OK | Failed }
```

% Percent Sign. A percent sign precedes a number that is not in decimal notation. The % notation precedes an octal number. The %B notation precedes a binary number. The %H notation precedes a hexadecimal number. For example:

```
%005400
```

```
%B101111
```

```
%H2F
```

P=%*p-register* E=%*e-register*

Notation for Management Programming Interfaces

The following list summarizes the notation conventions used in the boxed descriptions of programmatic commands, event messages, and error lists in this manual.

UPPERCASE LETTERS. Uppercase letters indicate names from definition files; enter these names exactly as shown. For example:

```
ZCOM-TKN-SUBJ-SERV
```

lowercase letters. Words in lowercase letters are words that are part of the notation, including Data Definition Language (DDL) keywords. For example:

```
token-type
```

!r. The !r notation following a token or field name indicates that the token or field is required. For example:

```
ZCOM-TKN-OBJNAME      token-type ZSPI-TYP-STRING.          !r
```

!o. The !o notation following a token or field name indicates that the token or field is optional. For example:

```
ZSPI-TKN-MANAGER      token-type ZSPI-TYP-FNAME32.         !o
```

Change Bar Notation

Change bars are used to indicate substantive differences between this edition of the manual and the preceding edition. Change bars are vertical rules placed in the right margin of changed portions of text, figures, tables, examples, and so on. Change bars highlight new or revised information. For example:

> The message types specified in the REPORT clause are different in the COBOL85 environment and the Common Run-Time Environment (CRE).
>
> The CRE has many new message types and some new message type codes for old message types. In the CRE, the message type SYSTEM includes all messages except LOGICAL-CLOSE and LOGICAL-OPEN.

HP Encourages Your Comments

HP encourages your comments concerning this document. We are committed to providing documentation that meets your needs. Send any errors found, suggestions for improvement, or compliments to docsfeedback@hp.com.

Include the document title, part number, and any comment, error found, or suggestion for improvement you have concerning this document.

1 Introduction

This section introduces the Safeguard software and presents important basic concepts:

- Safeguard security-management features
  - User Authentication
  - Object-access authorization
  - Auditing
  - Control of logon dialog
  - Security groups
  - File-sharing groups
  - Event-exit processes
- Definition of the terms *`authentication`*, *`remote`*, and *`local`*
- Who can use the SAFECOM commands and how this authority is granted
- Definition of the super ID’s capabilities and limitations

For more information about the Safeguard features and about creating and managing access control lists for protected objects, see the *Safeguard User’s Guide* and *Safeguard Administrator’s Manual*. For more information about using the Safeguard auditing facilities, see the *Safeguard Audit Service Manual*.

Safeguard Security-Management Features

The features of the Safeguard security-management facility fall into three categories: user authentication, object-access authorization, and auditing. This subsection briefly describes each category of Safeguard features.

User Authentication

To log on to a system controlled by the Safeguard software, a user must be authenticated by the Safeguard software. The Safeguard subsystem controls both interactive and procedural logon attempts by verifying a user’s user name and logon password. In addition, the Safeguard subsystem can manage other user attributes such as password expiration and can record attempts to log on or to access a user’s authentication record in the audit trail.

A user can also be assigned one or more aliases that can be used to log on to the system. The attribute values in a user alias authentication record can differ from values in the authentication record of the underlying user ID.

The following summary lists the user and alias authentication features. (The SAFECOM command is given in parentheses.) Any task that the owner of a Safeguard record can perform, the owner’s group manager can perform also.

- Group managers can create Safeguard authentication records (ADD USER and ADD ALIAS) unless an alternative list of users has been specified with the access control list for OBJECTTYPE USER.

  A user-authentication record contains these attributes:

  - OWNER
  - OWNER-LIST (supported on systems running G06.27 and later G-series RVUs and H06.07 and later H-series RVUs)
  - PASSWORD
  - PASSWORD-EXPIRES
  - PASSWORD-MUST-CHANGE EVERY *n* DAYS
  - PASSWORD-EXPIRY-GRACE
  - USER-EXPIRES
  - Auditing specifications
  - REMOTEPASSWORD
  - DEFAULT-PROTECTION for disk files
  - GUARDIAN DEFAULT SECURITY
  - GUARDIAN DEFAULT VOLUME
  - PRIMARY-GROUP
  - TEXT-DESCRIPTION
  - BINARY-DESCRIPTION-LENGTH
  - RESET-TEXT-DESCRIPTION
  - RESET-BINARY-DESCRIPTION
  - Command interpreter specification for logon at a Safeguard terminal
  - Initial program and directory specifications for HP NonStop Open System Services (OSS)
  - CREATION-TIME of the user (supported only on systems running J06.04 and later J-series RVUs, H06.15 and later H-series RVUs, and G06.32 and later G-series RVUs.)
  - Creator details specifying name, type, user ID, and node number where the user is created (supported only on systems running J06.04 and later J-series RVUs and H06.15 and later H-series RVUs)

- The primary and secondary owners of a record can modify the record (ALTER USER and ALTER ALIAS).

- The primary and secondary owners of a record can freeze and thaw the ability of a user or alias to log on (FREEZE USER or ALIAS and THAW USER or ALIAS).
- The primary and secondary owners of a record can delete the record (DELETE USER or DELETE ALIAS).
- The primary and secondary owners of a record can display record information using the INFO USER command.

Object-Access Authorization

With the Safeguard software, you can secure these types of system objects:

- Disk files
- Disk volumes and subvolumes
- Devices and subdevices (including terminals, tape drives, communication lines, and printers)
- Processes and subprocesses (both named and unnamed)

You protect objects by defining an access control list (ACL) with the ACCESS attribute. ACLs specify who can access an object and what authorities they have. Except for the super ID and the group manager of the protection record owner, users are implicitly denied all access authorities if they do not appear on an object's access control list.

The Safeguard software provides the object-access control features listed in the following summaries. (The SAFECOM command appears in parentheses.)

Control Features for Disk Files

- The owner of a disk file can create a Safeguard disk file authorization record (ADD DISKFILE) unless a list of users has been designated with an access control list for OBJECTTYPE DISKFILE.

  Each file authorization record contains these attributes:

  - OWNER—ownership can be transferred to another user
  - ACCESS—an access-control list to authorize access: Read, Write, Execute, Purge, Create, Owner (RWEPCO)
  - Auditing specifications
  - LICENSE
  - PROGID
  - CLEARONPURGE
  - PERSISTENT
- The owner of an authorization record for that file can modify the record (ALTER DISKFILE).

- The owner of a file authorization record can freeze and thaw access to the file (FREEZE DISKFILE and THAW DISKFILE).
- The owner of a file record can delete the record (DELETE DISKFILE).
- Diskfile patterns reduce administrative burden by supplying one pattern that can match many volumes, subvolumes, or filenames. For more information, see the *Safeguard User’s Guide.*

Control Features for Disk Volumes

- Any local super-group user can create a Safeguard disk volume authorization record (ADD VOLUME) unless specific users have been designated with an access control list for OBJECTTYPE VOLUME.

  A volume authorization record contains these attributes:

  - OWNER—ownership can be transferred to another user
  - ACCESS—an access control list to authorize access: Read, Write, Execute, Purge, Create, Owner (RWEPCO)
  - Auditing specifications
- The owner of a volume authorization record can modify the record (ALTER VOLUME).
- The owner of a volume record can freeze and thaw access to the disk volume (FREEZE VOLUME and THAW VOLUME).
- The owner of a volume record can delete the record (DELETE VOLUME).

Control Features for Subvolumes

- Any user can create a Safeguard subvolume authorization record (ADD SUBVOLUME) unless a specific list of users has been designated with an access control list for OBJECTTYPE SUBVOLUME.

  A subvolume authorization record contains these attributes:

  - OWNER—ownership can be transferred to another user
  - ACCESS—an access control list to authorize access: Read, Write, Execute, Purge, Create, Owner (RWEPCO)
  - Auditing specifications
- The owner of a subvolume authorization record can modify the record (ALTER SUBVOLUME).
- The owner of a subvolume record can freeze and thaw access to the subvolume (FREEZE SUBVOLUME and THAW SUBVOLUME).
- The owner of a subvolume record can delete the record (DELETE SUBVOLUME).

Control Features for Devices

- Any local super-group user can create a Safeguard device authorization record (ADD DEVICE) unless a specific list of users has been designated with an access control list for OBJECTTYPE DEVICE.

  A device authorization record contains these attributes:

  - OWNER—ownership can be transferred to any user
  - ACCESS—an access control list to authorize access: Read, Write, Owner (RWO)
  - Auditing specifications

- The owner of a device authorization record can modify the record (ALTER DEVICE).
- The owner of a device record can freeze and thaw access to the device (FREEZE DEVICE and THAW DEVICE).
- The owner of a device record can delete the record (DELETE DEVICE).

Control Features for Subdevices

- Any local super-group user can create a Safeguard subdevice authorization record (ADD SUBDEVICE) unless a specific list of users has been designated with an access control list for OBJECTTYPE SUBDEVICE.

  A subdevice authorization record contains these attributes:

  - OWNER—ownership can be transferred to any user
  - ACCESS—an access control list to authorize access: Read, Write, Owner (RWO)
  - Auditing specifications

- The owner of a subdevice authorization record can modify the record (ALTER SUBDEVICE).
- The owner of a subdevice record can freeze and thaw access to the device (FREEZE SUBDEVICE and THAW SUBDEVICE).
- The owner of a subdevice record can delete the record (DELETE SUBDEVICE).

Control Features for Processes

- Any user can create a Safeguard process-name record (ADD PROCESS) unless a specific list of users has been designated with an access control list for OBJECTTYPE PROCESS.

  A process name authorization record contains these attributes:

- OWNER—ownership can be transferred to any user
- ACCESS—an access control list to authorize access: Read, Write, Purge (stop), Create, Owner (RWPCO)
- Auditing specifications

- The owner of a process name authorization record can modify the record (ALTER PROCESS).
- The owner of a process-name record can freeze and thaw access to the process name (FREEZE PROCESS and THAW PROCESS).
- The owner of a process-name record can delete the record (DELETE PROCESS).

Control Features for Subprocesses

- Any user can create a Safeguard subprocess-name record (ADD SUBPROCESS), unless a specific list of users has been designated with an access control list for OBJECTTYPE SUBPROCESS.

  A subprocess name authorization record contains these attributes:

  - OWNER—ownership can be transferred to any user
  - ACCESS—an access control list to authorize access: Read, Write, Owner (RWO)
  - Auditing specifications
- The owner of a subprocess name authorization record can modify the record (ALTER SUBPROCESS).
- The owner of a subprocess-name record can freeze and thaw access to the subprocess name (FREEZE SUBPROCESS and THAW SUBPROCESS).
- The owner of a subprocess-name record can delete the record (DELETE SUBPROCESS).

Control Features for OBJECTTYPE Access Authorities

- Any local super-group user can create a Safeguard OBJECTTYPE authorization record (ADD OBJECTTYPE) unless an alternate list of users has been specified with an access control list for OBJECTTYPE OBJECTTYPE.

  An OBJECTTYPE authorization record contains these attributes:

  - OWNER—ownership can be transferred to any user
  - ACCESS—an access control list to authorize access: Create, Owner (CO)

  - ○ Auditing specifications

Note. Starting with H06.24/J06.13 RVUs, the OBJECTTYPE USER is granted additional access permissions, WRITE (W) and PURGE (P), along with the existing CREATE (C) and OWNER (O) permissions. Members having the WRITE (W) permission on OBJECTTYPE USER can modify any subject records. Members having the PURGE (P) permission on OBJECTTYPE USER can purge any subject records.

Note. Starting with H06.26/J06.15 RVUs, the OBJECTTYPE DISKFILE/VOLUME/SUBVOLUME is granted additional access permissions, WRITE (W) and PURGE (P), along with the existing CREATE (C) and OWNER (O) permissions. Members having the WRITE (W) permission on OBJECTTYPE DISKFILE/VOLUME/SUBVOLUME can modify the respective DISKFILE/VOLUME/SUBVOLUME protection records. Members having the PURGE (P) permission on OBJECTTYPE DISKFILE/VOLUME/SUBVOLUME can purge the respective DISKFILE/VOLUME/SUBVOLUME protection records.

- The owner of an OBJECTTYPE authorization record can modify the record (ALTER OBJECTTYPE).
- The owner of an OBJECTTYPE record can freeze and thaw access to the OBJECTTYPE (FREEZE OBJECTTYPE and THAW OBJECTTYPE).
- The owner of an OBJECTTYPE record can delete the record (DELETE OBJECTTYPE).

File-Sharing Groups

The Safeguard software allows you to create user groups for file-sharing purposes. With the GROUP commands, users can be assigned to multiple groups and group membership can be extended beyond 256 users. Section 7, Group Commands, describes how to create and maintain file-sharing groups.

Security Groups

The Safeguard software allows you to create the following security groups to restrict the use of SAFECOM commands: SECURITY-ADMINISTRATOR, SYSTEM-OPERATOR, SECURITY-OSS-ADMINISTRATOR, SECURITY-PRV-ADMINISTRATOR, SECURITY-AUDITOR, SECURITY-MEDIA-ADMIN, and SECURITY-PERSISTENCE-ADMIN. After the security groups are created, only the group members can execute certain TERMINAL, EVENT-EXIT-PROCESS, SAFEGUARD, and audit service commands. Security group membership also determines who can alter the Safeguard configuration and stop the Safeguard software. For more information on how to create and maintain the security groups, see Section 13, Security Group Commands.

Terminal Control

The TERMINAL commands allow you to define terminals on your system so that the Safeguard software controls those terminals. When the Safeguard software controls a terminal, you can specify that a particular command interpreter be started automatically after user authentication at the terminal. Prior to D30, an extended logon dialog was available only at Safeguard terminals. Effective with D30, the HP Tandem Advanced Command Language (TACL) command interpreter also provides these extended features as long as Safeguard is running. Section 14, Terminal Security Commands, describes how to define terminals so that they are controlled by the Safeguard software.

Event-Exit Process

The Safeguard software can be configured to pass authorization, authentication, and password-change requests to a user-written process, thereby allowing that process to participate in security policy enforcement. Section 15, Event-Exit-Process Commands, describes how to configure an event-exit process and provides guidelines for designing and developing such a process.

Auditing

A Safeguard record owner can define auditing for any protected object or user. Each Safeguard record contains auditing specifications that determine which system events are recorded in the Safeguard audit files. Each auditing specification consists of an auditing attribute and its current defined value.

The auditing specifications are fully described in the syntax for the SET commands. For more information, see Sections 5 through 13. You use a separate set of commands to manage the Safeguard audit service itself. For the audit service commands, see the *Safeguard Audit Service Manual*.

Object Auditing

The four auditing attributes for objects:

```
AUDIT-ACCESS-PASS
AUDIT-ACCESS-FAIL
AUDIT-MANAGE-PASS
AUDIT-MANAGE-FAIL
```

For protected objects, AUDIT-ACCESS attributes control the auditing of attempts to access the object. The two AUDIT-MANAGE attributes control the auditing of attempts to manage (change, read, or delete) the Safeguard protection record for that object.

The four possible values for each auditing attribute:

```
ALL
LOCAL
REMOTE
NONE
```

User Auditing

For users, the following auditing specifications are available:

```
AUDIT-AUTHENTICATE-PASS
AUDIT-AUTHENTICATE-FAIL
AUDIT-MANAGE-PASS
AUDIT-MANAGE-FAIL
AUDIT-USER-ACTION-PASS
AUDIT-USER-ACTION-FAIL
```

For users, the two AUDIT-AUTHENTICATE attributes control the auditing of user authentication attempts. The two AUDIT-MANAGE attributes control the auditing of attempts to manage (change, read, or delete) the Safeguard protection record for that user. The two AUDIT-USER-ACTION attributes control the auditing of attempts by the user to perform an event.

The four possible values for each auditing attribute:

```
ALL
LOCAL
REMOTE
NONE
```

Definition of Terms: Authentication, Local, and Remote

Two important security-related characteristics of a user are whether the user is authenticated and whether the request made by the user is local or remote. The following paragraphs define the terms *authentication*, *local user*, *remote user*, *local request*, and *remote request*.

Authentication

The verification of a user's claimed identity as a valid local user. Authentication might or might not be followed by logging the user on to the system. That is, authentication is always a part of logon, but logon does not always occur after authentication.

A user must be authenticated before logon is permitted. Typically, after a user has been authenticated, a session is started by logging the user on to the system and initializing a process to function on the user's behalf.

When a user logs on through a command interpreter, the command interpreter assumes the identity of the user by adopting the user's user ID as its PAID (process accessor ID).

Local User

A term used in this manual to refer to a locally authenticated user. A process belonging to a user who has been authenticated by the local system.

Remote User

A term used in this manual to refer to either a remotely authenticated user or an unauthenticated user.

Remotely Authenticated User

A valid network user who has been authenticated by a node other than the local node.

Unauthenticated User

A process that lacks a valid user ID; for example, a user who has failed remote validation in an attempt to access objects on the local node.

Local Request

A local request originates on the same node in which the process receiving the request is running.

Remote Request

A remote request made in relation to the process receiving the request. This request originates on another node in a network.

Interaction of Local and Remote Users and Requests

In determining whether to audit a request, the Safeguard software considers a local user making a local request to be a local request. Any request from a remote user is considered a remote request.

Thus, if REMOTE is the value of AUDIT-ACCESS-PASS for a disk file, the Safeguard software audits any attempt that it authorizes to access the file (that is, any authorized OPEN requests for READ, WRITE, EXECUTE, or PURGE access) that originate from a remote system or are sent by a remote user.

The remote user described here is not the same as the network user (defined in Section 2, Common SAFECOM Language Elements). A network user is a user who has been added to several systems and has matching remote passwords between those systems. A remote user is simply a process that is not locally authenticated. The remote user does not necessarily have a matching password definition on the local system. For more information, see Network Users on page 2-17.

Components of the Safeguard Subsystem

The Safeguard subsystem consists of a number of processes and security database files. These processes cooperate to manage the contents of the security database, to authenticate users, and to authorize attempts to access protected objects.

The following components reside on every system where the Safeguard software has been installed:

- The subject database, which contains a user-authentication record for every user authorized to use the system.
- The object database, which contains an object authorization record for every disk file, disk volume, disk subvolume, device, process name, and OBJECTTYPE protected by the Safeguard facility.
- The Security Manager Process (SMP), which runs under the process name $ZSMP and is responsible for managing all changes to the subject and object databases. The SMP also verifies that the name and password supplied by the user match a user name and associated password stored in the subject database.
- SAFECOM, which is the command interpreter that provides an interactive interface to the SMP.
- The Security Monitor (SMON), which authorizes all attempts to access protected objects. A separate SMON process runs in every CPU in a protected system. Each SMON performs authorization operations for all security-related transactions in that CPU. The SMP ensures that all SMONs are operational.
- The Safeguard Helper Process (SHP), which assists SMP to identify and update process attributes whenever the following user attributes in the user database files are modified:
    - AUDIT-USER-ACTION-PASS
    - AUDIT-USER-ACTION-FAIL
    - Primary group
    - Supplementary group list
    - Group count

A separate SHP process runs in every processor in a protected system. Each SHP updates the process attributes of every process in its own processor running with the user identity whose above-mentioned user attributes are changed.

Note. The SMP ensures that all SHPs are operational.

Who Can Use SAFECOM Commands

Only users who have EXECUTE authority for the SAFECOM program object disk file can run SAFECOM. By creating an access control list for the SAFECOM object file, a security administrator can restrict EXECUTE authority for the SAFECOM program to a few users or to a single user.

The Safeguard software also limits who can execute certain SAFECOM commands. For example, some restrictions are placed on the ADD command, which creates a

Safeguard record for an object or user. In general, the SAFECOM commands that manage an existing Safeguard record are restricted to the user who owns the record and to that user’s group manager. However, the record can also be managed by any user who has been granted OWNER authority on the object’s access control list. This includes both the primary owner and any secondary owners.

Table 1-1 lists the SAFECOM commands according to command stem and object type and identifies the users who can execute each. This table assumes that you have not created OBJECTTYPE protection records, which specify who can use the ADD command for a particular object type.

The table does not list the local super ID because this user can execute any SAFECOM command for any user or object unless the local super ID is specifically denied those privileges on an access control list.

Also, the table does not list the audit service commands and TERMINAL, EVENT-EXIT-PROCESS, and SAFEGUARD commands, which can be used only by members of the security groups, as discussed in Section 13, Security Group Commands.

Table 1-1. Who Can Use SAFECOM Commands

Command Prefix	Who Can Use The Command
ADD USER	Group manager (*n*,255)
ADD ALIAS	Users with authority to ADD USER plus authority to ALTER USER of the underlying user associated with the alias
ADD GROUP	Local super-group user
ADD DISKFILE	Local file owner, primary owner’s group manager
ADD DISKFILE-PATTERN	Any local user
ADD VOLUME	Local super-group user
ADD SUBVOLUME	Any local user
ADD DEVICE	Local super-group user
ADD SUBDEVICE	Local super-group user
ADD PROCESS	Any local user
ADD SUBPROCESS	Any local user
ADD OBJECTTYPE	Local super-group user
ADD SECURITY-GROUP	Local super-group user
ALTER	Primary and secondary record owners, primary owner’s group manager
DELETE	Primary and secondary record owners, primary owner’s group manager
FREEZE	Primary and secondary record owners, primary owner’s group manager
INFO GROUP	The user, primary and secondary record owners, primary owner’s group manager
INFO USER	The user, primary and secondary record owners, primary owner’s group manager
INFO non-USER	Any user {Other than INFO GROUP}
RESET	Any user
SET USER/ALIAS LIKE	The user, primary and secondary record owners, primary owner’s group manager
SET	Any user
SHOW	Any user
SYNC VOLUME	Any local user
THAW	Primary and secondary record owners, primary owner’s group manager

Note. To alter the behavior of ADD commands, use the OBJECTTYPE protection records. For more information, see Section 12, OBJECTTYPE Security Commands.

Note. The Command Prefix, INFO GROUP, is supported only on systems running H06.09 and later H-series RVUs.

Abbreviating SAFECOM Commands

You can abbreviate any SAFECOM reserved words, including commands, attributes, and keywords. In most instances, you can abbreviate a reserved word to its first three characters although you can use more than three characters for clarity. However, you cannot use fewer than three characters.

Some abbreviations must be longer than three characters so that the Safeguard software can distinguish between similar reserved words, such as SUBVOLUME and SUBPROCESS.

Hyphenated reserved words require at least three characters for each component, and you must include the hyphens.

To illustrate the use of abbreviations, consider this command:

```
ALTER DISKFILE myfile, PROGID on
```

The same command using abbreviated reserved words follows:

```
ALT DISK myfile, PRO on
```

These three pairs of reserved words represent a special case in which you can use the same three-character abbreviation for either reserved word:

Reserved Words		**Abbreviation**
ALL	ALLOCATE	ALL
NAME	NAMED	NAM
REMOTE	REMOTEPASSWORDS	REM

In addition to command abbreviations, SAFECOM allows you to enter certain combinations of audit attributes in shorthand form. For more information, see the *Safeguard Audit Service Manual*.

The Super ID

The local super ID (user ID 255,255) is the group manager for the super group and initially has the user name SUPER.SUPER. By default, the super ID can execute any command without restriction. However, for day-to-day system-management operations, the super ID acts only in emergency situations.

The super ID is needed to:

- Install an operating system using a site update tape (SUT).
- Perform a systemwide backup.
- License a program object file for use by other users (FUP LICENSE command).

- Revoke the license of a licensed object file (FUP REVOKE command).
- Add the first member of a new group or add a group manager (ADDUSER program or SAFECOM ADD USER command) unless specified by the OBJECTTYPE USER.

You can restrict the authority of the super ID in several ways. For example, you can use the FREEZE USER command to freeze the super ID except for emergency situations. Then use the THAW USER command to thaw the super ID as required.

You can deny the super ID the ability to add Safeguard protection records for users or system objects. For example, to deny the super ID the ability to add disk volume protection records, create an OBJECTTYPE VOLUME protection record and specifically deny the super ID all authorities on the access control list.

Similarly, you can control the ability of the super ID to access any system object or to manage any object’s protection record. To do so, create a Safeguard protection record for the object and specifically deny the super ID access authority. The super ID is implicitly granted all access authorities unless you use the DENY clause to deny those authorities. The DENY clause takes precedence over access authorities implicitly or explicitly granted to any user.

For additional information about the super ID, refer to the *Security Management Guide*.

2

Common SAFECOM Language Elements

Many syntax elements in the SAFECOM command language are common to several SAFECOM commands. This simplifies learning and using the language. For example, the language elements that identify users are used in the user security commands and as components of object access lists.

This section describes the syntax for these SAFECOM language elements:

- Disk file names
- Disk volume and subvolume names
- Device and subdevice names
- Process and subprocess names
- User IDs and user names
- User aliases
- User sets
- User lists
- Group names

If you are familiar with HP command languages, you need not read this section in detail. The syntax for the SAFECOM language elements is the same as for other HP command languages. SAFECOM allows the extended use of wild-card characters in object names and user names.

Wild-Card Characters

In most SAFECOM commands, you can use wild-card characters to match characters in an object name. In certain instances, you can also specify wild-card characters in user names. The following wild-card characters are supported:

*   An asterisk (*) matches any number of characters (zero, one, or more).

?   A question mark (?) matches a single character.

The following examples illustrate the use of wild-card characters in SAFECOM commands.

To illustrate the difference between the asterisk and question mark wild cards, consider these commands. The first command (using an asterisk in the file name) displays the attributes of all disk files in the current subvolume whose names begin with the letters *ACCT*. The second command (using a question mark in the file name) displays the attributes of all disk files whose names are five characters long and whose first four characters are *ACCT*.

```
=INFO DISKFILE acct*
```

```
=INFO DISKFILE acct?
```

Similarly, the following command displays the attributes of all disk files in the current subvolume whose names begin with the letter *C* and end with the letter *T*:

```
=INFO DISKFILE c*t
```

This command displays the attributes of all disk files in the current subvolume whose names are three characters long, begin with the letter *C*, and end with the letter *T*:

```
=INFO DISKFILE c?t
```

Wild-card characters make it easy to execute commands on sets of objects with similar names. Typical uses of wild-card characters include adding disk files with similar names to the Safeguard database or displaying information about files with similar names.

Consider the following important points about the use of wild-card characters:

- You cannot use wild cards in SECURITY-GROUP and OBJECTTYPE commands.
- You can use some wild cards in ADD commands:
    - You can use wild cards in ADD commands for disk files, volumes, and subvolumes. An ADD command for one of these types of objects affects only objects that already exist. For example, the following command protects only files that currently exist on volume $VOL1 and subvolume DATA.

      ```
      =ADD DISKFILE $VOL1.DATA.*
      ```
    - You cannot use wild cards in ADD commands for users, devices, subdevices, processes, subprocesses, audit pools, and terminal definitions.
- You can use wild cards in user names or object names with any ALTER, INFO, DELETE, FREEZE, or THAW command. For example:

  ```
  =FREEZE USER op?.user*
  =ALTER DISKFILE $sys*.??rx.*
  =INFO DEVICE $lp*
  ```
- You cannot use wild cards in SET, RESET, and SHOW commands, or in LIKE clauses.

  Wild cards are not meaningful in these commands.
- You cannot use wild cards in WHERE GROUP and WHERE PRIMARY-GROUP clauses or in WHERE clauses for USER and ALIAS commands.

- You must begin a device name with $ and a subdevice name with # even when you use wild cards. For example:

  ```
  =DELETE DEVICE $*
  =INFO TERM $c0.#*
  ```

- Do not mix wild cards with characters in user names when you specify an access control list. For example, the name PROG*.DON is invalid in specifying an access control list entry.

  Wild cards can only be used in only two instances when you specify user names for an access control list. The only valid forms are *`group-name.*`* and *`*.*`*. The form *`group-name.*`* indicates all members of the named group. The form *`*.*`* indicates all members of all groups on the local system.

- Do not mix wild cards with numbers in group numbers and member numbers under any circumstances. For example, the user ID 2*,?0 is not valid.

  You can use wild cards in only two instances when you specify a user ID. The only valid forms *`group-number,*`* and *`*,*.`* All other forms, such as *`*,124`*, are invalid. The form *`group-number,*`* indicates all members of the specified group. The form *`*,*`* indicates all members of all groups on the local system.

- Diskfile patterns reduce administrative burden by supplying one pattern that can match many subvolumes or filenames. For more information, see the *Safeguard User's Guide.*

Object Names

You can use the Safeguard software to control user access to these types of system objects:

- Disk files
- Disk volumes
- Disk subvolumes
- Devices
- Subdevices
- Process names
- Subprocess names

When you enter an ADD command to place a system object under Safeguard control, you use the name of the object to identify the object and to name the Safeguard authorization record. Similarly, once an object is under Safeguard control, you use the object name to identify the object in SAFECOM commands to manage the Safeguard access controls.

You can use either fully qualified names or partially qualified names to identify a system object. A fully qualified file name consists of system, volume, subvolume, and file names. An example is \MYSYS.$SYSTEM.SYSTEM.SAFECOM. A partially qualified file name omits one or more parts of the name. SAFECOM uses the current default system, volume, and subvolume names to expand the name to a fully qualified name, possibly altered by SAFECOM SYSTEM and VOLUME commands.

Specifying Disk-File Names

You can identify a disk file with either a fully or a partially qualified disk file name. To specify more than one disk file in a command, you can use wild cards in a disk file name, or you can use a file name list.

Fully Qualified Disk-File Names

A fully qualified disk-file name includes the volume, subvolume, and file name, and can include the system name for systems in a network.

A pattern is a template that represents a fully qualified file name. Wildcards are not allowed in the volume dimension of a pattern. A one-dimensional search is limited to the volume dimension. For example, $DATA*.B*.C* is a search for the pattern “B*.C*” on all volumes that match “$DATA*”. For more information, see the *Safeguard User’s Guide*.

```
[\system-name.]$volume.subvolume.disk-filename
```

`\system-name`

is the name of a system in a network of HP computer systems. This name can be one to seven alphanumeric characters long, the first of which must be alphabetic.

Note. For disk files residing on a system that is not named, you must omit`\system-name`. See ALLPROCESSORS PARAGRAPH in the *System Generation Manual*.

`$volume`

is the name of a disk volume. This name can be one to seven alphanumeric characters long, the first of which must be alphabetic.

`subvolume`

is the name of a subvolume residing on a disk volume. This name can be one to eight alphanumeric characters long, the first of which must be alphabetic.

`disk-filename`

is the name of a disk file residing within a subvolume. This name can be one to eight alphanumeric characters long, the first of which must be alphabetic.

Examples

```
=INFO DISKFILE \tops.$data.stats.rpt1
=INFO DISKFILE \sfo.$users1.nelson.rpt*
=ALTER DISKFILE \sys*.$ops?.*.quarter2, OWNER 86,2
=ADD DISKFILE \pts.$*.stats.*, OWNER admin.bob
=INFO /OUT safelist/ DISKFILE \*.$*.*.*
```

Partially Qualified Disk-File Names

In a partially qualified disk-file name, one or more of the system, volume, and subvolume names is omitted. When you specify a partially qualified disk-file name, SAFECOM uses the current default system, volume, and subvolume names to create a fully qualified disk-file name.

```
[\system-name.][$volume.][subvolume.]disk-filename
```

\system-name

is a valid system name. If omitted, the current default system name is used.

$*volume*

is a disk volume name. If omitted, the current default disk volume name is used.

subvolume

is a disk subvolume name. If omitted, the current default subvolume name is used.

disk-filename

is the name of a disk file.

When you run SAFECOM, your command interpreter passes the current default volume and subvolume names to SAFECOM. SAFECOM uses these defaults as your initial session defaults. During your SAFECOM session, you can change the default volume and subvolume through the VOLUME command.

However, your command interpreter does not pass your current default system name to SAFECOM. The initial default system name for SAFECOM is the name of the system on which SAFECOM is running. To change the default system name during a SAFECOM session, use the SYSTEM command. For more information, see Section 4, SAFECOM Session-Control Commands. This example shows a partially qualified disk-file name:

```
=SYSTEM \london
=VOLUME $data.sales
=INFO DISKFILE report1
```

SAFECOM then uses the default system, volume, and subvolume names to create this fully qualified disk-file name:

```
\LONDON.$DATA.SALES.REPORT1
```

In this example, SAFECOM uses only the default system and subvolume names to create this partially qualified disk-file name:

```
=SYSTEM \london
=VOLUME $data.sales
=INFO DISKFILE $books.report1
```

A fully qualified disk-file name follows:

```
\LONDON.$BOOKS.SALES.REPORT1
```

Examples

```
=FREEZE DISKFILE report4
=INFO DISKFILE $data.stats.report4
=INFO DISKFILE stats.report4
=THAW DISKFILE \stl.$data.report4
=ADD DISKFILE $dat*.*.*, LIKE $data.master.index
=ALTER DISKFILE *.rep*, OWNER 14,4
```

File-Name Lists

A file name list is a list of fully qualified or partially qualified file names. A file-name list specifies a group of disk files on which the same operation is to be performed.

```
( disk-filename [ , disk-filename] ... )
```

disk-filename

is a fully qualified disk-file name or a partially qualified disk-file name.

For this example, this INFO command follows the file list:

```
=SYSTEM \la
=VOLUME $data.sales
=INFO DISKFILE (report1, report2, $system.query1)
```

It produces a report on these disk files:

```
\LA.$DATA.SALES.REPORT1
\LA.$DATA.SALES.REPORT2
\LA.$SYSTEM.SALES.QUERY1
```

Examples

```
=FREEZE DISKFILE (rpt1, rpt3, stats)
=ADD DISKFILE (rpt1, $prog.*.?stat*) LIKE report
=ADD DISKFILE ($data.sales.*, custom.*)
=INFO DISKFILE (secur??, data4, audit.a*)
```

Patterns

Diskfile patterns reduce administrative burden by supplying one pattern that can match many subvolumes or filenames. For more information, see the *Safeguard User's Guide.*

Specifying Disk Volume Names

You can identify a disk volume with either a fully or a partially qualified volume name. To specify more than one volume in a command, you can use wild cards in a volume name, or you can use a volume name list.

Fully Qualified Volume Names

A fully qualified volume name includes both a system name and a volume name.

```
\system-name.$volume
```

\system-name

is a system name as described in Fully Qualified Disk-File Names on page 2-4.

$*volume*

is a disk volume name as described in Fully Qualified Disk-File Names on page 2-4.

Examples

```
=ALTER VOLUME \tops.$data, LIKE \tops.$user2,
=ADD VOLUME \tops.$dat*
=INFO VOLUME \tops.$*
=INFO VOLUME \*.$SYS??
```

Partially Qualified Volume Names

A partially qualified volume name is a volume name without a system name.

```
$volume
```

$*volume*

is expanded to \system-name.$*volume*.

\system-name

is the current default system name.

Examples

```
=FREEZE VOLUME $data
=ADD VOLUME $mail*, LIKE $data
=INFO VOLUME $*
=THAW VOLUME $?com
```

Volume Name Lists

A volume name list is a list of fully qualified or partially qualified volume names. A volume name list specifies a group of disk volumes on which the same operation is to be performed.

```
( volume [ , volume ] ... )
```

volume

is either a fully or a partially qualified volume name.

Examples

```
=ALTER VOLUME ($mail, $sail, $trail), OWNER ops.sue
=ADD VOLUME ($mail*, \stl.$data, \sfo.$data)
=INFO VOLUME (\big?.$*, $*)
```

Specifying Subvolume Names

You can identify a subvolume with either a fully or partially qualified subvolume name. To specify more than one subvolume in a command, you can use wild cards in a subvolume name, or you can use a subvolume name list.

Fully Qualified Subvolume Names

A fully qualified subvolume name includes a system name, a volume name, and a subvolume name.

```
\system-name.$volume.subvolume
```

\system-name

is a system name as described in Fully Qualified Disk-File Names on page 2-4.

$*volume*

is a disk volume name as described in Fully Qualified Disk-File Names on page 2-4.

subvolume

is a subvolume name as described in Fully Qualified Disk-File Names on page 2-4.

Examples

```
=DELETE SUBVOLUME \tops.$data.jones
=ADD SUBVOLUME \tops.$dat*.*
=ADD SUBVOLUME \tops.$*.valdez
=INFO SUBVOLUME \*.$data.*
```

Partially Qualified Subvolume Names

A partially qualified subvolume name is a subvolume name with the system name or the volume name (or both) omitted.

```
[\system.][$volume.]subvolume
```

\system

is a system name. If omitted, the current default system name is used.

$*volume*

is a disk volume name. If omitted, the current default volume name is used.

subvolume

is a subvolume name.

Examples

```
=FREEZE SUBVOLUME data
=THAW SUBVOLUME $users.data
=INFO SUBVOLUME \sfo.data
=ADD SUBVOLUME $use*.dat*
=ALTER SUBVOLUME ?ray*, LIKE $user.data2
```

Subvolume Name Lists

A subvolume name list is a list of fully qualified or partially qualified subvolume names. A subvolume name list specifies a group of subvolumes on which the same operation is to be performed.

```
( subvol [ , subvol ] ... )
```

subvol

is either a fully or a partially qualified subvolume name or a subvolume name list.

Examples

```
=ALTER SUBVOLUME (data, report), ACCESS 14,* (r,w)
=ADD SUBVOLUME ($users.data, rpt*)
```

```
=THAW SUBVOLUME (\sfo.$users.data, $*.*)
=INFO SUBVOLUME (\*.$users.*, jones)
```

Specifying Device Names

You can identify a device with either a fully or a partially qualified device name. To specify more than one device in a command, you can use wild cards in a device name, or you can use a device name list. However, you cannot use wild cards to specify a device name in an ADD command.

Fully Qualified Device Names

A fully qualified device name includes the system and device name.

```
[\system-name.]$device-name
```

`\`*`system-name`*

is a system name as described in Fully Qualified Disk-File Names on page 2-4. For devices residing on a system that is not part of a network, omit `\`*`system-name`* from the device name.

`$`*`device-name`*

is the name of a logical device. This name can be one to seven alphanumeric characters long, the first of which must be alphabetic.

Examples

```
=ADD DEVICE \apex.$lp1
=INFO DEVICE \apex.$lp*
=FREEZE DEVICE \tops.$lazer*
```

Partially Qualified Device Names

For devices that can be accessed over a network, a partially qualified device name is a device name without a system name. SAFECOM expands the partially qualified name by adding the current default system name.

```
$device-name
```

`$`*`device-name`*

is expanded to `\`*`system-name.`*`$`*`device-name`*.

`\`*`system-name`*

is the current default system name.

Examples

```
=ADD DEVICE $lp2 LIKE $lp1
=INFO DEVICE $lp*
=FREEZE DEVICE $lazer*
```

Device Name Lists

A device name list is a list of fully qualified or partially qualified device names. A device name list specifies a group of devices on which the same operation is to be performed.

```
( device-name [ , device-name ] ... )
```

device-name

is either a fully or a partially qualified device name.

Examples

```
=ADD DEVICE ($lp2, lp3) LIKE $lp1
=ALTER DEVICE ($lp*, $ta???, \apex.$lp2) OWNER ops.bill
=FREEZE DEVICE $lazer*
```

Specifying Subdevice Names

You can identify a subdevice with either a fully or a partially qualified subdevice name. To specify more than one subdevice in a command, you can use wild cards in a subdevice name, or you can use a subdevice name list. However, you cannot use wild cards to specify a subdevice name in an ADD command.

Fully Qualified Subdevice Names

A fully qualified subdevice name includes a system name, a device name, and a subdevice name.

```
\system-name.$device.#subdevice
```

\system-name

is a system name as described in Fully Qualified Disk-File Names on page 2-4.

$*device*

is a disk device name as described in Fully Qualified Disk-File Names on page 2-4.

#*subdevice*

is a subdevice name. The name can be one to seven alphanumeric characters, the first of which must be alphabetic.

Examples

```
=ADD SUBDEVICE \apex.$tc02.#p04
=INFO SUBDEVICE \apex.$tc12*.#t04
=FREEZE SUBDEVICE \tops.$cl4.#lazer*
```

Partially Qualified Subdevice Names

For subdevices that can be accessed over a network, a partially qualified subdevice name is a device name followed by a subdevice name. SAFECOM expands the partially qualified name by adding the current default system name.

```
$device.#subdevice
```

$*device.#subdevice*

is expanded to:

\system-name.$*device.#subdevice*

\system-name

is the current default system name.

Examples

```
=ADD SUBDEVICE $tc02.#p04
=INFO SUBDEVICE $tc12*.#t04
=FREEZE SUBDEVICE $cl4.#lazer*
```

Subdevice Name Lists

A subdevice name list is a list of fully qualified or partially qualified subdevice names. A subdevice name list specifies a group of subdevices on which the same operation is to be performed.

```
( subdevice-name [ , subdevice-name ] ... )
```

subdevice-name

is either a fully or a partially qualified subdevice name.

Examples

```
=ADD SUBDEVICE ($tc02.#pt04, $tc03.#pt01)
=INFO SUBDEVICE ($tc12*.#t*, $c*.#pt*)
=FREEZE SUBDEVICE ($cl4.#lazer*, $c12.#pt*)
```

Specifying Process Names

You can identify a named process with either a fully or a partially qualified process name. To specify more than one named process in a command, you can use wild cards in a process name, or you can use a process name list. However, you cannot use wild cards to specify a process name in an ADD command.

Fully Qualified Process Names

A fully qualified process name includes both the system name and process name.

```
[\system-name.]$process-name
```

\system-name

is a system name as described in Fully Qualified Disk-File Names on page 2-4. For processes running on a system that is not part of a network, omit *\system-name* from the process name.

$process-name

is the name defined for a process when that process is run. This name can be one to five alphanumeric characters long, the first of which must be alphabetic.

Examples

```
=ADD PROCESS \apex.$pc06
=INFO PROCESS \ajax.$rpt*
=FREEZE PROCESS \fred.$clup
```

Partially Qualified Process Names

Process names can be partially qualified. As with device names, the only part of a process name you can omit is the system name. When you enter a process name without a system name, SAFECOM expands the partially qualified name by adding the current default system name.

```
$process-name
```

$process-name

is expanded to *\system-name.$process-name.*

\system-name

is the current default system name.

Examples

```
=ADD PROCESS $spell LIKE $cedit
=INFO PROCESS $loc*
=THAW PROCESS $limit
```

Process Name Lists

A process name list is a list of fully qualified or partially qualified process names. A process name list specifies a group of processes on which the same operation is to be performed.

```
( process-name [ , process-name ] ... )
```

process-name

is either a fully or a partially qualified process name.

Examples

```
=ADD PROCESS ($trump, $frump)
=ALTER PROCESS ($pri*, $tc34) LIKE $tc01
=DELETE PROCESS ($tc2, $?ax, $line*)
```

Specifying Subprocess Names

You can identify a subprocess with either a fully or partially qualified subprocess name. To specify more than one subprocess in a command, you can use wild cards in a subprocess name, or you can use a subprocess name list. However, you cannot use wild cards to specify a subprocess name in an ADD command.

Fully Qualified Subprocess Names

A fully qualified subprocess name includes a system name, a process name, and a subprocess name.

```
\system-name.$process.#subprocess
```

\system-name

is a system name as described in Fully Qualified Disk-File Names on page 2-4.

$process

is a process name as described in Fully Qualified Disk-File Names on page 2-4.

#subprocess

is a subprocess name. The name can be one to seven alphanumeric characters, the first of which must be alphabetic.

Examples

```
=ADD SUBPROCESS \argon.$pc12.#tl06
=INFO SUBPROCESS \ajax.$rpt*.#prt
=FREEZE SUBPROCESS \fred.$ted*.#clup
```

Partially Qualified Subprocess Names

Subprocess names can be partially qualified. As with device names, the only part of a subprocess name you can omit is the system name. When you enter a subprocess name without a system name, SAFECOM expands the partially qualified name by adding the current default system name.

```
$process.#subprocess
```

$*process.#subprocess*

is expanded to:

\system-name.$*process.#subprocess*

\system-name

is the current default system name.

Examples

```
=ADD SUBPROCESS $spell.#us002 LIKE $spell.#us001
=FREEZE SUBPROCESS $*.#pr*
=THAW SUBPROCESS $alto.#pr02
```

Subprocess Name Lists

A subprocess name list is a list of fully qualified or partially qualified subprocess names. A subprocess name list specifies a group of subprocesses on which the same operation is to be performed.

```
( subprocess-name [ , subprocess-name ] ... )
```

subprocess-name

is either a fully or a partially qualified subprocess name.

Examples

```
=ADD SUBPROCESS (\argon.$pc12.#tl06, $rpt.#us05)
=INFO SUBPROCESS ($*.pri*, \ajax.$rpt*.#prt)
=DELETE SUBPROCESS ($fred.#ted, $fred.#red)
```

Identifying System Users

The system user community supported by the operating system is organized into 256 groups, each of which can include 256 individual users for purposes of administration. (Groups can include more than 256 users for file-sharing purposes.) Each system user added to a NonStop system is assigned a user ID and a user name.

A user ID is a pair of numbers separated by a comma:

group-num , member-num

A user name is a pair of names separated by a period:

group-name.member-name

In most instances, you can use either a user ID or a user name to identify a user in a SAFECOM command.

You can also identify users by aliases, which are described in User Aliases on page 2-19.

User IDs

A user ID consists of an administrative group number and a member number, separated by a comma.

```
group-num , member-num
```

group-num

is a number in the range 0 through 255. Two group numbers and their associated group names are defined by default:

group-name	group-num
SUPER	255
NULL	0

member-num

is a number in the range 0 through 255. The *member-num* value 255 is reserved for group managers. The manager of the super group (255,255) is the local super ID. Each member number must be unique within its administrative group.

A user’s administrative group is identified by the *member-num* portion of the user ID.

An example of a user ID is 12,104. The user’s administrative group is group number 12, and the user is member number 104 within the group.

User Names

A user name consists of a group name and a member name, separated by a period.

```
group-name.member-name
```

group-name

is the name of an administrative group. It is one to eight alphanumeric characters long, the first of which must be alphabetic. In most SAFECOM commands, an administrative group name is case-insensitive. The alphabetic characters are assumed to be uppercase. However, group names specified in GROUP commands are case-sensitive, and the alphabetic characters in an administrative group name must be entered as uppercase characters in these commands.

member-name

is the name of a user. It is one to eight alphanumeric characters long, the first of which must be alphabetic. Each member name must be unique within its administrative group.

A user's administrative group is identified by the *group-name* portion of the specific user's user name.

An example of a user name is PRS.HARRY. This user's administrative group is named PRS, and the user has the member name HARRY.

Network Users

A network user is a system user who has been granted the authority to access objects on a remote system.

Allowing a network user to access objects on your system requires cooperation between the system managers (or security administrators) on your system and on the network user's node.

First, your system manager or group manager must add the network user as a local user on your system (with the same user ID and user name as on the other system). Then, on your system, the manager (or any user with the proper authority) must give the network user remote passwords for the two systems. And finally, on the network user's local system, the system manager on the network user's node must give the network user matching remote passwords for the two systems. For more information, see the *Safeguard Administrator's Manual*, the *TACL Reference Manual*, or the *Security Management Guide*.

In a Safeguard access control list, network users are identified by either the network form of their user ID or the network form of their user name. (The network form of a user name is not valid for user authentication in a LOGON command or in a call to the USER_AUTHENTICATE_ procedure or the VERIFYUSER procedure.)

The network form of a user name and user ID can have the following form:

```
NETWORK FORM OF USER ID:

  \node-spec.group-num , member-num

NETWORK FORM OF USER NAME:

  \node-spec.group-name.member-name
```

Note. You may only use *node-spec* in the ACCESS clause of SAFECOM.

For example, suppose a network user has a user ID of 3,3, a user name of SALES.BOB, and is on the NYC node. This network user can be identified on an access list with either of these forms:

```
\NYC.3,3

\NYC.SALES.BOB
```

In summary, the network form of a user ID or user name implies two things:

- This user is known as a local user on more than one system.
- This user has matching remote passwords for both systems set up on your system and on the user's local system.

User Sets

To specify all the members of a user group in a SAFECOM command, use a user set.

```
USER NAME FORMS:

  {            group-name.* }
  {                    *.* }
  { \node-spec.group-name.* }
  {           \node-spec.*.* }

USER ID FORMS:

  {             group-num,* }
  {                     *,* }
  {  \node-spec.group-num,* }
  {          \node-spec.*,* }
```

group-name.*
group-num,*

The Safeguard software interprets *group-name*.* and *group-num*,* differently depending on the context.

In USER commands, these forms specify all the local users whose administrative group is identified by *group-name* or *group-num*.

In access control list entries, these forms specify all the local users who are members of the group identified by *group-name* or *group-num*. This includes users who have been specified as members of the group with the MEMBER clause in an ADD or ALTER GROUP command.

`*.* and *,*`

each specifies all the local users defined for your system.

\node-spec.*group-name*.*
\node-spec.*group-num*,*

each specifies all the users (local users and network users) who are members of a particular group.

\node-spec.*.*
\node-spec.*,*

both specify all the users (both local and network) who are defined for your system.

User-Set Lists

A user-set list provides a shortened way to specify two or more users and user groups.

```
( [\node-spec.]user [ , [\node-spec.]user] ... )
```

The value of *user* `can be any of these:`

group-num, *member-num*
group-name.*member-name*
group-num,*
group-name.*
.
,

SAFECOM also supports wild-card characters within group names, and user names except in ADD commands or when those names designate access control list entries.

This user-set list specifies all the members of the local group OPS, the local user PUBS.SUE, and all members of local groups whose names begin with the letters SY:

```
=ALTER USER (ops.*, pubs.sue, sy*.*), OWNER 12,2
```

User Aliases

A user alias is an alternate name assigned to a specific user ID or user name. The user can log on to the system using the alias. When the user is logged on as an alias, the Safeguard software bases access control decisions on the underlying user ID.

```
alias
```

alias

is a case-sensitive text string of up to 32 alphanumeric and special characters. The first character of an alias name must be alphabetic. The following special characters are allowed in an alias name: period (.), hyphen (-), and underscore (_).

Identifying User Groups

User groups are created implicitly with the ADD USER command and explicitly with the ADD GROUP command. Groups created with the ADD USER command are always administrative groups. Groups created with the ADD GROUP command can be either file-sharing groups or administrative groups.

Group Names

A user group name has the following form:

```
group-name
```

group-name

is a case-sensitive text string that can be up to 32 alphanumeric characters in length. In addition to alphabetic and numeric characters, the characters period (.), hyphen (-), and underscore(_) are permitted within the text string. The first character of a group name must be alphabetic or numeric.

Administrative group names have more restricted syntactical requirements. An administrative group name is one to eight alphanumeric characters long, the first of which must be alphabetic. In most SAFECOM commands, an administrative group name is case-insensitive. The alphabetic characters are assumed to be uppercase. However, group names specified in GROUP commands are case-sensitive, and the alphabetic characters in an administrative group name must be entered as uppercase characters in these commands.

Group Numbers

A user group number has the following form:

```
group-number
```

group-number

is a number in the range 0 through 65535. It must be unique within the local system.

Administrative group numbers must be in the range 0 through 255.

3

The Command to Run SAFECOM

This section contains the syntax description of the command to run SAFECOM, followed by examples that show each of the three modes of program operation. For more examples of running SAFECOM, see the *Safeguard User's Guide*. For instructions on starting the Safeguard software, see the *Safeguard Administrator's Manual*.

Modes of Program Operation

To run the SAFECOM program, you must have the necessary EXECUTE authority for the SAFECOM program object file ($SYSTEM.SYS*nn*.SAFECOM). Once authorized, you can run SAFECOM by entering a run command at the command-interpreter prompt.

You can run SAFECOM in any of three modes:

- Execute-and-quit-mode: SAFECOM executes specified commands and then returns control of your terminal to the command interpreter.
- Interactive session mode: SAFECOM prompts you for commands and displays the results of the commands on your terminal.
- Batch mode: SAFECOM accepts commands from an IN file (an EDIT-format file that contains SAFECOM commands) that you can specify in the command to run SAFECOM, or from an OBEY command file. (For more information, see Section 4, SAFECOM Session-Control Commands.)

Command Syntax

From the command interpreter, you start a SAFECOM process by entering a SAFECOM run command. Each command to run SAFECOM starts a separate SAFECOM process. The SAFECOM program object file resides in the current system subvolume ($SYSTEM.SYS*nn*, where *nn* is a 2-digit octal integer).

```
SAFECOM [ / run-opt [ , run-opt ] ... / ]

   [ cmd [ ; cmd ] ... ]
```

SAFECOM

is an implicit RUN command instructing your command interpreter to run the SAFECOM program object file. (If you need to know the subvolume on which the SAFECOM object file is installed, see your system manager.)

If you enter SAFECOM with no *cmd*, you start an interactive SAFECOM process that controls the terminal and displays the SAFECOM command prompt, an equal sign (=). You can enter a SAFECOM command at the prompt. To exit SAFECOM, enter the EXIT command. (For more information, see Section 4, SAFECOM Session-Control Commands.)

`run-opt`

is any run option for the RUN command of the command interpreter. (For a complete list of run options, see the description of the RUN[D] command in the *TACL Reference Manual*.)

Run options frequently used with SAFECOM include:

```
CPU cpu
IN filename
NAME [$process-name]
NOWAIT
OUT [listfile]
PRI priority
```

`CPU` *`cpu`*

is the number of the CPU in which this SAFECOM process is to run. If you omit this option, the command interpreter starts SAFECOM in the same CPU in which the command interpreter is running. (If your installation operates a $CMON process, $CMON can override a CPU specification.)

`IN` *`filename`*

specifies the file that SAFECOM uses as a command input file. If you omit the IN option, SAFECOM uses the IN file currently defined for your command interpreter (usually your home terminal).

You can specify an EDIT file containing SAFECOM commands as an IN file. When you use an EDIT file as the IN file, *`filename`* can be a partially qualified disk file name.

Note. If you include a *`cmd`* in your command to run SAFECOM, SAFECOM will run in execute-and-quit mode, execute the command *`cmd`*, and ignore the IN option, if specified.

If any error occurs while processing the IN file, the processing does not stop and continues till the end of file. Once all the commands are processed, ABEND is called and TACL returns -6 as the MESSAGECODE and a non-zero value as the COMPLETIONCODE.

Note. Supported only on systems running H06.28 and later H-series RVUs and J06.17 and later J-series RVUs.

`NAME [$`*`process-name`*`]`

assigns a process file name to your SAFECOM process.

If you give your SAFECOM process a name, that name appears in the destination-control table (DCT). You can use the process name in commands that manage or monitor processes (such as the command-interpreter STATUS command).

If you include NAME but omit $*process-name*, the system assigns a name to your SAFECOM process.

If you omit the NAME option, your SAFECOM process runs as an unnamed process. It can be identified only by the system-assigned process number.

`NOWAIT`

instructs your command interpreter to return to your terminal for more commands after starting a SAFECOM process. Typically, you use NOWAIT in the batch mode, in which you specify an EDIT command file as the IN file for SAFECOM, or in the execute-and-quit mode. The NOWAIT option means that SAFECOM runs in the background, and you can use your terminal for other work.

If you omit NOWAIT, your command interpreter starts SAFECOM and then pauses while SAFECOM runs. Typically, you omit NOWAIT when you are starting an interactive SAFECOM session. When you run SAFECOM without the NOWAIT option, SAFECOM accepts input commands from your terminal and displays its error messages and output reports on your terminal. When you exit SAFECOM, the command interpreter regains control of your terminal and displays its prompt.

`OUT [`*`listfile`*`]`

specifies a file that SAFECOM uses as an output file. SAFECOM writes all input commands and their responses to those commands to its current output file. For *listfile*, specify any file name. SAFECOM appends its output to *listfile*. If *listfile* does not exist, SAFECOM creates an EDIT-format file and writes its output to that file.

If you specify an IN *filename* and include OUT with no *listfile*, SAFECOM produces no output text.

If you omit the OUT option, SAFECOM uses the OUT file currently defined for your command interpreter (usually your home terminal).

If the OUT file is specified and an error is encountered in processing the command, SAFECOM will ABEND on exit and TACL returns -6 as the MESSAGECODE and a non-zero value as the COMPLETIONCODE.

For example,

```
SAFECOM/OUT <filename>/
=info badcommand
=exit
ABENDED: 0,29


SAFECOM /OUT <filename>/info badcommand
ABENDED: 0,28
```

Note. The ABEND on exit feature is supported only on systems running H06.28 and later H-series RVUs and J06.17 and later J-series RVUs.

PRI *priority*

assigns an execution priority for the SAFECOM process. For *priority*, specify an integer in the range 1 through 199. (Processes with higher priorities generally run faster than processes with lower priorities.)

If you omit the PRI option, the command interpreter starts SAFECOM with a *priority* that is one less than the priority of your command interpreter.

If your installation operates a $CMON process, $CMON can override a PRI specification.

cmd

is any SAFECOM command except the FC, ! , and ? commands. When you include *cmd*, the new SAFECOM process runs in the execute-and-quit mode.

Note. If you include a *cmd* in your command to run SAFECOM, SAFECOM will run in execute-and-quit mode, execute the command *cmd*, and ignore the IN option, if specified.

Considerations

- Running SAFECOM without a local SMP

You can run SAFECOM even if the Safeguard security-manager process ($ZSMP) is not currently running on your system. However, any attempt to access either your local subject or object data base results in the following warning message:

```
* ERROR *   CANNOT OPEN SMP : FILE ERROR = 014
```

You can still use SAFECOM to access the object databases on a remote system that has the Safeguard software installed if the SMP on that system is running and you are a network user with access to that system.

Examples

The following examples illustrate various ways to run SAFECOM:

1. This example runs SAFECOM in execute-and-quit mode:

```
SAFECOM INFO DISKFILE $data.sales.report1
```

The report displays information about a disk file:

```
                  LAST-MODIFIED     OWNER    STATUS     WARNING-MODE
$DATA.SALES
 REPORT1         14OCT87, 11:46       8,7    THAWED          OFF

        002,004 DENY R
     \*.008,007      R,W,E,P
        002,*        R
        008,*        R
```

The access-control list for this file shows a network user entry for user 8,7.

2. Start an interactive SAFECOM session:

```
SAFECOM / CPU 4, PRI 143, NAME $scom4 /
```

The screen displays:

```
SAFEGUARD COMMAND INTERPRETER -T9750xnn- (ddmmmyy) SYSTEM \sys
=
```

This command includes three run options that specify that the SAFECOM process run in CPU 4 with a priority of 143 and with the process name $SCOM4.

3. This sample command runs SAFECOM in batch mode:

```
SAFECOM / IN $data.security.setup , NOWAIT/
```

The IN option in this command specifies the disk file $DATA.SECURITY.SETUP as the input file for the SAFECOM process.

4

SAFECOM Session-Control Commands

The SAFECOM session-control commands establish a working environment for your SAFECOM session. For example, the SYSTEM and VOLUME commands establish the default system, volume, and subvolume names that SAFECOM uses to expand partially qualified disk file names. Similarly, the ASSUME command establishes a default object class (such as DISKFILE or USER) so that you can enter object-management commands without specifying the default object class in each command.

Other session-control commands, such as FC and HELP, make SAFECOM easy to use. HELP lets you check the syntax of SAFECOM commands. FC lets you correct typing mistakes or quickly enter a number of similar commands that differ by only a few characters.

Several DISPLAY session-control commands allow you to control the displays produced by INFO commands. For example, with the DISPLAY DETAIL command you can turn on the DETAIL option of the INFO command for an entire SAFECOM session.

This section begins with a summary of the session-control commands. The syntax of each session-control command follows.

Session-Control Command Summary

Table 4-1 on page 4-2 summarizes the SAFECOM session-control commands. The remainder of this section describes these commands in detail.

Session-Control Command Syntax

Table 4-1. Session-Control Command Summary (page 1 of 2)

Command	Function
ASSUME	Establishes a default object class for subsequent object-management commands during the current session.
DISPLAY COMMANDS	Displays the output of an INFO or SHOW command as SAFECOM commands.
DISPLAY DETAIL	Controls the DETAIL option of the INFO command for an entire session.
DISPLAY HEADERS	Controls the display of headings in INFO command reports.
DISPLAY PROMPT	Controls the text displayed in your SAFECOM prompt.
DISPLAY USER	Displays the identities of users as either user IDs or user names.
DISPLAY WARNINGS	Controls the display of warnings about unprotected files.
ENV	Displays the current default values of the environmental parameters and DISPLAY options (SYSTEM, VOLUME, OUT, LOG, ASSUME, and DISPLAY).
EXIT	Ends an interactive SAFECOM session. Control of your terminal returns to your command interpreter. You can also use Ctrl-Y to end an interactive session.
FC	Displays and allows you to edit a previously entered SAFECOM command.
HELP	Displays help screens for SAFECOM commands.
HISTORY	Displays a specified number of your most recently entered SAFECOM commands.
LOG	Specifies a log file to which SAFECOM writes a record of your SAFECOM session.
OBEY	Specifies an input command file (an EDIT file) containing SAFECOM commands.
OUT	Redirects SAFECOM output to a specified file.
SYNTAX	Directs SAFECOM to check command syntax only and not to execute commands.
SYSTEM	Establishes a default system name to be used for file-name expansion.
VOLUME	Establishes a default disk volume name and a default subvolume name to be used for expanding disk file and subvolume names.
?	Displays a specified command that you previously entered during the current session.

Table 4-1. Session-Control Command Summary (page 2 of 2)

Command	Function
!	Displays and executes a specified command that you previously entered during the current session.
-- (two hyphens)	Delimits comments in SAFECOM commands.
& (ampersand)	Indicates that the command is continued on the next line.

The rest of this section contains individual syntax descriptions. Commands appear in alphabetic order, and most descriptions contain these elements:

- A summary of the function performed by the command
- Descriptions of the command parameters and variables
- Special considerations for the use of the command
- Examples of command usage

ASSUME Command

ASSUME establishes a default object class so that the object class need not be entered in each command that requires it. The object class specified in the ASSUME command is used for subsequent RESET, SET, SHOW, ADD, INFO, ALTER, FREEZE, THAW, and DELETE commands during the current session. You can assume a new default object class at any time during the session.

After you specify a default object class, you can enter an object-management command for the default object class without specifying the object class again.

```
ASSUME [ object-type ]
```

`ASSUME`

entered with no *`object-type`*, establishes no default object class. If you omit object class, you must specify an object class in all subsequent object-management commands.

`object-type`

is the default object class assumed for any subsequent user-management or object-management commands that do not include either USER or an object class. For *`object-type`*, specify one of these:

```
ALIAS
DEVICE
DISKFILE
DISKFILE-PATTERN
EVENT-EXIT-PROCESS
PROCESS
SUBDEVICE
```

```
   SUBPROCESS
   SUBVOLUME
   TERMINAL
   USER
   VOLUME
```

Note. The ASSUME command is not valid for OBJECTTYPE, GROUP, or SECURITY-GROUP.

Example

In this example, ASSUME establishes DISKFILE as the default object class:

```
=ASSUME DISKFILE
=SET ACCESS (sales.*, admin.*) r
=ADD $data.q3.report
```

SAFECOM then executes the SET and ADD commands as though you had entered:

```
=SET DISKFILE ACCESS (sales.*, admin.*) r
=ADD DISKFILE $data.q3.report
```

DISPLAY Command

DISPLAY specifies several different command options that alter the output of the INFO and SHOW commands. You can enter several DISPLAY command options in a single list. This form is convenient if you want to change several default DISPLAY settings at the start of a SAFECOM session. You can also execute the DISPLAY command options individually, as described later.

```
DISPLAY command [ , command ] ...
```

command

is one the following DISPLAY command options:

```
[ AS ] COMMANDS [ ON | OFF ]
DETAIL [ ON | OFF ]
HEADERS [ ON | OFF | ONCE ]
PROMPT ( prompt-item )
     [ ( prompt-item [ ,prompt-item ] )... ]
USER [ AS ] { NAME | NUMBER }
WARNINGS [ ON | OFF ]
```

For detailed descriptions of these options, see the individual DISPLAY options in this section.

Examples

In this example, the DISPLAY command list turns warnings off, displays user identities as names, and causes a single heading to be displayed on INFO reports:

```
DISPLAY WARNINGS OFF, USER AS NAME, HEADERS ONCE
```

DISPLAY AS COMMANDS Option

DISPLAY AS COMMANDS controls whether the output of an INFO or SHOW command is displayed as a report or as a list of SAFECOM commands. Normally, INFO and SHOW commands produce reports. To display the output of INFO and SHOW as commands, use the DISPLAY AS COMMANDS option.

```
DISPLAY [ AS ] COMMANDS [ ON | OFF ]
```

`DISPLAY [ AS ] COMMANDS`

entered without ON or OFF, specifies that the output of INFO and SHOW commands is displayed as equivalent SAFECOM commands. The keyword AS is optional.

`ON`

specifies that the output of INFO and SHOW commands is displayed as equivalent SAFECOM commands.

`OFF`

specifies that the output of INFO and SHOW commands is displayed in report form.

Considerations

- DISPLAY AS COMMANDS does not support the GENERAL, AUDIT, and CI options of the INFO SAFEGUARD command.

Examples

This example illustrates how the INFO command output can be changed with the DISPLAY AS COMMANDS command.

Execute the following INFO command at the start of a SAFECOM session to display the authorization record for the disk file RPT01:

```
=INFO DISKFILE rpt01, DETAIL
```

The following information appears:

```
                   LAST-MODIFIED     OWNER    STATUS     WARNING-MODE
$DATA.SALES
 RPT01            26JUL88, 13:04       2,5    THAWED         OFF

        002,005      R,W,E,P,  O
        002,*        R

  AUDIT-ACCESS-PASS = NONE        AUDIT-MANAGE-PASS = NONE
  AUDIT-ACCESS-FAIL = NONE        AUDIT-MANAGE-FAIL = NONE

  LICENSE = OFF  PROGID = OFF  CLEARONPURGE = OFF  PERSISTENT = OFF
```

By default, the INFO command output is displayed in report form. To view this output as SAFECOM commands rather than as a report:

```
=DISPLAY AS COMMANDS ON
=INFO DISKFILE rpt01, DETAIL
```

The following information appears:

```
ADD    DISKFILE   $DATA.SALES              .RPT01
ALTER  DISKFILE   $DATA.SALES              .RPT01     ,&
          ACCESS    002,005       (R,W,E,P,  0)
ALTER  DISKFILE   $DATA.SALES              .RPT01     ,&
          ACCESS    002,*         (R          )
```

To return to the default setting of displaying the INFO and SHOW command output in report form:

```
=DISPLAY COMMANDS OFF
```

DISPLAY DETAIL Option

DISPLAY DETAIL controls the DETAIL option of the INFO command for an entire SAFECOM session. Most INFO commands provide additional detailed information when you specify the DETAIL option. However, this option applies only to the current command, not the entire session. To turn this option on for an entire session, use the DISPLAY DETAIL command.

```
DISPLAY DETAIL [ ON | OFF ]
```

`DISPLAY DETAIL`

entered without ON or OFF, turns on the DETAIL option for an entire session.

`ON`

turns on the DETAIL option for a session.

`OFF`

turns off the DETAIL option for a session.

Considerations

- If you use DISPLAY DETAIL OFF to turn off the DETAIL option for a session, you can override it for a single INFO command by specifying the DETAIL option in that command.

Examples

This example turns on the DETAIL option of the INFO command for the entire session:

```
=DISPLAY DETAIL ON
```

DISPLAY HEADERS Option

DISPLAY HEADERS controls the display of heading lines in INFO command reports for a session. SAFECOM normally displays a heading line above each object reported on by an INFO command. DISPLAY HEADERS allows you to either suppress the display of the heading line or specify that it should appear only once in an INFO report. This feature can be convenient if, for example, you are requesting information on many objects in a single INFO command.

```
DISPLAY HEADERS [ ON | OFF | ONCE ]
```

`DISPLAY HEADERS`

entered without ON, OFF, or ONCE, turns on the display of heading lines for the session.

`ON`

turns on the display of heading lines.

`OFF`

turns off the display of heading lines

`ONCE`

causes the heading line to appear once at the start of the INFO report.

Examples

This example shows the difference in an INFO command display after using the DISPLAY HEADERS command to specify HEADERS ONCE.

Assume that the subvolume SALES contains three files that have been added to the Safeguard database. Use the following INFO command to check the authorization records for these files:

```
=INFO DISKFILE $data.sales.*
```

The following information appears:

```
                  LAST-MODIFIED     OWNER    STATUS    WARNING-MODE
$DATA.SALES
 REPORT1         18JUL88, 11:00       2,1    THAWED         OFF

 NO ACCESS CONTROL LIST DEFINED!

                  LAST-MODIFIED     OWNER    STATUS    WARNING-MODE
$DATA.SALES
 REPORT2         18JUL88, 11:02       2,1    THAWED         OFF

 NO ACCESS CONTROL LIST DEFINED!

                  LAST-MODIFIED     OWNER    STATUS    WARNING-MODE
$DATA.SALES
 REPORT3         18JUL88, 11:05       2,1    THAWED         OFF

 NO ACCESS CONTROL LIST DEFINED!
```

Use the following DISPLAY HEADERS command to eliminate the multiple heading lines and make the report more legible:

```
=DISPLAY HEADERS ONCE
```

Then issue the same INFO command:

```
=INFO DISKFILE $data.sales.*
```

The following information appears:

```
                  LAST-MODIFIED     OWNER    STATUS    WARNING-MODE
$DATA.SALES
 REPORT1         18JUL88, 11:00       2,1    THAWED        OFF

 NO ACCESS CONTROL LIST DEFINED!

$DATA.SALES
 REPORT2         18JUL88, 11:02       2,1    THAWED        OFF

 NO ACCESS CONTROL LIST DEFINED!

$DATA.SALES
 REPORT3         18JUL88, 11:05       2,1    THAWED        OFF

 NO ACCESS CONTROL LIST DEFINED!
```

DISPLAY PROMPT Option

DISPLAY PROMPT controls the text displayed as your SAFECOM prompt. Normally, the SAFECOM prompt is an equal sign (=). The DISPLAY PROMPT command allows you to change this prompt for your current SAFECOM session.

```
DISPLAY PROMPT [ prompt-item ]
               [ ( prompt-item [ , prompt-item ] ... ) ]
```

`DISPLAY PROMPT`

entered by itself without any *`prompt-item`*, causes the default SAFECOM prompt (=) to be displayed.

`prompt-item`

specifies the text to be added to the standard SAFECOM prompt. If you include multiple prompt items in a DISPLAY PROMPT command, they must be separated by commas and enclosed in parentheses.

You can specify the following prompt items:

```
"string"
ASSUME OBJECTTYPE
COMMAND NUMBER
CPU
DATE
END
PROCESS NAME
PROCESS NUMBER
SUBVOLUME
SYSTEM NAME
SYSTEM NUMBER
TIME
USER NAME
USER NUMBER
VOLUME
```

`"`*`string`*`"`

is variable text string, up to 80 characters, to be displayed in the SAFECOM prompt. Enclose the text string in single or double quotation marks. To include quotation marks as part of the actual the string, enclose the string within the other type of quotation marks.

`ASSUME OBJECTTYPE`

specifies that the currently assumed object type is displayed in the SAFECOM prompt. If no object type is assumed, nothing is displayed.

`COMMAND NUMBER`

specifies that the current SAFECOM command line number is displayed in the SAFECOM prompt.

`CPU`

specifies that the number of the CPU in which SAFECOM is currently running is displayed in the SAFECOM prompt.

`DATE`

specifies that the current date is displayed in the SAFECOM prompt. The date is displayed in the form *`mm/dd/yyyy`*.

`END`

specifies that the equal sign (=) is not displayed to terminate the SAFECOM prompt and that any *`prompt-item`* following the word END is ignored.

`PROCESS NAME`

specifies that the current process name is displayed in the SAFECOM prompt.

`PROCESS NUMBER`

specifies that the current process number is displayed in the SAFECOM prompt. The process number is a number from 0 through 255.

`SUBVOLUME`

specifies that the current subvolume name is displayed in the SAFECOM prompt.

`SYSTEM NAME`

specifies that the current system name is displayed in the SAFECOM prompt.

`SYSTEM NUMBER`

specifies that the current system number is displayed in the SAFECOM prompt. The system number is a number from 0 through 255.

`TIME`

specifies that the current time is displayed in the SAFECOM prompt. The time is displayed in the form *`hh:mm`*.

`USER NAME`

specifies that the current user name is displayed in the SAFECOM prompt. The name is displayed in the form *`group name.member name.`*

`USER NUMBER`

specifies that the current user ID is displayed in the SAFECOM prompt. The user ID is displayed as *`group number,member number`*.

`VOLUME`

specifies that the current volume name is displayed in the SAFECOM prompt.

Considerations

- If used, END should be the last prompt item specified. Any prompt items following END are ignored.

Examples

1. This command adds the current command line number to the SAFECOM prompt:

   ```
   =DISPLAY PROMPT COMMAND NUMBER
   2=
   ```

2. This command adds the user name for the user ADMIN.BILL to the SAFECOM prompt:

   ```
   =DISPLAY PROMPT USER NAME
   ADMIN.BILL=
   ```

3. This command changes the SAFECOM prompt for the user ADMIN.BILL to the user name followed by a space and a greater-than sign (>). END removes the equal sign from the prompt.

   ```
   =DISPLAY PROMPT (USER NAME, " >", END)
   ADMIN.BILL >
   ```

DISPLAY USER Option

DISPLAY USER controls whether user identities are displayed as user IDs or as user names in SAFECOM reports. SAFECOM normally identifies users by their user IDs. To identify users by their user names, use the DISPLAY USER command.

If a user or node has been deleted, the Safeguard software cannot display that user's user name or node's node name. Such a user or node is always identified by its ID.

```
DISPLAY USER [ AS ] { NAME | NUMBER }
```

AS

is an optional keyword that you can use for clarity. It is not required in the command.

`NAME`

specifies that users and nodes are identified by user and node names.

`NUMBER`

specifies that users and nodes are identified by user and node IDs.

Examples

In this example, DISPLAY USER specifies that user identities are displayed as user names rather than user IDs:

```
=DISPLAY USER NAME
```

DISPLAY WARNINGS Option

DISPLAY WARNINGS controls the display of warning messages on INFO DISKFILE reports for the current session. SAFECOM normally displays a warning message if you issue an INFO DISKFILE command for a file that has not been added to the Safeguard database. DISPLAY WARNINGS provides a means of suppressing this message. This option can be a convenient if, for example, you are requesting information on all files in a subvolume.

```
DISPLAY WARNINGS [ ON | OFF ]
```

`DISPLAY WARNINGS`

entered without ON or OFF, turns display warnings on.

`ON`

turns display warnings on.

`OFF`

turns display warnings off.

Considerations

- You can also control display warnings with the WARNINGS option of the INFO DISKFILE command.

Examples

This example shows the difference in an INFO command display after DISPLAY WARNINGS is used to turn the display warnings off.

Assume that the subvolume SALES contains three files, and two of them have not been added to the Safeguard database. Use the following INFO command to check the authorization records for these files:

```
=INFO DISKFILE $data.sales.*
```

The following information appears:

```
                  LAST-MODIFIED     OWNER    STATUS    WARNING-MODE
$DATA.SALES
 REPORT1         18JUL88, 11:00       2,1    THAWED         OFF

 NO ACCESS CONTROL LIST DEFINED!

* WARNING * RECORD FOR DISKFILE $DATA.SALES.REPORT2 NOT FOUND

* WARNING * RECORD FOR DISKFILE $DATA.SALES.REPORT3 NOT FOUND
```

Use the following command to turn off the warning messages:

```
=DISPLAY WARNINGS OFF
```

Then issue the same INFO command:

```
=INFO DISKFILE $data.sales.*
```

The following information appears when the warnings are off:

```
                  LAST-MODIFIED     OWNER    STATUS    WARNING-MODE
$DATA.SALES
 REPORT1         18JUL88, 11:00       2,1    THAWED        OFF

 NO ACCESS CONTROL LIST DEFINED!
```

ENV Command

ENV displays the current default values of the environmental parameters (that is, the current default system, volume, OUT file, log file, assumed object class, and DISPLAY settings).

The ENV listing reports the current environmental parameter values as valid SAFECOM commands. For this reason, you can use the ENV command to save your current environment in an EDIT or TEDIT file and later execute those commands to reestablish the saved environment, through the OBEY command. (For more information, see OBEY Command on page 4-20.)

```
ENV [ / OUT listfile / ] [ env-parm ] [ , env-parm ]...
```

ENV

entered with no *env-parm*, produces an ENV report on all the environmental parameters. (They are listed here under *env-parm*.)

listfile

redirects the ENV listing to *listfile*. Specify *listfile* as any file name.

If *listfile* does not exist, SAFECOM creates an EDIT-format file by that name and writes the ENV report to that file. If *listfile* does exist, SAFECOM opens the file and appends the ENV report.

env-parm

is any one of these environmental parameters:

```
SYSTEM
VOLUME
OUT
LOG
ASSUME
WARNINGS
USER
DETAIL
AS COMMANDS
HEADERS
PROMPT
```

Examples

1. This ENV command requests a report on all the current environmental parameter values. The report is sent to a file called $DATA.SECURE.ENVPARMS.

```
=ENV / OUT envparms /
=
```

After this ENV command completes, $DATA.SECURE.ENVPARMS contains these text lines:

```
SYSTEM    \LA
VOLUME    $DATA.SECURE
OUT       \LA.$TERM            -- Interactive, OUT=IN --
LOG       $DATA.SECURE.SESSLOG
ASSUME    DISKFILE
DISPLAY WARNINGS ON
DISPLAY USER AS NAME
DISPLAY DETAIL OFF
DISPLAY AS COMMANDS OFF
DISPLAY HEADERS ON
DISPLAY PROMPT
```

The comment following the OUT parameter indicates that SAFECOM is running interactively. ($TERM is being used for both input and output.)

2. This ENV command lists the current VOLUME environmental parameter:

```
=ENV VOLUME
     VOLUME    $DATA.SECURE
```

EXIT Command

EXIT terminates an interactive SAFECOM session and returns control of your terminal to your command interpreter.

```
EXIT
```

Consideration

- You can also press Ctrl-Y to exit SAFECOM. Ctrl-Y is equivalent to end-of-file (EOF). (Ctrl-Y means to hold down the Ctrl key while pressing the Y key.)

FC Command

The FC command lets you retrieve, edit, and execute a command line you have previously entered during the current session. FC is useful for correcting mistyped commands and for entering a series of commands that differ by only a few characters.

FC displays the specified command line and prompts you to make your changes on the next blank line. You can enter FC subcommands (such as R, I, or D) on the new blank line to delete, insert, or replace characters.

After you enter the FC subcommands, pressing the RETURN key instructs FC to do three things: make the changes you specified, redisplay the command line as modified, and prompt for more changes.

```
FC [ linenum      ]
   [ -linenum     ]
   [ string       ]
   [ "string"     ]
```

`FC`

entered with no line number or text string, specifies that the last command line in the command history buffer is displayed.

linenum

is a positive integer that specifies the number of the command line in the history buffer that you want to retrieve.

-linenum

is a negative integer that specifies the number of the command line to be retrieved relative to the current line number.

string

is a text string. The FC command finds and displays the most recent command in the history buffer that begins with the specified text string.

"*string*"

is a text string enclosed in quotes. The FC command finds and displays the most recent command in the history buffer that contains the specified text string. The command need not begin with the specified string.

FC Editing Subcommands

When you execute the FC command, it displays the specified command and positions the cursor on the next line. This blank line is the command editing line in which you can use the FC editing subcommands. The editing subcommands modify the characters displayed above them in the command line.

When you move the cursor in the command editing line, use only the spacebar and the backspace key. Do not use the arrow keys to move the cursor in the command editing line.

FC supports the following editing subcommands:

`//`

separates two or more subcommands in the command editing line.

`R | r` *`replacement-string`*

replaces characters in the command line, starting with the character directly above the R or r. A *`replacement-string`* preceded by R or r can be any string of characters, including spaces, and can itself begin with R, I, or D (or r, i, or d). Characters in *`replacement-string`* replace characters in the command line on a one-for-one-basis.

If the separator character sequence (//) follows this subcommand, all characters in replacement-string up to //, including blanks, replace characters in the command line. Otherwise, replacement ends when you press RETURN.

`I | i` *`insertion-string`*

inserts characters into the command line in front of the character displayed above the I or i. If // follows this subcommand, all characters in *`insertion-string`* up to //, including blanks, are inserted into the command line. If no // appears, all characters up to the RETURN are inserted.

`D | d`

deletes characters in the command line. The character directly above the D or d is deleted.

`replacement-string`

is any subcommand that does not begin R, I, or D (r, i, or d). Characters in *`replacement-string`* replace the characters directly above them in the command line on a one-for-one basis.

To execute the modified command displayed by FC, press RETURN at the first character position on the command editing line.

Considerations

- To abort the FC command, enter only the subcommand separator (//) on the new blank line and then immediately press RETURN. The (possibly altered) command line is discarded without execution. You can also press Ctrl-Y to stop the FC command.
- If you enter FC alone, the last command you entered is displayed.

Examples

1. The following FC command retrieves command line number 6 and then uses the i subcommand to change the displayed command:

```
=FC 6
=ADD DISKFILE $DATA.ACCT.RPT04, LIKE OUT01
.                  i01
=ADD DISKFILE $DATA01.ACCT.RPT04, LIKE OUT01
```

2. The following FC command retrieves the last command in the history buffer. The implied replacement string subcommand is used to change the displayed command.

```
=FC
=ADD DISKFILE $DATA01.ACCT.RPT04, LIKE OUT01
.                              5
=ADD DISKFILE $DATA01.ACCT.RPT05, LIKE OUT01
```

3. The following FC command retrieves the last command in the history buffer that begins with ADD. The implied replacement string subcommand, the d subcommand, and the r subcommand are used to change the displayed command.

```
=FC ADD
=ADD DISKFILE $DATA01.ACCT.RPT05, LIKE OUT01
. LT//             dd//                rRP
=ALT DISKFILE $DATA.ACCT.RPT05, LIKE RPT01
```

HELP Command

Enter the HELP command to display the syntax for all the SAFECOM commands.

```
HELP [ / OUT listfile / ] [ topic  ]
                          [ ALL    ]
                          [ *      ]
```

listfile

redirects the HELP command listing to *listfile*. For *listfile*, specify any file name.

If *listfile* does not exist, SAFECOM creates an EDIT-format file and then writes the HELP report to that file. If *listfile* does exist, SAFECOM opens the file and appends the HELP report.

topic

is the topic for which SAFECOM displays information. For a brief list of topics, enter HELP without any parameters.

ALL

displays help text for all commands (typically used with the OUT *listfile* option).

*

displays all the help screens (typically used with the OUT *listfile* option).

HISTORY Command

SAFECOM maintains a history buffer of the command lines you enter during an interactive session. You can display a specified number of these commands with the HISTORY command. This command also allows you to clear the last command or all commands from the history buffer.

The HISTORY command adds line numbers to the commands it displays even if your normal SAFECOM prompt does not include line numbers.

```
HISTORY [ lines      ]
        [ RESET LAST ]
        [ RESET ALL  ]
```

HISTORY

entered by itself, displays the 10 most recent lines in the history buffer.

lines

specifies the number of most recently entered commands to be displayed.

RESET LAST

specifies that the most recently entered command in the history buffer is cleared and the current command line number is set to the number of the cleared command.   The HISTORY RESET LAST command is also cleared from the history buffer.

RESET ALL

specifies that all commands in the history buffer are cleared and the command line number is reset to 1.

Examples

1. The following command displays the last four commands entered during the current session. The example assumes that the HISTORY command is the ninth command entered during the session.

```
=HISTORY 4

6=ASSUME DISKFILE
7=SHOW
8=ADD RPT05, ACCESS 14,* R
9=HISTORY 4

=
```

2. The following sequence of commands shows the function of the HISTORY RESET LAST command. To help you understand the example, the DISPLAY PROMPT command is used to display the command line number in the SAFECOM prompt. In this example, the HISTORY RESET LAST command deletes the INFO DISKFILE command from the history buffer and resets the current command line number to that of the deleted command. The HISTORY RESET LAST command itself is not retained in the history buffer.

```
=DISPLAY PROMPT COMMAND NUMBER
5=INFO DISKFILE $DATA.HAROLD.SEC07
6=HISTORY RESET LAST
5=HISTORY 2

4=DISPLAY PROMPT COMMAND NUMBER
5=HISTORY 2

6=
```

LOG Command

LOG enables and disables the SAFECOM logging feature. When logging is enabled, SAFECOM writes a complete record of your session to a log file. All the command lines you enter and all the text lines SAFECOM produces in response to those commands are saved in the log file.

Entering a LOG command with a log-file name turns the logging feature on. To turn off logging, enter a LOG command without a log-file name.

```
LOG [ logfile ]
```

`LOG`

entered without *logfile*, turns the logging feature off. The current *logfile*, if any, is closed, and no log output is produced.

logfile

establishes a log file for SAFECOM. You can specify any file name. If *logfile* does not exist, SAFECOM creates an EDIT-format file and writes the session log records to that file. If *logfile* exists, SAFECOM opens the file and appends the log records.

Considerations

- Changing log files in the middle of a session

  You can change the log file in the middle of a session by entering a LOG command that specifies a different file as *logfile*. For example, suppose you are currently logging your session in a file called LOGFILE1, but you want to change to LOGFILE2:

  ```
  =LOG logfile2
  ```

  SAFECOM closes LOGFILE1, opens LOGFILE2, and writes all subsequent input commands and output text to LOGFILE2.

- Determining your current log file

  To display the name of your current log file:

  ```
  =ENV LOG
      LOG      $DATA.SALES.LOGFILE2
  ```

Example

This LOG command sends a record of your subsequent SAFECOM session to the file $GARB.RECORD.MAY5:

```
=LOG $garb.record.may5
```

OBEY Command

OBEY temporarily suspends input from your terminal and instead accepts input commands from a command file. SAFECOM displays each command as it is performed. After executing all the commands in the command file, SAFECOM prompts you for the next command.

You can use command files to perform any number of SAFECOM commands through a single OBEY command. A command file can contain any SAFECOM command except FC.

Command files are typically created through an editor such as TEDIT. Each text line in the file is treated as a separate SAFECOM command line. The same rules that apply to interactive SAFECOM commands apply to lines in a command file.

```
O[BEY] [ / OUT listfile / ] command-file
```

`OUT` *`listfile`*

redirects SAFECOM output to *`listfile`* for all the commands in *`command-file`*. For *`listfile`*, specify any file name.

If *`listfile`* does not exist, SAFECOM creates an EDIT-format file and then writes all output text to that file. If *`listfile`* exists, SAFECOM opens the file and appends the output text.

`command-file`

is the name of a file containing SAFECOM commands (usually an EDIT-format file). The name can be any file name.

Considerations

- Nesting command files

  Command files can be nested, with one OBEY command file within another.

  Command files can be nested to a depth of four. For example, from the SAFECOM prompt, you could enter the command OBEY FILE1. FILE1 could contain the command OBEY FILE2, FILE2 could contain the command OBEY FILE3, and FILE3 could contain the command OBEY FILE4.

- Command-file error handling

  When SAFECOM encounters a syntax error or an invalid object type while executing a command file, it aborts processing the command file and calls ABEND.

  If it encounters other errors, it processes the commands till the end of file and then calls ABEND.

  As a result, TACL returns -6 as the MESSAGECODE and a non-zero value as the COMPLETIONCODE.

Note. ABEND (in case of any error in OBEY file) is supported only on systems running H06.28 and later H-series RVUs and J06.17 and later J-series RVUs.

- Stopping the execution of a command file

  Press the BREAK key to stop the execution of a command file. SAFECOM stops processing commands and closes the command file.

- Using FC after executing a command file

  When you enter the FC command after executing a command file, the FC command displays the OBEY command rather than the last command executed in the command file.

Example

This example shows the use of a command file named $DATA.SECURE.INFOADMN. This command file prints an INFO report about all the users in the ADMIN group:

```
=OBEY $data.secure.infoadmn
=OUT $s.#lp                       -- redirect output to $S.#LP
=INFO USERS admin.* , DETAIL      -- report on all admin users
=OUT                              -- redirect output to terminal
=
```

OUT Command

OUT directs SAFECOM output text to a specified file. SAFECOM output text includes both input commands and the response to those commands. Output text directed to a specific file is not echoed to the screen. Typically, OUT is used in command files, where it directs the command-file output to a list file.

```
OUT [ listfile ]
```

`OUT`

entered without *listfile*, closes the current *listfile*. SAFECOM then opens the file specified by the OUT option when SAFECOM was started (typically your home terminal).

listfile

is the file to which SAFECOM is to write its output text. For *listfile*, specify any file name.

If *listfile* does not exist, SAFECOM creates an EDIT-format file and then writes all output text to that file. If *listfile* exists, SAFECOM opens the file and appends the output text.

Considerations

- To display the use of your current OUT file, use the ENV OUT command. For example:

```
=ENV OUT
    OUT      $TERM3           -- Interactive, OUT = IN --
=
```

- SAFECOM error messages appear on the terminal screen when you specify an OUT file.

Example

This example shows the use of the OUT command to produce a listing of all the SAFECOM help screens:

```
=OUT $s.#lp1
=HELP ALL
=OUT
```

The last OUT command redirects SAFECOM output text to the original output file; in this case, the home terminal.

SYNTAX Command

SYNTAX enables and disables syntax-only mode. In syntax-only mode, SAFECOM only checks the syntax of commands. It does not execute the commands. You can execute only four SAFECOM commands in syntax-only mode: SYNTAX, ASSUME, OBEY, and EXIT.

```
SYNTAX [ ONLY ] { ON | OFF }
```

`SYNTAX [ ONLY ]`

specifies that SAFECOM syntax-only mode is enabled or disabled. The keyword ONLY is optional and can be included for readability.

`ON`

specifies that syntax-only mode is enabled.

`OFF`

specifies that syntax-only mode is disabled.

Consideration

After each command is checked for syntactical correctness, SAFECOM issues a message.

For each command that is syntactically correct:

```
* WARNING * SAFECOM IS IN SYNTAX ONLY MODE; COMMAND NOT
EXECUTED.
```

For each command that contains a syntax error:

```
ILLEGAL SYNTAX
```

Example

To enable syntax-only mode:

```
=SYNTAX ON
```

SYSTEM Command

SYSTEM establishes a default system name. SAFECOM uses the default system name to expand partially qualified file names that do not include a system name.

When you first run SAFECOM, the name of your local system is used as the initial default system name.

```
SYSTEM [ \system-name ]
```

`SYSTEM`

entered without `\system-name`, sets the default system name to the name of the system you are currently using.

`\system-name`

is a valid system name to be used as the default for subsequent SAFECOM commands.

Considerations

- The SYSTEM command has no effect on user names because the system name is not part of a user name. For example, if you execute an INFO USER command after executing a SYSTEM command for a remote system, you still get information on the local user.
- To display your current default system name, use the ENV SYSTEM command. For example:

```
=ENV SYSTEM
   SYSTEM  \LA
```

Example

This example uses the SYSTEM command in a sequence of commands:

```
=ENV SYSTEM                       -- Displays the default
    SYSTEM \NY                       system name, \NY

=SYSTEM \LA                       -- Establishes a new
                                     default system name

=ENV SYSTEM                       -- Displays the new
    SYSTEM \LA                       default system name, \LA

=ASSUME DISKFILE                  -- Establishes DISKFILE as
                                     the default object class
```

```
=INFO $data.sales.report1        -- Displays a report on
                                    \LA.$DATA.SALES.REPORT1
```

```
                  LAST-MODIFIED     OWNER    STATUS    WARNING-MODE
\LA.$DATA.SALES
 REPORT1         14OCT85, 11:46       1,2    THAWED         OFF

    001,002      R,W,E,P
 \*.001,*        R
```

```
=SYSTEM
=ENV SYSTEM                      -- Reestablishes \NY as the
    SYSTEM \NY                      default system name
```

VOLUME Command

VOLUME establishes a default disk volume name and a default subvolume name. SAFECOM uses the current default volume and subvolume names to expand partially qualified disk file names. SAFECOM uses the current volume name to expand partially qualified subvolume names.

When you start SAFECOM, the default volume and subvolume names established through your command interpreter become your current SAFECOM default volume and subvolume names.

```
VOLUME [ $volume           ]
       [ $volume.subvolume ]
       [ subvolume         ]
```

VOLUME

entered with no volume or subvolume names, sets the default volume and subvolume names to the original names you established through your command interpreter before running SAFECOM.

$*volume*

sets the current default volume name to $*volume* but leaves the current default subvolume name unchanged.

$*volume.subvolume*

sets both the current default volume and subvolume names to $*volume.subvolume*.

subvolume

sets the current default subvolume name to *subvolume* but leaves the current default volume name unchanged.

Consideration

To display your current default volume and subvolume names, use the ENV VOLUME command. For example:

```
=ENV VOLUME
   VOLUME  $DATA.SALES
=
```

Example

This example uses the VOLUME command to establish default volume and subvolume names within a sequence of commands:

```
=OUT $s.#lp           -- Directs output to $S.#LP

=ASSUME DISKFILE      -- Establishes DISKFILE as the
                         default object class

=VOLUME $data.sales   -- Establishes the default volume
                         and subvolume

=INFO report1         -- Reports on $DATA.SALES.REPORT1

=INFO $system.report2 -- Reports on $SYSTEM.SALES.REPORT2

=INFO admin.report3   -- Reports on $DATA.ADMIN.REPORT3

=INFO *               -- Reports on all disk files on the
                         $DATA.SALES subvolume
```

? Command

The ? command lets you retrieve a command you previously entered during the current session. You can request the command by line number, relative line number, or text string.

```
? [ linenum  ]
  [ -linenum ]
  [ string   ]
  [ "string" ]
```

?

entered with no line number or text string, specifies that the last command line in the command history buffer is displayed.

linenum

is a positive integer that specifies the number of the command line in the history buffer that you want to retrieve.

-linenum

is a negative integer that specifies the number of the command line to be retrieved relative to the current line number.

string

is a text string. The ? command finds and displays the most recent command in the history buffer that begins with the specified text string.

"*string*"

is a text string enclosed in quotation marks. The ? command finds and displays the most recent command in the history buffer that contains the specified text string. The command need not begin with the specified string.

Examples

1. The following ? command retrieves command line number 6:

```
=? 6
=ADD DISKFILE SEC03, LIKE SEC01
```

2. The following ? command retrieves the last command in the history buffer:

```
=?
=INFO VOLUME $DATA
```

3. The following ? command retrieves the last command in the history buffer that contains the character string RPT06:

```
=? "RPT06"
=ADD DISKFILE $DATA01.ACCT.RPT06, LIKE RPT01
```

4. The following ? command retrieves the last command in the history that begins with the text string ALT:

```
=? ALT
=ALTER DISKFILE SEC12, ACCESS 12,34 *
```

! Command

The ! command lets you retrieve and execute a command you previously entered during the current session. You can request the command by line number, relative line number, or text string.

```
! [ linenum  ]
  [ -linenum ]
  [ string   ]
  [ "string" ]
```

`!`

entered with no line number or text string, specifies that the last command line in the command history buffer is to be displayed and executed.

`linenum`

is a positive integer that specifies the number of the command line in the history buffer that you want to retrieve and execute.

`-linenum`

is a negative integer that specifies the number of the command line to be retrieved and executed relative to the current line number.

`string`

is a text string. The ! command displays and executes the most recent command in the history buffer that begins with the specified text string.

`"`*`string`*`"`

is a text string enclosed in quotes. The ! command displays and executes the most recent command in the history buffer that contains the specified text string. The command need not begin with the specified string.

Examples

1. The following ! command retrieves and executes command line number 13:

```
=! 13
=RESET DISKFILE
=
```

2. The following ! command retrieves and executes the last command in the history buffer:

```
=!
=SET DISKFILE PERSISTENT ON
=
```

Comment Delimiters

Use two hyphens (--) to delimit comments within a command line. The comment can appear anywhere within the command line but cannot be embedded within a word. In addition, the comment should not contain a semicolon (;). Any part of a comment following a semicolon is lost because the semicolon causes line termination.

Comments can also appear on their own line when delimited before and, optionally, after with two hyphens.

```
-- any-phrase { -- | end-of-line }
```

`--`

(two hyphens) are the delimiters that separate the command from the comment.

any-phrase

is any phrase or descriptive remark that follows the delimiter. The phrase cannot contain -- (two hyphens).

end-of-line

the carriage return is the *end-of-line* terminator. Alternately, the two hyphens can be eliminated as delimiters if the *end-of-line* terminator is included after your comment in *any-phrase*. However, the comment line must end with either the delimiters or the *end-of-line* character.

Example

Comment lines are particularly useful in command files, as an example, to annotate the way a file works.

This example demonstrates the use of the comment line with the INFO command:

```
=OBEY --year-end report-- data.sales.report1
=--sent to operations line printer--
=OUT $s.#lp
=--report on all admin users--
=INFO USER admin.* , DETAIL
=OUT --redirect output to the home terminal--
```

Continuation Character

The ampersand character (&) is used to indicate that a SAFECOM command line is continued on the next line. The ampersand must be the last character to appear in the line to be continued. You can break the command to be continued at any point.

You can use the ampersand to enter a single command string across any number of command lines, but the total number of characters in the entire command string cannot exceed 528.

Example

In the following example, the & continuation character allows for continuation of a command over three lines:

```
=ALTER DISKFILE $data.user2.reports, AUDIT-ACCESS-FAIL all,&
=ACCESS 12,56 *; (test.sharon, software.arthur, &
=software.karl) (read, write)
```

In the example, an ALTER DISKFILE command alters the authorization record for the disk file $data.user2.reports. The ALTER DISKFILE command includes an AUDIT-ACCESS-FAIL specification and four access control list entries. Semicolons separate the elements of the ACCESS specification. Commas separate other command elements.

5 User Security Commands

SAFECOM user security commands are restricted such that, only specific users can execute the commands and thereby control user security. These users include system managers, security administrators, and group managers, as qualified by the list of users specified with OBJECTTYPE USER.

SAFECOM commands can add user IDs to the system, delete user IDs from the system, and suspend user IDs ability to log on to the system. They can also specify auditing for attempts to authenticate users, attempts by the user to perform certain actions, and attempts to manage user authentication records.

This section contains:

- A description of who can add new users to the system and who can manage the Safeguard security controls for system users
- A summary table of the user security commands
- Detailed syntax for each user security command

Who Can Manage User Security

If no access control list has been defined for OBJECTTYPE USER, only the super ID can initially add user IDs to the system. When the super ID (with the user ID 255,255) adds the first user to a group, that group is created implicitly. A group created in this manner is known as an administrative group because it is used to administer user authentication records. Groups can also be created with the ADD GROUP command. For more information about groups, see Section 7, Group Commands.

Frequently, the first user added to an administrative group is the group manager (with user ID *n*,255). Then the group manager ID can add other users to form that administrative group. The super ID also can add users to any group, but only the super ID can add group managers. However, the access control list for OBJECTTYPE USER can specify a list of users who can add other users. For more information, see Section 12, OBJECTTYPE Security Commands.

A user authentication record can have multiple owners. The OWNER attribute in a user authentication record designates the record’s primary owner. The OWNER-LIST attribute optionally designates one or more secondary owners. By default, the OWNER attribute contains the user ID of the user who first created the user authentication record. The OWNER and OWNER-LIST attributes can be changed with a SET USER command before the record is created, or they can be changed with an ALTER USER command after the record is created. These record owners can change the security attributes in the user’s authentication record and therefore control the user’s ability to log on to the system.

Only the primary and secondary record owners, the primary owner’s group manager, and the super ID can change a user’s security attributes, suspend and restore the user's ability to log on to the system, and delete the user (ALTER USER, FREEZE USER, THAW USER, and DELETE USER commands, respectively).

The original primary owner and the secondary owners of a user authentication record can change the OWNER attribute to the user ID of any other user. That other user then has control of the user's ability to access the system. At any time, the new primary owner (or the secondary owners or the primary owner's group manager or the super ID) can transfer ownership to yet another user.

When the Safeguard software converts a user authentication record that was added by the Guardian ADDUSER program, it sets the OWNER attribute of that record to the group manager of the administrative group to which the user belongs.

Note. All the Guardian file security settings like Safeguard ACLs and SEEP authorization rulings are ignored if the PAID of the user is 255,255. Any security checks within SAFECOM are ignored for users with a PAID of 255,255.

The ability to display a user's security attributes through the INFO USER command is restricted to these users:

- The user
- The primary and secondary owners of the user's authentication record
- The primary owner's group manager
- The super ID

Table 5-1 shows who can use the user security commands to display, add, modify, or delete a user's authentication record.

Table 5-1. Who Can Use the User Security Commands (page 1 of 2)

USER Command	Who Can Use
SET USER LIKE	User, primary and secondary record owners, primary owner's group manager, and super ID
INFO GROUP	User, primary and secondary record owners, primary owner's group manager, and super ID
INFO USER	User, primary and secondary record owners, primary owner's group manager, and super ID
ALTER USER	Primary and secondary record owners, primary owner's group manager, and super ID
FREEZE USER	Primary and secondary record owners, primary owner's group manager, and super ID
THAW USER	Primary and secondary record owners, primary owner's group manager, and super ID
DELETE USER	Primary and secondary record owners, primary owner's group manager, and super ID

Table 5-1. Who Can Use the User Security Commands (page 2 of 2)

USER Command	Who Can Use
ADD USER	If no ACL exists for OBJECTTYPE USER, the local group manager can add a member of an existing group. The local super ID can add members of a group or add a group manager. If an ACL exists for OBJECTTYPE USER, only members listed in that ACL can add users. (Thereafter, the owner can manage the user record). For a description of the OBJECTTYPE command, see Section 12, OBJECTTYPE Security Commands.

Note. The USER Command, INFO GROUP, is supported only on systems running H06.09 and later H-series RVUs.

User Security Command Summary

Table 5-2 summarizes each of the user security commands.

Table 5-2. User Security Command Summary

Command	Function
ADD USER	Adds a user to the system and creates an authentication record for that user with the user attribute values specified in the command. For any unspecified attributes, the current default values are used. (To set default attribute values, use the SET USER command.) After being added, a user can log on to the system.
ALTER USER	Changes the value of one or more user attributes in a user authentication record.
DELETE USER	Deletes a user from the system (and deletes the user's authentication record). The user cannot log on to the system.
FREEZE USER	Temporarily suspends a user's ability to log on to a system. (The THAW USER command restores a user's ability to log on.)
INFO USER	Displays the existing attribute values defined for a user in the user's authentication record.
RESET USER	Sets one or more current default user attribute values to predefined values.
SET USER	Sets one or more default user attribute values to specified values. When a user is added, the current default attributes are used for any attribute not specified in the ADD USER command.
SHOW USER	Displays the current default values of the user attributes.
THAW USER	For frozen users, restores their ability to log on to the system.

The Syntax of User Security Commands

The rest of this section contains individual syntax descriptions for the user security commands. Commands appear in alphabetical order, and most of the descriptions contain these elements:

- A summary of the function performed by the command, including the restrictions on who can use the command
- Descriptions of the command parameters and variables
- The format for the command listing or report (for commands that produce displays or listings)
- Considerations for the use of the command
- Examples of command usage

ADD USER Command

ADD USER adds a user to the system and creates a Safeguard authentication record for that user. Once a new user is added to the system (and knows what the initial password is), the user can log on to the system.

By default, only the super ID (user ID 255,255) can use the ADD USER command to add the first member of a new administrative group to a system. However, if an access control list has been defined for OBJECTTYPE USER, that list specifies the users who are allowed to add other users. For more information, see Section 12, OBJECTTYPE Security Commands. (Adding the first member of a new group implicitly adds a new administrative group to a system.)

Also by default, only the group’s manager and the super ID are allowed to add users to a particular administrative group unless an access control list has been specified for OBJECTTYPE USER.

You can use the SET USER command to set default values for the user attributes and then use ADD USER to identify the user name and user ID of the record to which the default values are to be assigned. You can also specify attribute values in your ADD USER command. The current default values are used for any attributes left unspecified.

```
ADD USER group-name.member-name , group-num,member-num [ , ]

   [ LIKE user | user-attribute ] [ , user-attribute ] ...
```

USER

identifies USER as the object class of the ADD command. Omit it if USER is the assumed class. (For more information about assumed commands, see the ASSUME Command on page 4-3.)

group-name.member-name

is the user name of the user added to the system. If *group-name* does not match any existing *group-name* and if *group-num* is not assigned to an existing *group-name*, a new administrative group is added to the system, with this user being the first member of the group.

If *group-name* already exists, the *group-num* of the new user must match the *group-num* of the existing *group-name*, and the *member-name* and *member-num* assigned to the new user must not already be assigned to an existing group member.

Both *group-name* and *member-name* are from one to eight alphanumeric characters, the first of which must be alphabetic.

The *group-name* and *member-name* variables cannot contain wild-card characters.

group-num,member-num

is the user ID of the user to be added to the system. Both *group-num* and *member-num* must be numbers from 0 through 255. A group manager's user ID is *group-num*,255.

LIKE *user*

adopts the attribute values from the existing *user* authentication record as the attribute values for the authentication record being added. The *user* variable is one of:

```
alias
group-num,member-num
group-name.member-name
```

The *alias* variable is an existing user alias. For more information, see Section 6, User Alias Security Commands.

LIKE defines values for all user attributes except:

```
PASSWORD [password]
REMOTEPASSWORD \system-name remote-password
GUARDIAN [DEFAULT] SECURITY ["]string["]
GUARDIAN [DEFAULT] [SUB]VOLUME [\system.]$vol.subvol
CI-NAME [process-name]
PRIMARY-GROUP [ [ NAME ] group-name | NUMBER group-num ]
INITIAL-DIRECTORY [dir-path]
INITIAL-PROGRAM [prog-path]
INITIAL-PROGTYPE [prog-type]
TEXT-DESCRIPTION "[text]"
```

user-attribute

defines a user attribute value for the user being added. (The current default user values are used for any attributes not specified in the ADD USER command. For the initial default values, see RESET USER Command on page 5-35. You can change the initial default values with the SET command.)

The user-attributes values are:

```
OWNER [owner-id]
OWNER-LIST [[-]user-list]
PASSWORD [password]
USER-EXPIRES [date [, time]]
PASSWORD-MUST-CHANGE [EVERY num DAYS]
PASSWORD-EXPIRY-GRACE [num [DAYS]]
PASSWORD-EXPIRES [ date [ , time] ]
AUDIT-AUTHENTICATE-PASS [audit-spec]
AUDIT-AUTHENTICATE-FAIL [audit-spec]
AUDIT-MANAGE-PASS [audit-spec]
AUDIT-MANAGE-FAIL [audit-spec]
AUDIT-USER-ACTION-PASS [audit-spec]
AUDIT-USER-ACTION-FAIL [audit-spec]
REMOTEPASSWORD \system-name remote-password
DEFAULT-PROTECTION [ obj-attr ]
                   [ ( obj-attr [ , obj-attr ] ... ) ]
GUARDIAN [DEFAULT] SECURITY ["]string["]
GUARDIAN [DEFAULT] [SUB]VOLUME [\system.]$vol.subvol
INITIAL-DIRECTORY [dir-path]
INITIAL-PROGRAM [prog-path]
INITIAL-PROGTYPE [prog-type]
CI-PROG [prog-filename]
CI-LIB [lib-filename]
CI-CPU [cpu-number | ANY]
CI-NAME [process-name]
CI-SWAP [$vol.[subvol.filename]]
CI-PRI [priority]
CI-PARAM-TEXT [startup-param-text]
TEXT-DESCRIPTION "[text]"
```

Multiple *remote-password* entries are not allowed in a single command.

Because the CI-PARAM-TEXT, INITIAL-DIRECTORY, and INITIAL-PROGRAM attributes are text strings, each of them must be the last attribute specified in the command string. Therefore, you can specify only one of these attributes in a single command.

For a complete description of each *user-attribute*, see SET USER Command on page 5-40 and ALTER USER Command on page 5-10.

Considerations

- The ADDUSER program does not support all the ADD USER command attributes.

  The ADD USER command performs the same function as the ADDUSER program in the command interpreter. However, the ADDUSER program does not support all of the user attributes that the ADD USER command supports. To avoid confusion, you might want to remove the ADDUSER program from systems where the Safeguard software is installed.

Adding a user with the ADD USER command creates a Safeguard user authentication record for the user that contains predefined user attribute values. (For the predefined values, see RESET USER Command on page 5-35.)

- Newly installed Safeguard software expands the existing USERID file.

  When the Safeguard software is installed on a system that has an existing user community, it expands the existing USERID file to add the Safeguard user attributes for every user currently defined on the system. The record for each user is expanded the first time that user logs on after the Safeguard software has been installed. Safeguard retains existing security attributes that are common to both Safeguard and standard Guardian security, such as OWNER and GUARDIAN DEFAULT SECURITY.   Each user security attribute unique to Safeguard security is given its predefined value. For a description, see RESET USER Command on page 5-35.

- The user who implicitly creates a new group becomes the owner of that group.

  When you add the first user to a group with the ADD USER command, you implicitly create that group. The user who executes that ADD USER command becomes the group owner and can subsequently use GROUP commands to manage the group. For more information, see Section 7, Group Commands.

- The ADD USER command does not check for group ownership.

  If the group name and group number identify a group created explicitly with the ADD GROUP command, the group already has an owner. Group ownership is not relevant to the ADD USER command. However, only the group owner can use GROUP commands to manage the group.

- Implicitly created group names are treated as uppercase.

  When you implicitly create a group with the ADD USER command, the group name is not case-sensitive. It is assumed to contain uppercase alphabetic characters. Therefore, to manage that group with GROUP commands, you must specify the group name with capital letters in the GROUP commands.

- A new user’s primary group is set to the user’s administrative group.

  When you add a user, the administrative group for the user is also that user’s primary group. To change the primary group, use the ALTER USER command to alter the PRIMARY-GROUP attribute.

- PASSWORD-EXPIRES takes precedence over PASSWORD-MUST-CHANGE

  If the PASSWORD-EXPIRES and PASSWORD-MUST-CHANGE attributes are set in the same ADD command, the setting of the PASSWORD-EXPIRES attribute takes precedence over the PASSWORD-EXPIRES date calculated as a result of setting the PASSWORD-MUST-CHANGE attribute.

Examples

1. The group manager for a new marketing group (group name PRS and group number 86) uses this command to add the first member (other than the group manager) to the group:

   ```
   =ADD USER prs.darlene , 86,1 , PASSWORD market
   ```

   This command adds a user who has the user name PRS.DARLENE and the user ID 86,1. Darlene’s logon password is market. The other user attributes for PRS.DARLENE have their default values.

2. This command adds another member to the PRS group:

   ```
   =ADD USER prs.harry, 86,2 , PASSWORD SELLit
   ```

   The new user has the user name PRS.HARRY, the user ID 86,2, and the logon password SELLit.

3. Now the PRS group manager adds two more group members.

   First, the group manager uses the SET command to create a pattern of attribute values—a useful procedure for adding a number of users who share attributes:

   ```
   =ASSUME USER
   =SET USER-EXPIRES jun 26 2005, AUDIT-AUTHENTICATE-PASS all
   =SET PASSWORD-MUST-CHANGE EVERY 60 DAYS
   =SET GUARDIAN SECURITY nunu
   =SET TEXT-DESCRIPTION "Fred’s Group"
   ```

   These users must change their passwords every 60 days. Their ability to log on expires at midnight on June 26, 2005. All successful authentication attempts are audited by the Safeguard software, and the users are assigned the Guardian default security string NUNU. The text description identifies the group manager.

   Then the group manager issues the SHOW command to check that the user attributes were entered correctly:

   ```
   =SHOW
   ```

The report shows:

```
TYPE        OWNER
 USER       86,255

 PASSWORD =
 USER-EXPIRES               = 26JUN05,  0:00
 PASSWORD-EXPIRES           =    * NONE *
 PASSWORD-MUST-CHANGE EVERY =    60 DAYS
 PASSWORD-EXPIRY-GRACE      =    * NONE *
 GUARDIAN DEFAULT SECURITY  = NUNU
 GUARDIAN DEFAULT VOLUME    = $SYSTEM.NOSUBVOL

 AUDIT-AUTHENTICATE-PASS  = ALL          AUDIT-MANAGE-PASS  = NONE
 AUDIT-AUTHENTICATE-FAIL  = NONE         AUDIT-MANAGE-FAIL  = NONE
 AUDIT-USER-ACTION-PASS   = NONE
 AUDIT-USER-ACTION-FAIL   = NONE

 TEXT-DESCRIPTION = "Fred’s Group"

 CI-PROG = * NONE *
 CI-LIB  = * NONE *
 CI-NAME = * NONE *
 CI-SWAP = * NONE *
 CI-CPU  = ANY
 CI-PRI  = * NONE *
 CI-PARAM-TEXT =

 INITIAL-PROGTYPE      = PROGRAM
 INITIAL-PROGRAM       =
 INITIAL-DIRECTORY     =

SUBJECT DEFAULT-PROTECTION SECTION UNDEFINED!

SUBJECT OWNER-LIST SECTION UNDEFINED!
```

To add the two new users, the group manager uses the ADD command:

```
=ADD prs.mabel, 86,10, OWNER 86,2, OWNER-LIST 86,1, &
=PASSWORD SaleS
=ADD prs.jack, 86,8, OWNER 86,2, OWNER-LIST 86,1 &
=PASSWORD 4sale
```

PRS.MABEL has the user ID 86,10 and the password SaleS, and PRS.JACK has the user ID 86,8 and the password 4sale. The primary owner of the authentication records for both users is PRS.HARRY (user 86,2). The secondary owner for both records is PRS.DARLENE (86,1). The secondary owner has the same privileges as the records’ primary owner.

4. To add a fifth member, the PRS manager uses the LIKE clause with the ADD command:

```
=ADD USER prs.benny , 86,4, LIKE 86,10, PASSWORD perCent
```

This LIKE clause gives the new user (PRS.BENNY, who has the user ID 86,4) the same user attributes given to PRS.MABEL (user 86,10).

ALTER USER Command

ALTER USER changes one or more user attributes in a user's authentication record.

Only the primary owner and secondary owners of a user's authentication record, the primary owner's group manager, or the local super ID can use ALTER USER to change the user-attribute values in a user's authentication record.

For all attributes other than REMOTEPASSWORD, the ALTER USER command replaces the current attribute value with the newly specified value. For the REMOTEPASSWORD attribute, ALTER USER updates the remote password list by adding, deleting, or changing the corresponding remote password, as indicated.

```
ALTER USER { user-spec | ( user-spec [ , user-spec ] ... ) }

   [ , ] { LIKE user | user-attribute }

   [ , user-attribute ] ...
   [ [ , ] WHERE expression ]
```

`USER`

specifies USER as the object type of the ALTER command. Omit it if USER is the assumed type. (For more information about assumed types, see ASSUME Command on page 4-3.)

`user-spec`

specifies the user or users whose authentication records are to be changed. The *`user-spec`* variable can be any of:

```
group-num , member-num
group-name.member-name
group-num , *
*,*
```

The *`group-name`* and *`member-name`* variables can contain wild-card characters.

`LIKE` *`user`*

changes the user attributes of the user authentication record being altered to the same as some of the attribute values currently defined for the user identified by *`user`*. For *`user`*, specify one of:

```
alias
group-num,member-num
group-name.member-name
```

The *`alias`* variable is an existing user alias. For details, see Section 6, User Alias Security Commands.

LIKE changes the values of all user attributes except:

```
PASSWORD [password]
REMOTEPASSWORD \system-name remote-password
GUARDIAN [DEFAULT] SECURITY ["]string["]
GUARDIAN [DEFAULT] [SUB]VOLUME [\system.]$vol.subvol
CI-NAME [process-name]
PRIMARY-GROUP [ [ NAME ] group-name | NUMBER group-num ]
INITIAL-DIRECTORY [dir-path]
INITIAL-PROGRAM [prog-path]
INITIAL-PROGTYPE [prog-type]
TEXT-DESCRIPTION "[text]"
```

user-attribute

changes the current value of the specified user attribute. The user attributes are:

```
OWNER owner-id
OWNER-LIST [[-]user-list]
PASSWORD [password]
USER-EXPIRES [ date [ , time ] ]
PASSWORD-MUST-CHANGE [EVERY num DAYS]
PASSWORD-EXPIRY-GRACE [num [DAYS]]
PASSWORD-EXPIRES [ date [ , time] ]
AUDIT-AUTHENTICATE-PASS [audit-spec]
AUDIT-AUTHENTICATE-FAIL [audit-spec]
AUDIT-MANAGE-PASS [audit-spec]
AUDIT-MANAGE-FAIL [audit-spec]
AUDIT-USER-ACTION-PASS [audit-spec]
AUDIT-USER-ACTION-FAIL [audit-spec]
REMOTEPASSWORD [ \sys-name [ remote-password ] ]
DEFAULT-PROTECTION [ obj-attr ]
                   [ ( obj-attr [ , obj-attr ] ...) ]
GUARDIAN [DEFAULT] SECURITY ["]string["]
GUARDIAN [DEFAULT] [SUB]VOLUME [\system.]$vol.subvol
PRIMARY-GROUP [ [ NAME ] group-name | NUMBER group-num ]
INITIAL-DIRECTORY [dir-path]
INITIAL-PROGRAM [prog-path]
INITIAL-PROGTYPE [prog-type]
CI-PROG [prog-filename]
CI-LIB [lib-filename]
CI-CPU [cpu-number | ANY]
CI-NAME [process-name]
CI-SWAP [$vol.[subvol.filename]]
CI-PRI [priority]
CI-PARAM-TEXT [startup-param-text]
TEXT-DESCRIPTION "[text]"
WHERE expression
```

OWNER [*owner-id*]

transfers the primary ownership of a user’s authentication record to the user whose user ID is specified as *owner-id*. For *owner-id*, specify either:

```
[\*.]group-name.member-name
[\*.]group-num , member-num
```

If you omit *owner-id*, it is set to your user ID.

`OWNER-LIST [[-]`*`user-list`*`]`

changes the secondary ownership of a user's authentication record by adding or deleting owners in the owner list. A minus sign (-) preceding *user-list* indicates that the specified users are to be deleted from the existing owner list. If the minus sign is omitted, the specified users are added to the owner list. If *user-list* is omitted, the owner list is set to null (no secondary owners). A maximum of 50 users can be specified in *user-list*. For *user-list*, specify either:

```
net-user-spec
(net-user-spec [, net-user-spec ...])
```

net-user-spec is either:

```
[\node-spec.]group-name.member-name
[\node-spec.]group-num , member-num
```

node-spec is one of:

```
*
node-name
node-number
```

node-name

specifies the system name.

node-number

specifies the Expand node number.

group-name

specifies the name of any group.

group-num

specifies the group number of any group.

`PASSWORD [`*`password`*`]`

changes a user's logon password.

password

is a string of one to 64 characters. It can contain any alphanumeric characters except blanks, commas, semicolons, and the ASCII null character. The case of the letters is preserved. Lowercase letters remain lowercase, and uppercase remain uppercase.

If omitted, the value for *password* is set to null. In this case, a password is not required for the user to log on to the system.

The password is subject to the restrictions imposed by the configuration options described in Section 16, Safeguard Subsystem Commands.

▲ **WARNING.** Only the first eight characters of the password will be considered.

`USER-EXPIRES [ date [ , time] ]`

changes the user-expiration date to the specified date and time. (Both are local civil time.)

After USER-EXPIRES suspends a user’s ability to log on to the system, changing the USER-EXPIRES attribute to some future date restores that ability to log on.

If you omit both *date* and *time*, the user-expiration attribute value is set to null, and the user’s ability to log on to the system never expires.

If omitted, *time* is set to 0:00 (midnight).

The form of *date* [, *time*] is:

```
{ month-name day } year [,hour:min]
{ day month-name }
```

month-name

is the first three letters of the month name: JAN, FEB, MAR, and so on, using uppercase or lowercase letters.

day

is a 1-digit or 2-digit integer from 1 through 31, specifying the day of the month.

year

is a 4-digit integer specifying the year.

hour

is an integer from 0 through 23, specifying the hour.

min

is an integer from 0 through 59, specifying the minute.

`PASSWORD-MUST-CHANGE [EVERY` *`num`* `DAYS]`

changes the maximum number of days that a user can use the same password. For *num*, specify an integer from 1 through 32,767.

Changing the PASSWORD-MUST-CHANGE attribute causes the Safeguard software to calculate a new PASSWORD-EXPIRES date. The PASSWORD-EXPIRES date is set to the current date plus *num* days.

After PASSWORD-EXPIRES suspends a user’s ability to log on to the system, extending the user’s PASSWORD-MUST-CHANGE period can restore that ability. (For details on how the PASSWORD-MUST-CHANGE operation works, see SET USER Command on page 5-40.)

Setting the PASSWORD-EXPIRES attribute after setting the PASSWORD-MUST-CHANGE attribute causes the explicit setting of the PASSWORD-EXPIRES attribute to override the date previously calculated as a result of setting PASSWORD-MUST-CHANGE.

Omitting “EVERY *num* DAYS” disables the PASSWORD-MUST-CHANGE mechanism. That is, the user’s password never expires.

`PASSWORD-EXPIRY-GRACE [`*`num`*` [DAYS]]`

changes the number of days after password expiration during which users can change their passwords. For *num*, specify an integer from 0 through 32,767. If you omit *num*, the value for PASSWORD-EXPIRY-GRACE in the Safeguard configuration record is used. In this instance, the value *NONE* appears in this field of the user protection record.

`PASSWORD-EXPIRES [` *`date`* `[ ,` *`time`*`] ]`

changes the date and time after which the password expires. Specify *date* and *time* as local civil time.

If you omit both *date* and *time*, no expiration is set for the password.

If you omit only *time*, it is set to 0:00 (midnight).

The form of *date* [, *time*] is:

```
{ month-name day } year [,hour:min]
{ day month-name }
```

month-name

is the first three letters of the month name: JAN, FEB, MAR, and so on. (You can use either uppercase or lowercase letters.)

day

is a 1-digit or 2-digit integer from 1 to 31, specifying the day of the month.

year

is a 4-digit integer, specifying the year.

Note. The YEAR can take any value in the range of one minus the current year up to a maximum value of 9999. For example, If the current year is 2010, the YEAR field can take any value in the range 2009 to 9999.

hour

is an integer from 0 to 23, specifying the hour.

min

is an integer from 0 to 59, specifying the minute.

Setting the PASSWORD-MUST-CHANGE attribute after setting the PASSWORD-EXPIRES attribute causes the PASSWORD-EXPIRES date calculated as a result of setting PASSWORD-MUST-CHANGE to override the explicit setting of the PASSWORD-EXPIRES attribute.

AUDIT-AUTHENTICATE-PASS [*audit-spec*]

changes the *audit-spec* for successful user authentication attempts. The form of *audit-spec* is:

```
{ ALL | LOCAL | REMOTE | NONE }
```

For a description of the *audit-spec*s, see the SET USER Command on page 5-40. Omitting *audit-spec* specifies NONE.

AUDIT-AUTHENTICATE-FAIL [*audit-spec*]

changes the *audit-spec* for unsuccessful user authentication attempts. The form of *audit-spec* is:

```
{ ALL | LOCAL | REMOTE | NONE }
```

For a description of *audit-spec*, see the SET USER Command on page 5-40. Omitting *audit-spec* specifies NONE.

Note. In prior product versions of the Safeguard software, the AUDIT-AUTHENTICATE user attributes were called AUDIT-ACCESS. The user attribute name AUDIT-ACCESS is still supported, but HP discourages its use.

AUDIT-MANAGE-PASS [*audit-spec*]

changes the *audit-spec* for successful attempts to manage a user’s authentication record. The form of *audit-spec* is:

```
{ ALL | LOCAL | REMOTE | NONE }
```

For a description of the *audit-spec*s, see the SET USER Command on page 5-40. Omitting *audit-spec* specifies NONE.

AUDIT-MANAGE-FAIL [*audit-spec*]

changes the *audit-spec* for unsuccessful attempts to manage a user's authentication record. The form of *audit-spec* is:

```
{ ALL | LOCAL | REMOTE | NONE }
```

For a description of the *audit-spec*s, see the SET USER Command on page 5-40. Omitting *audit-spec* specifies NONE.

AUDIT-USER-ACTION-PASS [*audit-spec*]

changes the *audit-spec* for successful events performed by this user. The form of *audit-spec* is:

```
{ ALL | LOCAL | REMOTE | NONE }
```

For a description of the *audit-spec*s, see the SET USER Command on page 5-40. Omitting *audit-spec* specifies NONE.

AUDIT-USER-ACTION-FAIL [*audit-spec*]

changes the *audit-spec* for unsuccessful events attempted by this user. The form of *audit-spec* is:

```
{ ALL | LOCAL | REMOTE | NONE }
```

For a description of the *audit-spec*s, see the SET USER Command on page 5-40. Omitting *audit-spec* specifies NONE.

REMOTEPASSWORD [\system-name [*remote-password*]]

adds a new remote password, changes the remote password currently defined for a particular system, or deletes a remote password. A user can have zero, one, or many remote passwords (one for each remote system to which the user is granted access, and one for the local system matching that remote system).

Specifying \system-name without a *remote-password* deletes a user's remote password for the specified system.

△ **Caution.** Omitting both \system-name and *remote-password* deletes all the remote passwords currently defined for the user on this system.

\system-name

specifies the system for which a remote password is to be assigned. For \system-name, specify a valid system name.

remote-password

specifies a remote password to be associated with \system-name. For *remote-password*, specify a string from one to eight characters long. You can use any alphanumeric characters except blanks, commas, semicolons, and the ASCII null character. The case of the letters is preserved; lowercase letters remain lowercase, and uppercase letters remain uppercase. You cannot set multiple remote passwords with one command.

```
DEFAULT-PROTECTION [ obj-attr ]
                   [ ( obj-attr [ , obj-attr ] ...) ]
```

changes one or more attributes to be assigned immediately to new disk files created by processes with a PAID equal to this user ID. If `obj-attr` is omitted, new disk files remain under Guardian protection. If any `obj-attr` is specified, the attribute updates the current default protection record for the specified user ID.

obj-attr

is one of:

```
OWNER [ owner-id ]
ACCESS [ access-spec [ ; access-spec ] ... ]
AUDIT-ACCESS-PASS [ audit-spec ]
AUDIT-ACCESS-FAIL [ audit-spec ]
AUDIT-MANAGE-PASS [ audit-spec ]
AUDIT-MANAGE-FAIL [ audit-spec ]
```

For more information about these object attributes as they apply to disk files, see Section 8, Disk-File Security Commands.

```
GUARDIAN [DEFAULT] SECURITY ["]string["]
```

changes the Guardian default disk file security string for the user. The word DEFAULT is optional, as are the quotes that surround the security string specifier. You can include them in the command for readability. The `string` variable is a four-character string that specifies the Guardian default security string. Each position in the string can contain one of these characters: O, U, G, C, A, or N.

For more information about Guardian default file-security string, see the *Safeguard User's Guide*.

```
GUARDIAN [DEFAULT] [SUB]VOLUME [\system.]$vol.subvol
```

changes the Guardian default subvolume. The word DEFAULT and the prefix SUB are optional. You can include them in the command for readability. `\system` is also optional. If you omit `\system`, the current system is assumed. `$vol` specifies the user's default volume, and `subvol` specifies the default subvolume.

```
PRIMARY-GROUP [ [ NAME ] group-name | NUMBER group-num ]
```

changes the name or number of the primary group for the user. The user must already belong to this group. The word NAME is optional when you specify *group-name*. You can include it in the command for readability. *group-name* is the name of a group to which the user already belongs. *group-num* is the number of a group to which the user already belongs.

You can specify the primary group by group name or by group number, but not both. You cannot include PRIMARY-GROUP NAME and PRIMARY-GROUP NUMBER attributes in the same command. A PRIMARY-GROUP NAME replaces a previously specified PRIMARY-GROUP NUMBER, and vice versa.

The Safeguard software does not implicitly add this group to the user's group list if the user does not already belong to this group. The previous primary group remains on the user's group list, but not as the primary group.

Without *group-name* or *group-num*, PRIMARY GROUP clears the primary group setting, and the user's administrative group becomes the primary group. (See Considerations on page 5-21.)

`INITIAL-DIRECTORY [`*`dir-path`*`]`

changes the initial working directory within the OSS file system for the user. *dir-path* is a case-sensitive text string of up to 256 characters. It must be a syntactically valid OSS pathname. If you specify the INITIAL-DIRECTORY attribute, it must be the last attribute in the command string.

If you omit *dir-path*, the string is set to null (no pathname).

`INITIAL-PROGRAM [`*`prog-path`*`]`

changes the initial program pathname within the OSS environment for the user. *prog-path* is a case-sensitive text string of up to 256 characters. It must be a syntactically valid OSS pathname. If you specify the INITIAL-DIRECTORY attribute, it must be the last attribute in the command string.

If you omit *prog-path*, the string is set to null (no pathname).

`INITIAL-PROGTYPE [`*`prog-type`*`]`

changes the initial program type within the OSS environment for the user.

prog-type

is one of:

```
PROGRAM
SERVICE
WINDOW
```

If you omit *prog-type*, the initial program type is set to PROGRAM.

`CI-PROG [`*`prog-filename`*`]`

changes the command interpreter to be started after this user is authenticated at a Safeguard terminal. *prog-filename* is the name of the command interpreter's object file. It must be a local file name.

If you omit *prog-filename*, the other user attributes associated with CI-PROG *prog-filename* in this record are not meaningful.

`CI-LIB [`*`lib-filename`*`]`

changes the library file to be used with the command interpreter that is started when this user is authenticated at a Safeguard terminal. *lib-filename* must be a local file name.

If you omit *lib-filename*, no library file is used.

`CI-CPU [`*`cpu-number`*` | ANY]`

changes the number of the CPU in which the command interpreter is to run. If you specify *ANY*, any CPU is used.

If you omit *cpu-number*, any CPU is used.

`CI-NAME [`*`process-name`*`]`

changes the process name to be assigned to the command interpreter specified by CI-PROG. *process-name* must be a local process name.

If you omit *process-name*, the Safeguard software generates a process name.

`CI-SWAP [$`*`vol`*`[.`*`subvol`*`.`*`filename`*`]]`

changes the name of the volume or file to be used as the swap volume or file for the command interpreter. $*vol* must be a local volume name. You can optionally supply a subvolume name and file name.

If you omit $*vol*, the same volume that contains the CI-PROG object file is used.

`CI-PRI [`*`priority`*`]`

changes the priority at which the command interpreter is to run.

If you omit *priority*, the value of CI-PRI in the Safeguard configuration record is used.

`CI-PARAM-TEXT [`*`startup-param-text`*`]`

changes the data to be supplied as the startup message text for the command interpreter specified by CI-PROG. If you specify the CI-PARAM-TEXT attribute, it must be the last attribute in the command string.

If you omit *startup-param-text*, the string is set to null. (No text is supplied in the startup message.)

`TEXT-DESCRIPTION "[`*`text`*`]"`

specifies a string of characters to replace the existing text description for this record. Because SAFECOM allows a maximum command length of 528 characters, the specified text string must contain fewer than 528 characters. You can specify a longer descriptive text string by using the Safeguard SPI interface, as described in the *Safeguard Management Programming Manual*.

All text within the quotation marks is considered descriptive text.

If you specify `TEXT-DESCRIPTION` "" without any text between the quotation marks, the text description for this record is removed.

`WHERE` *`expression`*

causes the command to apply to only authentication records for users who belong to the groups specified by *`expression`*.

`expression` has the form:

`group` `[ {AND | OR}` *`group`* `] ...`

`group` is one of these:

```
GROUP [NAME]=group-name
GROUP NUMBER=group-num
PRIMARY-GROUP [NAME]=group-name
PRIMARY-GROUP NUMBER=group-num
```

Wild-card characters are not allowed in the group names or group numbers. Multiple groups within the *`expression`* can be enclosed within parentheses to change the order of evaluation of a complex expression. See Examples 4 and 5 on page 5-22.

`group-name` is case-sensitive. Therefore, you must enter alphabetic characters in an administrative group name in uppercase.

`RESET-TEXT-DESCRIPTION`

resets the text description field to a null value (no descriptive text).

Note. The RESET-TEXT-DESCRIPTION field is supported only on systems running G06.27 and later G-series RVUs and H06.06 and later H-series.

`RESET-BINARY-DESCRIPTION`

resets the binary description field to zero length and null values. For more information about the binary description field, see the *Safeguard Management Programming Manual*.

Note. The RESET-BINARY-DESCRIPTION field is supported only on systems running G06.27 and later G-series RVUs and H06.06 and later H-series.

```
RESET-STATIC-FAILED-LOGON-COUNT
```

resets the value of the attribute STATIC-FAILED-LOGON-COUNT to 0.

Note. The RESET-STATIC-FAILED-LOGON-COUNT field is supported only on systems running H06.10 and later H-series and G06.32 and later G-series RVUs.

Considerations

- Changing your logon password

  Only the owner of a user's authentication record or the owner's group manager can use the ALTER USER command to change a user's password. However, with the Guardian PASSWORD program, any users can change their own password. In addition, users can change their own passwords during logon.

  For example, this command changes the user's logon password to itsme:

  ```
  4> PASSWORD itsme
  ```

  When you change your logon password, your Safeguard authentication record is automatically updated.

  Your password is subject to restrictions defined by the configuration attributes described in Section 16, Safeguard Subsystem Commands.

- Adding or deleting default protection while a user is logged on

  If you add or delete default protection for a user while that user is logged on, and the user subsequently creates a disk file during that session, the FUP INFO and TACL FILEINFO displays are not updated until the next time the disk file's protection record is altered.

- Changing Guardian default security while a user is logged on

  If you change the Guardian default disk file security for a user while that user is logged on, the change does not take effect until the next time the user logs on or issues a Guardian VOLUME command.

- Changing a user's primary group

  Programmatic logon sets the group list of a process to contain the user's entire group list and also copies the user's primary group to the real group ID, effective group ID, and saved set-group-ID of the process. Because a user's primary group can differ from that user's administrative group, the effective group ID of a process can differ from the administrative group of the process as defined by the PAID.

Examples

1. The PRS group manager owns the authentication record for PRS.DARLENE. The manager enters the following command to transfer primary ownership of that

record to the user who has user ID 86,2 and to require that Darlene change her logon password every 35 days:

```
=ALTER USER prs.darlene, OWNER admin.sue, &
=PASSWORD-MUST-CHANGE EVERY 35 DAYS
```

Because the OWNER attribute for PRS.DARLENE was changed to a member of another group, PRS.MANAGER can no longer manage this authentication record.

2. The primary owner of the user authentication record for ACCTG.HARRY sets up Safeguard auditing for successful and failed authentication attempts (both local and remote) made under Harry's user name:

```
=ALTER USER acctg.harry, AUDIT-AUTHENTICATE-PASS all,&
=AUDIT-AUTHENTICATE-FAIL all
```

3. The primary owner of the user authentication record for PRS.BENNY alters the record so that the command interpreter EDITF starts automatically after the user logs on at a Safeguard terminal. The object program file for EDITF is $SALES.PROG2.EDITF:

```
=ALTER USER prs.benny, CI-PROG $sales.prog2.editf
```

4. The super ID alters the record of every user who belongs to the group AUDIT and has a primary group number of 254 or 255. The AUDIT-USER-ACTION-FAIL attribute is set to ALL for each of these users:

```
=ALTER USER *.*, AUDIT-USER-ACT-FAIL all, &
=WHERE GROUP=AUDIT AND (PRIMARY-GROUP NUMBER=254 &
=OR PRIMARY-GROUP NUMBER=255)
```

5. The group manager of group 12 alters the user authentication record of each user in group 12 who belongs to the group AUDIT3 or who belongs to group AUDIT4 and has a primary group number of 54. Each user in group 12 who meets this criteria is given a password expiration date of July 15, 2005.

```
=ALTER USER 12,*, PASSWORD-EXPIRES jul 15 2005, &
=WHERE GROUP=AUDIT3 OR &
=(GROUP=AUDIT4 AND PRIMARY-GROUP NUMBER=54)
```

DELETE USER Command

DELETE USER removes a user from a system and deletes the user's authentication record. After a user is deleted, the user cannot log on to the system.

The primary owner and secondary owners of a user's authentication record, the primary owner's group manager, and the super ID can delete a user.

```
DELETE USER { user-spec | ( user-spec [ , user-spec ] ... ) }

   [ [,] WHERE expression ]
```

`USER`

specifies USER as the object type of the DELETE command. Omit it if USER is the assumed object type. (For more information about assumed types, see the ASSUME Command on page 4-3.)

`user-spec`

specifies the user or users to be deleted from the system. *`user-spec`* can be any of:

```
group-num , member-num
group-name.member-name
group-num , *
*,*
```

`group-name` and *`member-name`* can contain wild-card characters.

`WHERE` *`expression`*

causes the DELETE command to apply to only authentication records for users who belong to the groups specified by *`expression`*. For a description of WHERE *`expression`*, see the ALTER USER Command on page 5-10.

Considerations

- Deleting a user authentication record owner

  If the primary owner of a user's authentication record is deleted, only the secondary record owners, the group manager of the primary owner, or the local super ID can change the user's authentication record.

- Effect of the DELETE USER command on access lists

  Deleting a user does not delete that user ID from any access control lists for objects protected by the Safeguard software. When a user is deleted, every object owner must remove all access-list entries that grant the deleted user access to protected objects.

- Effect of the DELETE USER command on file-sharing group lists

  Deleting a user causes that user ID to be deleted from all file-sharing group lists.

- Effect of aliases on deleting a user authentication record

  A user authentication record cannot be deleted if any user aliases are associated with the user ID. The alias authentication records must be deleted before the user authentication record can be deleted.

- Deleting the last user also deletes the group

  If you delete a user who is the only member of that administrative group, the group is deleted automatically if the group's AUTO-DELETE attribute is ON. The group name and group number then become available for use in defining a new group. If the group’s AUTO-DELETE attribute is OFF, you can delete the group only with the DELETE GROUP command. (For more information, see INFO GROUP Command on page 7-14.)

Examples

1. The group manager for the ACCTG group enters this command to delete the user ACCTG.HARRY:

```
=DELETE USER acctg.harry
```

2. The group manager for the PROG group enters this command to delete all users in the PROG group who are also members of the TEMP group:

```
=DELETE USER prog.*, WHERE GROUP=TEMP
```

FREEZE USER Command

FREEZE USER temporarily suspends a user’s ability to log on to the system. You can later restore this ability through the THAW USER command.

Only the owner of a user’s authentication record, the owner’s group manager, or the local super ID can freeze a user’s access to the system. Depending on the value of the Safeguard AUTHENTICATE-FAIL-FREEZE configuration option, a user ID can be automatically frozen. For details, see Section 16, Safeguard Subsystem Commands.

```
FREEZE USER { user-spec | ( user-spec [ , user-spec ] ... ) }

   [ [,] WHERE expression ]
```

`USER`

specifies USER as the object type of the FREEZE command. Omit it if USER is the assumed object type. (For more information about assumed types, see the ASSUME Command on page 4-3.)

`user-spec`

specifies the user or users whose ability to log on is to be frozen. *`user-spec`* can be any of:

```
group-num , member-num
group-name.member-name
group-num , *
*,*
```

`group-name` and *`member-name`* can contain wild-card characters.

`WHERE` *`expression`*

causes the FREEZE command to apply only to authentication records for users who belong to the groups specified by *`expression`*. For a description of WHERE *`expression`*, see the ALTER USER Command on page 5-10.

Considerations

- Freezing the super ID (255,255)

  The super ID can be frozen. The result of freezing the super ID is that this ID cannot log on. Freezing the super ID has no effect on any existing processes owned by the super ID, including logged-on TACLs.

- Freezing a user who is currently logged on

  Although a user can be frozen while logged on, freezing has no effect on the user's current command interpreter session. After logging off, however, the user cannot log on until the ability to log on is restored through the THAW USER command.

- Freezing a network user's local access.

  Freezing the local access of a network user serves only to prohibit the network user from logging on to your system. The network user can still access objects on your system from the remote system.

- Local super ID logging on as another frozen user ID

  Even if a user's ability to log on to the system is frozen, the local super ID (and the user's group manager) can log on as that user. By default, the local super ID can log on as any user defined for the system without supplying a password unless the configuration attribute PASSWORD-REQUIRED is set to ON (described in Section 16, Safeguard Subsystem Commands).

- Effect of freezing a user on user aliases

  Freezing a user authentication record has no effect on user aliases associated with the user ID. The user can still log on using an alias.

Examples

1. Either of these two FREEZE commands suspends the user who has the user name PRS.HARRY and the user ID 86,2:

   ```
   =FREEZE USER prs.harry
   ```

   ```
   =FREEZE USER 86,2
   ```

2. The owner of the user authentication records for the ACCTG group can freeze all users whose administrative group is ACCTG:

   ```
   =FREEZE USER acctg.*
   ```

INFO USER Command

INFO USER displays a report about the user-attribute values currently stored in a user’s authentication record.

Use of the INFO USER command is limited to these users:

- The user
- The primary and secondary owners of the user’s authentication record
- The primary owner’s group manager
- The super ID

```
INFO [ / OUT listfile / ] USER

   { user-spec | (  user-spec [ , user-spec ] ... ) }

   [ [ , ] option ] [ , option  ] ...
```

`OUT` *`listfile`*

directs SAFECOM output to *`listfile`* for the INFO report. (After executing the INFO command, SAFECOM redirects its output to the current OUT file.)

For *`listfile`*, specify any file name. SAFECOM opens *`listfile`* and appends the output text to it. If *`listfile`* does not exist, SAFECOM creates an EDIT-format file and writes the INFO report to that file.

`USER`

specifies USER as the object type of the INFO command. Omit it if USER is the assumed type. (For more information about assumed types, see the ASSUME Command on page 4-3.)

`user-spec`

specifies the user or users for whom an INFO report is to be produced.

`user-spec` can be any of:

```
group-num , member-num
group-name.member-name
group-num , *
*,*
```

`group-name` and *`member-name`* can contain wild-card characters.

`option`

is one of:

```
GENERAL
DETAIL
```

```
AUDIT
CI
OSS
REMOTEPASSWORD
DEFAULT-PROTECTION
GROUP
OWNER-LIST
ALIAS
TEXT-DESCRIPTION
WHERE expression
```

`GENERAL`

displays the basic user attributes, including password settings, user expiration, UID, Guardian security, and Guardian default volume.

`DETAIL`

displays all user attributes, including those displayed by all other options.

`AUDIT`

displays only attributes related to auditing.

`CI`

displays only attributes related to the default command interpreter.

`OSS`

displays only attributes related to OSS initial settings.

`REMOTEPASSWORD`

displays all remote passwords defined for this user.

`DEFAULT-PROTECTION`

displays only attributes related to default disk file protection.

`GROUP`

displays only the user’s group list and primary group.

`OWNER-LIST`

displays the secondary owners of the user’s authentication record.

`ALIAS`

displays all aliases defined for this user.

`TEXT-DESCRIPTION`

displays the descriptive text associated with the user’s authentication record.

WHERE *expression*

causes information to be displayed only for users who belong to the groups specified by *expression*. For a description of WHERE *expression*, see ALTER USER Command on page 5-10.

INFO USER Brief Report

Figure 5-1 shows the format of the brief INFO USER report. A description of the user-attribute values and status fields immediately follows it.

Figure 5-1. INFO USER Brief Report Format

```
GROUP.USER        USER-ID    OWNER      LAST-MODIFIED  LAST-LOGON      STATUS
user-name         u-id       o-id   [+] date,time      date,time       status
```

GROUP.USER
user-name

is the user name of the user whose current user attributes appear.

USER-ID
u-id

is the structured view of the user ID of the user whose current attributes appear.

OWNER
o-id

is the user ID of the user who is the primary owner of this user authentication record. If *o-id* is the network form of a user ID, the primary owner is a network user.

[+]

indicates the existence of an OWNER-LIST for the user authentication record. The + sign does not appear if no OWNER-LIST exists. You can specify the OWNER-LIST keyword to display the list of owners.

LAST-MODIFIED
date,time

is the time and date when this user authentication record was last changed (in local civil time).

LAST-LOGON
date,time

is the time and date when this user last logged on to the system (in local civil time).

STATUS
status

indicates this user’s current status. *status* can be any of:

USER-EXP The user’s ability to log on to the system has expired. Until the user’s USER-EXPIRES date is changed to some future date, the user cannot log on to the system.

PSWD-EXP The user’s password has expired. Until the user’s password is changed or until the user’s PASSWORD-MUST-CHANGE period is extended (through the ALTER USER command), the user cannot log on to the system. The PASSWORD-EXPIRES attribute can also be changed directly with the ALTER USER command.

FROZEN The user’s ability to log on to the system has been frozen. Until the owner of the user’s authentication record or the owner’s group manager restores this ability through the THAW USER command, the user cannot log on to the system.

THAWED The user can log on to the system.

The values of the *status* field are listed in the order of their priority. When two or more of the conditions described by a *status* value apply to a user, only the highest priority is displayed. For example, if a user’s password is expired and the user is frozen, *status* is displayed as PSWD-EXP.

INFO USER Detailed Report

Figure 5-2 on page 5-30 shows the format of the detailed INFO USER report.

Figure 5-2. INFO USER Detailed Report Format

```
GROUP.USER        USER-ID    OWNER     LAST-MODIFIED   LAST-LOGON    STATUS
user-name         u-id       o-id  [+] date, time      date, time    status

  UID                        = uid
  USER-EXPIRES               = date, time   [-- EXPIRED --]
  PASSWORD-EXPIRES           = date, time   [-- EXPIRED --]
  PASSWORD-MAY-CHANGE        = date, time   [-- EXPIRED --]
  PASSWORD-MUST-CHANGE EVERY = num  DAYS
  PASSWORD-EXPIRY-GRACE      = num  DAYS
  LAST-LOGON                 = date, time
  LAST-UNSUCESSFUL-ATTEMPT   = * NONE *
  LAST-MODIFIED              = date, time
  CREATION-TIME              = date, time
  FROZEN/THAWED              = FROZEN | THAWED
  STATIC FAILED LOGON COUNT  = count
  STATIC-FAILED-LOGON-RESET  = * NONE *
  GUARDIAN DEFAULT SECURITY  = string
  GUARDIAN DEFAULT VOLUME    = $vol.subvol

  CREATOR-USER-NAME          = user-name/alias-name
  CREATOR-USER-TYPE          = USER/ALIAS ( uid )
  CREATOR-NODENUMBER         = num


  AUDIT-AUTHENTICATE-PASS  = a-spec     AUDIT-MANAGE-PASS  = a-spec
  AUDIT-AUTHENTICATE-FAIL  = a-spec     AUDIT-MANAGE-FAIL  = a-spec
  AUDIT-USER-ACTION-PASS   = a-spec
  AUDIT-USER-ACTION-FAIL   = a-spec

  TEXT-DESCRIPTION = ["text"]

  BINARY-DESCRIPTION-LENGTH = length

  CI-PROG = [prog-filename]
  CI-LIB  = [lib-filename]
  CI-NAME = [process-name]
  CI-SWAP = [$vol[.subvol.filename]
  CI-CPU  = {num | ANY}
  CI-PRI  = [num]
  CI-PARAM-TEXT = [text]

  INITIAL-PROGTYPE      = prog-type
  INITIAL-PROGRAM       = [prog-path]
  INITIAL-DIRECTORY     = [dir-path]

  PRIMARY-GROUP  = group
  GROUP          = group

 [REMOTEPASSWORD = \system remotepassword]

 [ALIAS = alias]

SUBJECT DEFAULT PROTECTION SECTION

SUBJECT OWNER-LIST SECTION
```

In addition to the user attributes and status fields displayed in the brief INFO USER report, the detailed INFO USER report also displays these user attributes and status fields:

`UID =` *`uid`*

is the scalar view of this user’s user ID.

`USER-EXPIRES =` *`date, time`*

is the date and time when this user’s ability to log on to the system will be suspended (in local civil time). After the USER-EXPIRES command suspends a user’s ability to log on to the system, changing the user’s USER-EXPIRES attribute to some future date restores that ability.

`PASSWORD-EXPIRES =` *`date, time`*

is the date and time when this user’s password will expire. Whenever the user’s password is changed through the ALTER USER command, the Guardian PASSWORD program, or the Safeguard logon dialog, the Safeguard software calculates a new PASSWORD-EXPIRES date by adding the number of days specified in the user’s PASSWORD-MUST-CHANGE attribute to the date of the password change. The Safeguard software also calculates a new PASSWORD-EXPIRES date whenever the user’s PASSWORD-MUST-CHANGE attribute is changed.

Immediate expiration of the user’s password can also be specified with the PASSWORD-EXPIRES user attribute.

After a user’s password expires, the user cannot log on to the system until one of the following actions occurs: the user’s password is changed, the user’s PASSWORD-EXPIRES date is extended, or the user’s PASSWORD-MUST-CHANGE period is extended. (If the user has an extension period established with PASSWORD-EXPIRY-GRACE, that user can log on to change the expired password.)

`date` and *`time`* are given in local civil time.

`PASSWORD-MAY-CHANGE =`*`date, time`*

specifies the date and time after which users can change their password.

`PASSWORD-MUST-CHANGE EVERY =` *`num`* `DAYS`

specifies the maximum number of days that this user can use the same password.

`PASSWORD-EXPIRY-GRACE =` *`num`* `DAYS`

specifies the number of days after password expiration that users can change their password during logon.

`CREATION-TIME  = date, time`

specifies the date and time when the user was created.

`FROZEN/THAWED = `*`frozen`*` | `*`thawed`*

indicates whether or not a user’s access to the system has been frozen. While a user’s access to the system is frozen, the user cannot log on to the system.

`STATIC FAILED LOGON COUNT = `*`count`*

is the number of total unsuccessful logon attempts that made with this user’s user name since it was created. The maximum value for this attribute is 2,147,483,647. If the count exceeds beyond this value, an EMS will be generated and further increment will not happen until the count value is manually reset. This attribute can be reset from H06.10 and later H-series RVUs. The RESET of the attribute, STATIC FAILED LOGON COUNT, brings back the attribute's value to 0 and the attribute LAST-UNSUCCESSFUL-ATTEMPT is forced to a value NONE.

`STATIC-FAILED-LOGON-RESET = * NONE *`

specifies the last time when the value of the attribute, STATIC FAILED LOGON COUNT, was reset.

`GUARDIAN DEFAULT SECURITY = `*`string`*

is the Guardian default security string for this user.

`GUARDIAN DEFAULT VOLUME = $`*`vol.subvol`*

is the Guardian default subvolume for this user.

`CREATOR-USER-NAME = user-name/alias-name`

specifies the username of the user who created the user.

```
CREATOR-USER-TYPE          = USER/ALIAS ( uid )
```

identifies if the creator is an alias or a user, followed by the user ID of the creator.

```
CREATOR-NODENUMBER         = num
```

specifies the system number where the user is created.

```
AUDIT-AUTHENTICATE-PASS = a-spec   AUDIT-MANAGE-PASS = a-spec
AUDIT-AUTHENTICATE-FAIL = a-spec   AUDIT-MANAGE-FAIL = a-spec
AUDIT-USER-ACTION-PASS  = a-spec
AUDIT-USER-ACTION-FAIL  = a-spec
```

indicate the conditions under which the Safeguard software is to audit attempts to log on with this user’s user name, attempts to manage the user’s authentication record, and attempts by the user to perform an event. *`a-spec`* can be:

```
{ ALL | LOCAL | REMOTE | NONE }
```

For a full description, see *`audit-spec`* for SET USER Command on page 5-40.

`TEXT-DESCRIPTION = [ "`*`text`*`" ]`

is the descriptive text associated with the user authentication record.

`BINARY-DESCRIPTION-LENGTH = `*`length`*

is the length in bytes of the binary description field for the user authentication record. If no binary description was specified for the record, *`length`* is 0. For more information about the binary description field, see the *Safeguard Management Programming Manual*.

`CI-PROG = [ `*`prog-filename`*` ]`

is the object file name of the command interpreter started after the user logs on at a Safeguard terminal.

`CI-LIB = [ `*`lib-filename`*` ]`

is the file name of the library file used with the command interpreter.

`CI-NAME = [ `*`process-name`*` ]`

is the process name assigned to the command interpreter.

`CI-SWAP = [ $`*`vol`*`[`*`.subvol.filename`*`] ]`

is the swap volume or file used with the command interpreter.

`CI-CPU = `*`num`*` | ANY`

is the number of the CPU in which the command interpreter runs. ANY indicates any CPU.

`CI-PRI = [ `*`num`*` ]`

is the priority at which the command interpreter runs.

`CI-PARAM-TEXT = [ `*`text`*` ]`

is the startup parameter text supplied to the command interpreter. It is blank if no text is specified.

`INITIAL-PROGTYPE = `*`prog-type`*

is the initial program type: PROGRAM, WINDOW, or SERVICE.

`INITIAL-PROGRAM = [ `*`prog-path`*` ]`

is the initial program pathname for the OSS file system. It is blank if no pathname is specified.

`INITIAL-DIRECTORY = [ `*`dir-path`*` ]`

is the initial directory pathname for OSS. It is blank if no pathname is defined.

`PRIMARY-GROUP =` *`group`*

is the group name of the user’s primary group.

`GROUP =` *`group`*

is the group name of each group in the user’s group list. The user’s administrative group always appears in the group list. Other groups are those specified by the MEMBER attribute of the ADD or ALTER GROUP commands.

`[REMOTEPASSWORD = \`*`system-name remotepassword`*` ]`

is a remote password defined for the specified system name.

When \*`system-name`* is displayed as \\???????, the remote password is defined for a system number that is no longer assigned to a system on the network.

When *`remotepassword`* is displayed as ;;;;;;;;, the remote password contains one or more characters that are not letters or digits.

If no remote passwords are defined, REMOTEPASSWORD does not appear.

`[ALIAS =` *`alias`*`]`

is a user alias assigned to this user ID. This field does not appear if the user has no aliases.

`SUBJECT DEFAULT PROTECTION SECTION`

shows the default protection assigned to the user’s disk files when they are added to the Safeguard database.

`SUBJECT OWNER-LIST SECTION`

lists the secondary owners of the user’s authentication record.

Examples

1. This example of the INFO USER command displays the user attributes for user PRS.HARRY before and after he is frozen:

   ```
   =INFO USER prs.harry
   ```

```
GROUP.USER    USER-ID    OWNER     LAST-MODIFIED    LAST-LOGON     STATUS
PRS.HARRY      86,2       86,255  23MAY05, 15:43   28MAY05, 9:22   THAWED
```

```
=FREEZE USER prs.harry
=INFO USER prs.harry
```

```
GROUP.USER    USER-ID    OWNER     LAST-MODIFIED    LAST-LOGON     STATUS
PRS.HARRY     86,2       86,255    29MAY05,  8:23   28MAY05, 9:22  FROZEN
```

2. This command displays the group list for each user in administrative group 255 who also belongs to group SECURE:

```
=INFO USER 255,*, GROUP, WHERE GROUP=SECURE
```

RESET USER Command

RESET USER resets the current default user-attribute values to predefined values. (The predefined reset values are the values of the default user attributes when you begin a SAFECOM session.)

When you add a user to your system, the current default user attribute values are used for any user attributes you do not specify in the ADD USER command. (Use the SET USER command to set the predefined default values to specific values for the user attributes.)

```
RESET USER [ [ , ] user-attribute-keyword ]

   [ , user-attribute-keyword ] ...
```

`RESET USER`

entered with no *`user-attribute-keyword`*, resets all current default user attribute values to their predefined values.

`USER`

specifies USER as the object type of the RESET command. Omit it if USER is the assumed type. (For more information, see the ASSUME Command on page 4-3.)

`user-attribute-keyword`

sets the current default value of the specified user attribute to a predefined value. The *`user-attribute-keyword`* can be:

```
OWNER
OWNER-LIST
PASSWORD
USER-EXPIRES
PASSWORD-MUST-CHANGE
PASSWORD-EXPIRY-GRACE
PASSWORD-EXPIRES
AUDIT-AUTHENTICATE-PASS
AUDIT-AUTHENTICATE-FAIL
AUDIT-MANAGE-PASS
AUDIT-MANAGE-FAIL
AUDIT-USER-ACTION-PASS
AUDIT-USER-ACTION-FAIL
TEXT-DESCRIPTION
REMOTEPASSWORD
DEFAULT-PROTECTION
GUARDIAN [DEFAULT] SECURITY
GUARDIAN [DEFAULT] [SUB]VOLUME
```

```
INITIAL-DIRECTORY
INITIAL-PROGRAM
INITIAL-PROGTYPE
CI-PROG
CI-LIB
CI-CPU
CI-NAME
CI-SWAP
CI-PRI
CI-PARAM-TEXT
```

The predefined values for the user attributes are:

`OWNER`

`owner-id` is set to the user ID of the current SAFECOM user.

`OWNER-LIST`

`user-list` is set to null (no secondary owners).

`PASSWORD`

`password` is set to null. (No password is required to log on).

`USER-EXPIRES`

`date`,*`time`* are set to null (no expiration date).

`PASSWORD-MUST-CHANGE`

`num` days is set to null. (The user never has to change the password.)

`PASSWORD-EXPIRY-GRACE`

`num` days is set to null. (The user has no extension period during which to change the expired password).

`PASSWORD-EXPIRES`

`date`,*`time`* are set to null (no expiration date).

`AUDIT-AUTHENTICATE-PASS`

`audit-spec` is set to NONE.

`AUDIT-AUTHENTICATE-FAIL`

`audit-spec` is set to NONE.

`AUDIT-MANAGE-PASS`

`audit-spec` is set to NONE.

AUDIT-MANAGE-FAIL

audit-spec is set to NONE.

AUDIT-USER-ACTION-PASS

audit-spec is set to NONE.

AUDIT-USER-ACTION-FAIL

audit-spec is set to NONE.

TEXT-DESCRIPTION

text is set to null (no description text).

REMOTEPASSWORD

The remote password list is set to null (no remote passwords).

DEFAULT-PROTECTION

The default protection record is set to null. (New files remain under Guardian protection until explicitly added to the Safeguard database.)

GUARDIAN [DEFAULT] SECURITY

string is set to “OOOO.”

GUARDIAN [DEFAULT] [SUB]VOLUME

$*vol.subvol* is set to $SYSTEM.NOSUBVOL.

INITIAL-DIRECTORY

dir-path is set to null (no pathname).

INITIAL-PROGRAM

prog-path is set to null (no pathname).

INITIAL-PROGTYPE

prog-type is set to PROGRAM.

CI-PROG

prog-filename is set to null. (No command interpreter is in this user record.)

CI-LIB

lib-filename is set to null (no library file).

`CI-CPU`

`cpu-number` is set to ANY.

`CI-NAME`

`process-name` is set to null. (The Safeguard software generates a name.)

`CI-SWAP`

`$`*`vol`* is set to null. (Use same volume as CI-PROG object file.)

`CI-PRI`

`priority` is set to null. (Use the value of CI-PRI in the Safeguard configuration record.)

`CI-PARAM-TEXT`

`startup-param-text` is set to null. (No data is supplied in startup message text.)

Considerations

- Specifying an attribute name without a value in an ADD or ALTER command causes the attribute to be assigned the predefined default value (as defined for RESET USER Command on page 5-35).
- Executing RESET USER without specifying a *`user-attribute-keyword`*

  If you enter RESET USER (or RESET when the assumed object type is USER) and you do not include a *`user-attribute-keyword`*, all the user attributes are returned to their predefined values. The predefined values are listed earlier in the syntax description.

Examples

The PRS group manager wants to restore the current default user attributes (set in previous SET USER commands) to their predefined values. First, the manager enters the SHOW USER command, which displays the current user attributes:

```
=SHOW USER
```

The report shows:

```
TYPE        OWNER
 USER       86,2

  PASSWORD =
  USER-EXPIRES               =    * NONE *
  PASSWORD-EXPIRES           =    * NONE *
  PASSWORD-MUST-CHANGE EVERY =    30 DAYS
  PASSWORD-EXPIRY-GRACE      =    * NONE *
  GUARDIAN DEFAULT SECURITY  = NUNU
  GUARDIAN DEFAULT VOLUME    = $DATA2.FRED

  AUDIT-AUTHENTICATE-PASS  = ALL          AUDIT-MANAGE-PASS  = REMOTE
  AUDIT-AUTHENTICATE-FAIL  = ALL          AUDIT-MANAGE-FAIL  = NONE
  AUDIT-USER-ACTION-PASS   = NONE
  AUDIT-USER-ACTION-FAIL  = NONE

  TEXT-DESCRIPTION = "Fred’s Group"

  CI-PROG = * NONE *
  CI-LIB  = * NONE *
  CI-NAME = * NONE *
  CI-SWAP = * NONE *
  CI-CPU  = ANY
  CI-PRI  = * NONE *
  CI-PARAM-TEXT =

  INITIAL-PROGTYPE      = PROGRAM
  INITIAL-PROGRAM       =
  INITIAL-DIRECTORY     =

 SUBJECT DEFAULT-PROTECTION SECTION UNDEFINED!

 SUBJECT OWNER-LIST SECTION UNDEFINED!
```

Then the manager enters the following RESET command:

```
=RESET USER
```

Finally, the manager enters another SHOW USER command to display the attributes that have been reset:

```
=SHOW USER
```

The report shows:

```
 TYPE        OWNER
  USER       86,255

   PASSWORD =
   USER-EXPIRES               =    * NONE *
   PASSWORD-EXPIRES           =    * NONE *
   PASSWORD-MUST-CHANGE EVERY =    * NONE *
   PASSWORD-EXPIRY-GRACE      =    * NONE *
   GUARDIAN DEFAULT SECURITY  = 0000
   GUARDIAN DEFAULT VOLUME    = $SYSTEM.NOSUBVOL

   AUDIT-AUTHENTICATE-PASS  = NONE         AUDIT-MANAGE-PASS  = NONE
   AUDIT-AUTHENTICATE-FAIL  = NONE         AUDIT-MANAGE-FAIL  = NONE
   AUDIT-USER-ACTION-PASS   = NONE
   AUDIT-USER-ACTION-FAIL   = NONE

   TEXT-DESCRIPTION =

   CI-PROG = * NONE *
   CI-LIB  = * NONE *
   CI-NAME = * NONE *
   CI-SWAP = * NONE *
   CI-CPU  = ANY
   CI-PRI  = * NONE *
   CI-PARAM-TEXT =

   INITIAL-PROGTYPE      = PROGRAM
   INITIAL-PROGRAM       =
   INITIAL-DIRECTORY     =

  SUBJECT DEFAULT-PROTECTION SECTION UNDEFINED!

  SUBJECT OWNER-LIST SECTION UNDEFINED!
```

SET USER Command

SET USER establishes current default values for one or more user attributes. Later, when you add a user to your system, use these current default user-attribute values are used for any attribute you do not specify in the ADD USER command.

After you set the current default user-attribute values, use the SHOW USER command to display the current default user-attribute values before you add a user through ADD USER.

```
SET USER [ , ] { LIKE user | user-attribute }

   [ , user-attribute ] ...
```

`USER`

specifies USER as the object type of the SET command. Omit it if USER is the assumed type. (For more information, see the ASSUME Command on page 4-3.)

LIKE *user*

sets some of the current default user attribute values to the same as those currently defined for the user or alias specified with *user*. *user* is one of the following:

```
group-num,member-num
group-name.member-name
alias
```

LIKE sets the current default values for all user attributes except:

```
PASSWORD [password]
REMOTEPASSWORD \system-name remote-password
GUARDIAN [DEFAULT] SECURITY ["]string["]
GUARDIAN [DEFAULT] [SUB]VOLUME [\system.]$vol.subvol
CI-NAME [process-name]
PRIMARY-GROUP [ [ NAME ] group-name | NUMBER group-num ]
INITIAL-DIRECTORY [dir-path]
INITIAL-PROGRAM [prog-path]
INITIAL-PROGTYPE [prog-type]
TEXT-DESCRIPTION "[text]"
```

user-attribute

establishes a current default user-attribute value to be used in subsequent ADD USER commands. The user attributes are:

```
OWNER [owner-id]
OWNER-LIST [[-]owner-list]
PASSWORD [password]
USER-EXPIRES [ date [ , time] ]
PASSWORD-MUST-CHANGE [EVERY num DAYS]
PASSWORD-EXPIRY-GRACE [num [DAYS]]
PASSWORD-EXPIRES [ date [ , time] ]
AUDIT-AUTHENTICATE-PASS [audit-spec]
AUDIT-AUTHENTICATE-FAIL [audit-spec]
AUDIT-MANAGE-PASS [audit-spec]
AUDIT-MANAGE-FAIL [audit-spec]
AUDIT-USER-ACTION-PASS [audit-spec]
AUDIT-USER-ACTION-FAIL [audit-spec]
TEXT-DESCRIPTION "[text]"
REMOTEPASSWORD \system-name remote-password
DEFAULT-PROTECTION [ obj-attr ]
                   [ ( obj-attr [ , obj-attr ] ...) ]
GUARDIAN [DEFAULT] SECURITY ["]string["]
GUARDIAN [DEFAULT] [SUB]VOLUME [\system.]$vol.subvol
INITIAL-DIRECTORY [dir-path]
INITIAL-PROGRAM [prog-path]
INITIAL-PROGTYPE [prog-type]
CI-PROG [prog-filename]
CI-LIB [lib-filename]
CI-CPU [cpu-number | ANY]
CI-NAME [process-name]
CI-SWAP [$vol.[subvol.filename]]
CI-PRI [priority]
```

CI-PARAM-TEXT [*startup-param-text*]

OWNER [*owner-id*]

specifies the owner of a user authentication record. For *owner-id*, specify either of:

[*.]*group-name*.*member-name*
[*.]*group-num , member-num*

If you omit *owner-id*, your user ID becomes the current *owner-id*.

OWNER-LIST [[-]*user-list*]

changes the secondary ownership of a user's authentication record by adding or deleting owners in the owner list. A minus sign (-) preceding *user-list* indicates that the specified users are to be deleted from the existing owner list. If the minus sign is omitted, the specified users are added to the owner list. If you omit *user-list*, the owner list is set to null (no secondary owners). A maximum of 50 users can be specified in *user-list*. For *user-list*, specify either:

net-user-spec
(*net-user-spec* [, *net-user-spec* ...])

net-user-spec is either:

[\node-spec.]*group-name*.*member-name*
[\node-spec.]*group-num , member-num*

node-spec is one of:

*
node-name
node-number

node-name

specifies the system name.

node-number

specifies the Expand node number.

group-name

specifies the name of any group.

group-num

specifies the group number of any group.

`PASSWORD [`*`password`*`]`

specifies a logon password for a user. Typically, users must enter their user name and a password to log on to a system.

For *`password`*, specify the user’s logon password, which can be one to eight characters long. Use any alphanumeric characters except blanks, commas, semicolons, and the ASCII null character. The case of letters in a password is preserved. Lowercase letters remain lowercase, and uppercase letters remain uppercase.

If you omit *`password`*, its value is set to null. (No password is required for logon.)

`USER-EXPIRES [ `*`date`*` [ , `*`time`*`] ]`

establishes a date and time after which a user cannot log on to the system. Specify *`date`* and *`time`* in local civil time.

If you omit both *`date`* and *`time`*, the user-expiration attribute value is set to null (no expiration date).

If omitted, *`time`* is set to 0:00 (midnight).

The form of *`date`* [, *`time`*] is:

```
{ month-name day } year [,hour:min]
{ day month-name }
```

`month-name`

is the first three letters of the month name: JAN, FEB, MAR, and so on. (You can use either uppercase or lowercase letters.)

`day`

is a 1-digit or 2-digit integer from 1 through 31.

`year`

is a 4-digit integer.

`hour`

is an integer from 0 through 23.

`min`

is an integer from 0 through 59.

`PASSWORD-MUST-CHANGE [EVERY `*`num`*` DAYS]`

specifies the maximum number of days that a user can use the same password. For *`num`*, specify an integer from 1 to 32,767.

When you add a user with a PASSWORD-MUST-CHANGE attribute, the Safeguard software calculates a PASSWORD-EXPIRES date by adding *num* days to the current date. If the user's password is not changed before the PASSWORD-EXPIRES date, the user cannot log on to the system after that date (unless a PASSWORD-EXPIRY-GRACE period has been established).

Each time the password is changed, the Safeguard software calculates a new PASSWORD-EXPIRES date by adding *num* days to the date of the password change.

Omitting the EVERY *num* DAYS clause disables PASSWORD-MUST-CHANGE. (That is, the user's password never expires unless the PASSWORD-EXPIRES attribute is set.)

`PASSWORD-EXPIRY-GRACE` *`num`* `[DAYS]`

specifies the number of days after password expiration during which users can change their password during logon. For *num*, specify an integer from 0 to 32,767. A value of 0 means no grace period.

Omitting *num* specifies that the value of PASSWORD-EXPIRY-GRACE in the Safeguard configuration record is used to determine the user's extension period. In this instance, the value *NONE* appears in this field of the user protection record.

`PASSWORD-EXPIRES [` *`date`* `[ ,` *`time`*`] ]`

establishes a date and time after which a user's password expires. Specify *date* and *time* in local civil time.

If you omit both *date* and *time*, no expiration is set for the password. (However, an expiration date is calculated and set if a PASSWORD-MUST-CHANGE period is subsequently specified or altered.)

If you omit only *time*, it is set to 0:00 (midnight).

The form of *date* [, *time*] is:

```
{ month-name day } year [,hour:min]
{ day month-name }
```

month-name

is the first three letters of the month name: JAN, FEB, MAR, and so on. (You can use either uppercase or lowercase letters.)

day

is a 1-digit or 2-digit integer from 1 through 31.

year

is a 4-digit integer.

`hour`

is an integer from 0 through 23.

`min`

is an integer from 0 through 59.

`AUDIT-AUTHENTICATE-PASS [`*`audit-spec`*`]`

establishes an *`audit-spec`* for successful user authentication attempts. The *`audit-spec`* specifies the conditions under which the Safeguard software writes an audit record to the audit file when the user successfully logs on to the system.

The form of *`audit-spec`* is:

```
{ ALL | LOCAL | REMOTE | NONE }
```

`ALL`

All successful logons are audited.

`LOCAL`

Only successful logons from the local system are audited.

`REMOTE`

This form has no effect. Remote authentication is not supported.

`NONE`

No successful logons are audited.

Omitting *`audit-spec`* specifies NONE.

Note. In prior product versions of the Safeguard software, the AUDIT-AUTHENTICATE user attributes were called AUDIT-ACCESS. The user attribute name AUDIT-ACCESS is still supported, but HP discourages its use.

`AUDIT-AUTHENTICATE-FAIL [`*`audit-spec`*`]`

establishes an *`audit-spec`* for unsuccessful user authentication attempts. The *`audit-spec`* specifies the conditions under which an audit record is written to the audit file if the user fails to log on properly.

The form of *`audit-spec`* is:

```
{ ALL | LOCAL | REMOTE | NONE }
```

`ALL`

All unsuccessful logons are audited.

`LOCAL`

Only unsuccessful logons from the local system are audited.

`REMOTE`

This form has no effect. Remote authentication is not supported.

`NONE`

No unsuccessful logons are audited.

Omitting *`audit-spec`* specifies NONE.

`AUDIT-MANAGE-PASS [`*`audit-spec`*`]`

establishes an *`audit-spec`* for successful attempts to manage a user's authentication record. The *`audit-spec`* specifies the conditions under which an audit record is written to the audit file when the user's authentication record is managed.

The form of *`audit-spec`* is:

```
{ ALL | LOCAL | REMOTE | NONE }
```

`ALL`

All successful management attempts are audited.

`LOCAL`

Only successful management attempts from the local system are audited.

`REMOTE`

Only successful management attempts from remote systems are audited.

`NONE`

No successful management attempts are audited.

Omitting *`audit-spec`* specifies NONE.

`AUDIT-MANAGE-FAIL [`*`audit-spec`*`]`

establishes an *`audit-spec`* for unsuccessful attempts to manage a user's authentication record. The *`audit-spec`* specifies the conditions under which an audit record is written to the audit file when somebody tries, but fails, to manage the user's authentication record.

The form of *`audit-spec`* is:

```
{ ALL | LOCAL | REMOTE | NONE }
```

`ALL`

All unsuccessful management attempts are audited.

`LOCAL`

Only unsuccessful management attempts from the local system are audited.

`REMOTE`

Only unsuccessful management attempts from a remote system are audited.

`NONE`

No unsuccessful management attempts are audited.

Omitting *`audit-spec`* specifies NONE.

`AUDIT-USER-ACTION-PASS [`*`audit-spec`*`]`

establishes an *`audit-spec`* for successful events performed by this user, including attempts to access objects and attempts to create or manage Safeguard protection records. The *`audit-spec`* specifies the conditions under which the Safeguard software writes an audit record to the audit file when the user successfully performs an event.

Note. When the Safeguard global configuration attributes AUDIT-CLIENT-OSS and AUDIT-OSS-FILTER are enabled, the AUDIT-USER-ACTION-PASS attribute takes effect for OSS auditing**.**

The form of *`audit-spec`* is:

```
{ ALL | LOCAL | REMOTE | NONE }
```

`ALL`

All successful events are audited.

`LOCAL`

Only successful events on the local system are audited.

`REMOTE`

Only successful events by a remote user are audited.

`NONE`

No successful events are audited.

Omitting *`audit-spec`* specifies NONE.

`AUDIT-USER-ACTION-FAIL [`*`audit-spec`*`]`

establishes an *`audit-spec`* for unsuccessful events attempted by this user, including attempts to access objects and attempts to create or manage Safeguard protection records. The *`audit-spec`* specifies the conditions under which the Safeguard software writes an audit record to the audit file when the user unsuccessfully attempts to perform an event.

Note. When the Safeguard global configuration attributes AUDIT-CLIENT-OSS and AUDIT-OSS-FILTER are enabled, the AUDIT-USER-ACTION-PASS attribute takes effect for OSS auditing.

The form of *`audit-spec`* is:

```
{ ALL | LOCAL | REMOTE | NONE }
```

`ALL`

All unsuccessful events are audited.

`LOCAL`

Only unsuccessful events on the local system are audited.

`REMOTE`

Only unsuccessful events by a remote user are audited.

`NONE`

No unsuccessful events are audited.

Omitting *`audit-spec`* specifies NONE.

`TEXT-DESCRIPTION "[`*`text`*`]"`

specifies a string of descriptive text to be associated with the user record. The text must consist of printable characters. This attribute is provided for documentation purposes only and has no effect on the user record. All text between the quotation marks is considered to be descriptive text.

Because SAFECOM allows a maximum command length of 528 characters, the specified string must contain fewer than 528 characters. You can specify a longer descriptive text string by using the Safeguard SPI interface, as described in the *Safeguard Management Programming Manual*.

If you omit *`text`*, no descriptive text is included in the user authentication record.

`REMOTEPASSWORD \`*`system-name remote-password`*

establishes a remote password for a local user ID.

`\system-name`

is the system for which the remote password is to be assigned. The `\system-name` value must be a valid system name.

`remote-password`

is the remote password assigned to `\system-name`. For `remote-password`, specify a string of one to eight characters. You can use any character in a remote password except blanks, commas, semicolons, and the ASCII null character. The case of letters is preserved. Lowercase letters remain lowercase, and uppercase letters remain uppercase. Only one remote password can be set with a SET command.

Note. Use RESET USER REMOTEPASSWORD to clear a default remote password that you have previously established with the SET command.

```
DEFAULT-PROTECTION [ obj-attr ]
                   [ ( obj-attr [ ,obj-attr ] ...) ]
```

specifies one or more attributes to be assigned immediately to new disk files created by processes with a PAID equal to the user ID. If `obj-attr` is omitted, new disk files remain under Guardian protection. If any `obj-attr` is specified, the attribute updates the current default protection record.

`obj-attr`

is one of:

```
OWNER [ owner-id ]
ACCESS [ access-spec [ ; access-spec ] ... ]
AUDIT-ACCESS-PASS [ audit-spec ]
AUDIT-ACCESS-FAIL [ audit-spec ]
AUDIT-MANAGE-PASS [ audit-spec ]
AUDIT-MANAGE-FAIL [ audit-spec ]
```

For more information about these object attributes as they apply to disk files, see Section 8, Disk-File Security Commands.

```
GUARDIAN [DEFAULT] SECURITY ["]string["]
```

specifies the Guardian default disk file security string for the user. The word DEFAULT is optional, as are the quotes that surround the security string specifier. You can include them in the command for readability. `string` is a four-character string that specifies the Guardian default security string. Each position in the string can contain one of these characters: O, U, G, C, A, or N.

If no GUARDIAN SECURITY is specified, the default Guardian security string is set to “OOOO.”

For more information about Guardian default file-security string, see the *Safeguard User’s Guide*.

`GUARDIAN [DEFAULT] [SUB]VOLUME [\`*`system`*`.]$`*`vol`*`.`*`subvol`*

specifies the Guardian default subvolume. The word DEFAULT and the prefix SUB are optional. You can include them in the command for readability. `\`*`system`* is also optional. If you omit `\`*`system`*, the current system is assumed. `$`*`vol`* specifies the user's default volume, and *`subvol`* specifies the default subvolume.

If no GUARDIAN VOLUME is specified, the default subvolume is set to $SYSTEM.NOSUBVOL.

`INITIAL-DIRECTORY [`*`dir-path`*`]`

specifies the initial working directory within the OSS file system for the user. *`dir-path`* is a case-sensitive text string of up to 256 characters. It must be a syntactically valid OSS pathname. If you specify the INITIAL-DIRECTORY attribute, it must be the last attribute in the command string.

If you omit *`dir-path`*, no pathname is used.

`INITIAL-PROGRAM [`*`prog-path`*`]`

specifies the initial program pathname within the OSS environment for the user. *`prog-path`* is a case-sensitive text string of up to 256 characters. It must be a syntactically valid OSS pathname. If you specify the INITIAL-PROGRAM attribute, it must be the last attribute in the command string.

If you omit *`prog-path`*, no pathname is used.

`INITIAL-PROGTYPE [`*`prog-type`*`]`

specifies the initial program type within the OSS environment for the user.

`prog-type`

is one of these:

```
PROGRAM
SERVICE
WINDOW
```

If you omit *`prog-type`*, PROGRAM is used.

This feature is not currently implemented on NonStop systems. It is reserved for future use.

`CI-PROG [`*`prog-filename`*`]`

specifies the command interpreter to be started after this user is authenticated at a Safeguard terminal. *`prog-filename`* is the name of the command interpreter's object file. It must be a local file name.

If you omit *`prog-filename`*, the other user attributes associated with CI-PROG in this record are not meaningful.

If you omit CI-PROG *`prog-filename`* in the user authentication record, the Safeguard software starts the PROG (with associated parameters) in the definition record for the terminal at which the user logs on. If you do not specify PROG in the terminal definition record, the Safeguard software starts the CI-PROG (with associated parameters) specified in the Safeguard configuration record.

`CI-LIB [`*`lib-filename`*`]`

specifies the library file to be used with the command interpreter started when this user is authenticated at a Safeguard terminal. *`lib-filename`* must be a local file name.

If you omit *`lib-filename`*, no library file is used.

`CI-CPU [`*`cpu-number`*` | ANY]`

specifies the number of the CPU in which the command interpreter is to run. If you specify *`ANY`*, any CPU is used.

If you do not specify *`cpu-number`*, any CPU is used.

`CI-NAME [`*`process-name`*`]`

specifies the process name to be assigned to the command interpreter specified by CI-PROG.

If you omit *`process-name`*, Safeguard assigns a name to it. *`process-name`* must be a local process name.

`CI-SWAP [$`*`vol.`*`[`*`subvol.filename`*`]]`

specifies the name of the volume or file to be used as the swap volume or file for the command interpreter. `$`*`vol`* must be a local volume name. You can optionally supply a subvolume name and file name.

If you omit `$`*`vol`*, the same volume that contains the CI-PROG object file is used.

`CI-PRI [`*`priority`*`]`

specifies the priority at which the command interpreter is to run.

If you omit *`priority`*, the value of CI-PRI in the Safeguard configuration record is used.

`CI-PARAM-TEXT [`*`startup-param-text`*`]`

specifies the data to be supplied as the startup message text for the command interpreter specified by CI-PROG. If you specify the CI-PARAM-TEXT attribute, it must be the last attribute in the command string.

If you omit *`startup-param-text`*, no startup parameter text is used.

Considerations

- An expired user cannot log on.

  When a user’s access expires, the user cannot log on to the system, but the user’s authentication record is not deleted.

- GUARDIAN DEFAULT attributes are equivalent to using the Guardian DEFAULT command.

  Setting the user’s Guardian default file security or default subvolume with the GUARDIAN SECURITY or GUARDIAN VOLUME attributes is equivalent to using the Guardian DEFAULT command. Similarly, you can use the DEFAULT command to change these attributes.

- PASSWORD-EXPIRES takes precedence over PASSWORD-MUST-CHANGE.

  If the PASSWORD-EXPIRES and PASSWORD-MUST-CHANGE attributes are set in the same command, the setting of the PASSWORD-EXPIRES attribute takes precedence over the PASSWORD-EXPIRES date calculated as a result of setting the PASSWORD-MUST-CHANGE attribute.

Examples

The group manager for group 18 enters the following SET and ADD commands to add a temporary user to the system. The user has user name TEMP.KEVIN and user ID 18,38. The user cannot log on after noon on September 30, 2005.

First the group manager issues the SET USER commands:

```
=ASSUME USER
=SET PASSWORD lintel
=SET USER-EXPIRES 30 SEP 2005, 12:00
=SET AUDIT-AUTHENTICATE-FAIL all
=SET GUARDIAN VOLUME $data2.kevin
=SET TEXT-DESCRIPTION "Group 18"
```

Then the group manager issues the SHOW USER command to display the changes that have taken place:

```
=SHOW
```

The report displays:

```
 TYPE        OWNER
  USER       18,255

   PASSWORD = lintel
   USER-EXPIRES               = 30SEP05, 12:00
   PASSWORD-EXPIRES           =    * NONE *
   PASSWORD-MUST-CHANGE EVERY =    * NONE *
   PASSWORD-EXPIRY-GRACE      =    * NONE *
   GUARDIAN DEFAULT SECURITY  = OOOO
   GUARDIAN DEFAULT VOLUME    = $DATA2.KEVIN

   AUDIT-AUTHENTICATE-PASS  = NONE        AUDIT-MANAGE-PASS  = NONE
   AUDIT-AUTHENTICATE-FAIL  = ALL         AUDIT-MANAGE-FAIL  = NONE
   AUDIT-USER-ACTION-PASS   = NONE
   AUDIT-USER-ACTION-FAIL   = NONE

   TEXT-DESCRIPTION = "Group 18"

   CI-PROG = * NONE *
   CI-LIB  = * NONE *
   CI-NAME = * NONE *
   CI-SWAP = * NONE *
   CI-CPU  = ANY
   CI-PRI  = * NONE *
   CI-PARAM-TEXT =

   INITIAL-PROGTYPE      = PROGRAM
   INITIAL-PROGRAM       =
   INITIAL-DIRECTORY     =

  SUBJECT DEFAULT-PROTECTION SECTION UNDEFINED!

  SUBJECT OWNER-LIST SECTION UNDEFINED!
```

Next the group manager issues the ADD USER command to add Kevin to group 18:

```
=ADD temp.kevin, 18,38
```

Finally the group manager issues the INFO USER command to display a report of the entire process:

```
=INFO temp.kevin,GENERAL,AUDIT
```

The report displays:

```
GROUP.USER        USER-ID    OWNER     LAST-MODIFIED    LAST-LOGON
STATUS
TEMP.KEVIN         18,38     18,255    6MAY05, 10:47     * NONE *
THAWED

  UID                        =       4646
  USER-EXPIRES               = 30SEP05, 12:00
  PASSWORD-EXPIRES           =    * NONE *
  PASSWORD-MUST-CHANGE EVERY =    * NONE *
  PASSWORD-EXPIRY-GRACE      =    * NONE *
  LAST-LOGON                 =    * NONE *
  LAST-UNSUCESSFUL-ATTEMPT   =    * NONE *
  LAST-MODIFIED                =  6MAY05, 10:47
  CREATION-TIME              = 5APR09, 1:00
  FROZEN/THAWED              = THAWED
  STATIC FAILED LOGON COUNT  =          0
  STATIC-FAILED-LOGON-RESET  =    * NONE *
  GUARDIAN DEFAULT SECURITY  = 0000
  GUARDIAN DEFAULT VOLUME    = $DATA2.KEVIN

  AUDIT-AUTHENTICATE-PASS  = NONE        AUDIT-MANAGE-PASS  = NONE
  AUDIT-AUTHENTICATE-FAIL  = ALL         AUDIT-MANAGE-FAIL  = NONE
  AUDIT-USER-ACTION-PASS   = NONE
  AUDIT-USER-ACTION-FAIL   = NONE
```

SHOW USER Command

SHOW USER displays the current default user-attribute values.

When you add a user to the system, the current default user attribute values are used for any attribute you do not specify in the ADD USER command.

To set the default user attribute values to specific values, use SET USER.

```
SHOW [ / OUT listfile / ] USER
```

`OUT listfile`

directs SAFECOM output to *`listfile`* for the SHOW report.

For *`listfile`*, specify any file name. SAFECOM opens *`listfile`* and appends the output text to the file. If *`listfile`* does not exist, SAFECOM creates an EDIT-format file and writes the SHOW report to it.

`USER`

identifies USER as the object class of the SHOW command. Omit it if USER is the assumed type. (For more information, see the ASSUME Command on page 4-3.)

SHOW USER Report Format

The SHOW USER command displays the default user attributes and their current values in the format shown in Figure 5-3 on page 5-55.

Figure 5-3. SHOW USER Report Format

```
 TYPE        OWNER
  USER       gn,un

   PASSWORD = [password]
   USER-EXPIRES               =  { date,time | * NONE * }
   PASSWORD-EXPIRES           =  { date,time | * NONE * }
   PASSWORD-MUST-CHANGE EVERY =  { n DAYS | * NONE * }
   PASSWORD-EXPIRY-GRACE      =  { n DAYS | * NONE * }
   GUARDIAN DEFAULT SECURITY  = string
   GUARDIAN DEFAULT VOLUME    = $vol.subvol

   AUDIT-AUTHENTICATE-PASS = a-spec      AUDIT-MANAGE-PASS = a-spec
   AUDIT-AUTHENTICATE-FAIL = a-spec      AUDIT-MANAGE-FAIL = a-spec
   AUDIT-USER-ACTION-PASS  = a-spec
   AUDIT-USER-ACTION-FAIL  = a-spec

   TEXT-DESCRIPTION = ["text"]

   CI-PROG = { prog-filename | * NONE * }
   CI-LIB  = { lib-filename | * NONE * }
   CI-NAME = { process-name | * NONE * }
   CI-SWAP = { $vol[.subvol.filename] | * NONE * }
   CI-CPU  = { num | ANY }
   CI-PRI  = { num | * NONE * }
   CI-PARAM-TEXT = [ param-text ]

   INITIAL-PROGTYPE      = prog-type
   INITIAL-PROGRAM       = [prog-path]
   INITIAL-DIRECTORY     = [dir-path]

  SUBJECT DEFAULT-PROTECTION SECTION

  SUBJECT OWNER-LIST SECTION
```

These user attribute values are:

`OWNER` *`gn,un`*

is the user ID (group number and member number) of the user who will own the user authentication record.

`PASSWORD = [`*`password`*`]`

identifies the currently assigned logon password. If no password has been assigned, *`password`* does not appear.

`USER-EXPIRES = {` *`date,time`* `| * NONE * }`

either gives the date and time when the user's ability to log on will expire or indicates that no expiration date has been specified.

`PASSWORD-EXPIRES = {` *`date,time`* `| * NONE * }`

either gives the date and time when the user's password expires or indicates that no expiration date has been specified.

`PASSWORD-MUST-CHANGE EVERY = { n DAYS | * NONE * }`

either gives the maximum number of days that the user can retain the same password or indicates that no limit has been set.

`PASSWORD-EXPIRY-GRACE = { n DAYS | * NONE * }`

either gives the number of days after password expiration that the user can change his or her password during logon or indicates that no extension period is allowed.

`GUARDIAN DEFAULT SECURITY = string`

gives the Guardian default disk file security string.

`GUARDIAN DEFAULT VOLUME = $vol.subvol`

gives the Guardian default subvolume.

```
AUDIT-AUTHENTICATE-PASS = a-spec    AUDIT-MANAGE-PASS = a-spec
AUDIT-AUTHENTICATE-FAIL = a-spec    AUDIT-MANAGE-FAIL = a-spec
AUDIT-USER-ACTION-PASS  = a-spec
AUDIT-USER-ACTION-FAIL  = a-spec
```

indicate the conditions under which attempts to authenticate the user, attempts to manage the user’s authentication record, and attempts by the user to perform an operation are audited by the Safeguard software. For a detailed description of these fields, see SET USER Command on page 5-40.

`TEXT-DESCRIPTION = [ "text" ]`

is the descriptive text associated with the user authentication record.

`CI-PROG = { prog-filename | * NONE * }`

either gives the object file name of the command interpreter started after the user logs on at a Safeguard terminal or indicates no command interpreter.

`CI-LIB = { lib | * NONE * }`

either gives the file name of the library file used with the command interpreter or indicates no library file.

`CI-NAME = { process-name | * NONE * }`

either gives the process name assigned to the command interpreter or indicates no process name.

`CI-SWAP = { $vol[.subvol.filename] | * NONE * }`

either gives the swap volume or file used with the command interpreter or indicates no swap volume or file is specified.

`CI-CPU = {` *`num`* `| ANY }`

either gives the number of the CPU in which the command interpreter runs or indicates any CPU is used.

`CI-PRI = {` *`num`* `| * NONE * }`

either gives the priority at which the command interpreter runs or indicates that no priority is assigned in the user record.

`CI-PARAM-TEXT = [` *`text`* `]`

either gives the startup parameter text supplied to the command interpreter or appears blank to indicate that no parameter is supplied.

`INITIAL-PROGTYPE =` *`prog-type`*

gives the initial program type: PROGRAM, WINDOW, or SERVICE.

`INITIAL-PROGRAM = [` *`prog-path`* `]`

either gives the initial program pathname or appears blank to indicate that no pathname is defined.

`INITIAL-DIRECTORY = [` *`dir-path`* `]`

either gives the initial directory pathname or appears blank to indicate that no pathname is defined.

`SUBJECT DEFAULT-PROTECTION SECTION`

either gives the default protection to be assigned to the user’s disk files or indicates that no default protection is defined.

`SUBJECT OWNER-LIST SECTION`

lists the secondary owners of the user authentication record.

Examples

1. This sample SHOW USER command displays the predefined user-attribute settings for the user ID 86,2:

```
=SHOW USER
```

The report displays:

```
TYPE        OWNER
 USER       86,2

  PASSWORD =
  USER-EXPIRES               =    * NONE *
  PASSWORD-EXPIRES           =    * NONE *
  PASSWORD-MUST-CHANGE EVERY =    * NONE *
  PASSWORD-EXPIRY-GRACE      =    * NONE *
  GUARDIAN DEFAULT SECURITY  = OOOO
  GUARDIAN DEFAULT VOLUME    = $SYSTEM.NOSUBVOL

  AUDIT-AUTHENTICATE-PASS  = NONE        AUDIT-MANAGE-PASS  = NONE
  AUDIT-AUTHENTICATE-FAIL  = NONE        AUDIT-MANAGE-FAIL  = NONE
  AUDIT-USER-ACTION-PASS   = NONE
  AUDIT-USER-ACTION-FAIL   = NONE

  TEXT-DESCRIPTION =

  CI-PROG = * NONE *
  CI-LIB  = * NONE *
  CI-NAME = * NONE *
  CI-SWAP = * NONE *
  CI-CPU  = ANY
  CI-PRI  = * NONE *
  CI-PARAM-TEXT =

  INITIAL-PROGTYPE      = PROGRAM
  INITIAL-PROGRAM       =
  INITIAL-DIRECTORY     =

 SUBJECT DEFAULT-PROTECTION SECTION UNDEFINED!

 SUBJECT OWNER-LIST SECTION UNDEFINED!
```

2. In this example, the PRS group manager (group 86) enters the SHOW USER command after using the SET USER command to establish these user values:

   - A logon password of macaroon
   - A USER-EXPIRES date of December 15, 2005, at midnight
   - A PASSWORD-MUST-CHANGE requirement of 30 days
   - An AUDIT-AUTHENTICATE-PASS specification to audit all successful attempts to authenticate the user
   - An AUDIT-MANAGE-FAIL specification to audit all unsuccessful attempts to manage the user’s authentication record
   - A GUARDIAN SECURITY of NUNU and a GUARDIAN VOLUME of $tops.harry
   - Descriptive text of “Belongs to the ABC.COM domain”

   The PRS group manager enters:

   ```
   =SHOW USER
   ```

The report displays:

```
TYPE        OWNER
 USER       86,255

  PASSWORD = macaroon
  USER-EXPIRES               = 15DEC05, 0:00
  PASSWORD-EXPIRES           =    * NONE *
  PASSWORD-MUST-CHANGE EVERY =    30 DAYS
  PASSWORD-EXPIRY-GRACE      =    * NONE *
  GUARDIAN DEFAULT SECURITY  = NUNU
  GUARDIAN DEFAULT VOLUME    = $TOPS.HARRY

  AUDIT-AUTHENTICATE-PASS  = ALL         AUDIT-MANAGE-PASS  = NONE
  AUDIT-AUTHENTICATE-FAIL  = NONE        AUDIT-MANAGE-FAIL  = ALL
  AUDIT-USER-ACTION-PASS   = NONE
  AUDIT-USER-ACTION-FAIL   = NONE

  TEXT-DESCRIPTION = "Belongs to the ABC.DOM domain"

  CI-PROG = * NONE *
  CI-LIB  = * NONE *
  CI-NAME = * NONE *
  CI-SWAP = * NONE *
  CI-CPU  = ANY
  CI-PRI  = * NONE *
  CI-PARAM-TEXT =

  INITIAL-PROGTYPE      = PROGRAM
  INITIAL-PROGRAM       =
  INITIAL-DIRECTORY     =

 SUBJECT DEFAULT-PROTECTION SECTION UNDEFINED!

 SUBJECT OWNER-LIST SECTION UNDEFINED!
```

THAW USER Command

THAW USER restores a frozen user's ability to log on to a system. (After a FREEZE USER command, the user cannot log on to the system until a THAW USER command is executed.)

THAW USER has no effect on a user whose access is not currently frozen.

The primary owner and secondary owners of a user's authentication record, the primary owner's group manager, and the super ID can thaw a user's access to the system.

```
THAW USER { user-spec | ( user-spec [ , user-spec ] ... ) }

    [ [,] WHERE expression ]
```

USER

identifies USER as the object type of the THAW command. Omit it if USER is the assumed object type. (For more information, see the ASSUME Command on page 4-3.)

user-spec

specifies the user (or users) whose ability to log on is to be restored. *user-spec* can be any of:

```
group-num , member-num
group-name.member-name
group-num , *
*,*
```

group-name and *member-name* can contain wild-card characters.

WHERE *expression*

causes the THAW command to apply only to authentication records for users who belong to the groups specified by *expression*. For a description of WHERE *expression*, see the ALTER USER Command on page 5-10.

Examples

1. Either of these commands restores the logon ability for PRS.HARRY (user ID 86,2):

```
=THAW USER prs.harry
```

```
=THAW USER 86,2
```

2. The following command restores the logon ability for all users whose administrative group number is 48 and who are also members of the group temp3:

```
=THAW USER 48,* WHERE GROUP=temp3
```

6 User Alias Security Commands

Each user can be assigned one or more additional names, called “user aliases.” An alias is an alternate name that can be used to log on to the system. Each alias has its own alias authentication record and set of user attributes. The values assigned to the user attributes in the alias authentication record can differ from those values assigned to the user attributes in the user authentication record.

SAFECOM commands can add aliases to the system, delete aliases from the system, and suspend the ability of an alias to log on to the system. They can also specify auditing for attempts by an alias to log on to the system and attempts to manage an alias authentication record.

This section contains these subsections:

- A description of who can add new aliases to the system and who can manage the alias authentication records
- A summary table of the user alias commands
- Detailed syntax for each user alias command

Who Can Manage User Aliases

Because an important attribute of a user alias is an underlying user ID, special restrictions apply to the use of ALIAS commands. In particular, the ADD ALIAS command is subject to additional security. The general rule is that to add an alias authentication record, you must have the authority to add the underlying user ID and alter the record for that user ID. Specifically, the ADD ALIAS command is restricted as follows:

- If an OBJECTTYPE USER record exists, the person executing the ADD ALIAS command must meet these two qualifications:
    - Have CREATE (C) authority on the OBJECTTYPE USER access control list
    - Be the owner of the underlying user ID or be the group manager of the owner of the underlying user ID
- If an OBJECTTYPE USER record does not exist, the person executing the ADD ALIAS command must meet these two qualifications:
    - Be the group manager of the underlying user ID
    - Be the owner of the underlying user ID or be the group manager of the owner of the underlying user ID
- In addition, the local super ID can add an alias for any user, regardless of the existence of an OBJECTTYPE USER record (unless OBJECTTYPE USER specifically denies the super ID).

An alias authentication record can have multiple owners. The OWNER attribute in an alias authentication record designates the record's primary owner. The OWNER-LIST attribute optionally designates one or more secondary owners. By default, the OWNER attribute contains the user ID of the user who first created the alias authentication record. The OWNER and OWNER-LIST attributes can be changed with a SET ALIAS command before the record is created, or they can be changed with an ALTER ALIAS command after the record is created. These record owners can change the security attributes in the alias authentication record and therefore control the ability of the alias to log on to the system.

Only the primary and secondary record owners of the alias record, the primary owner's group manager, and the super ID can change an alias authentication record, suspend and restore the ability of the alias to log on to the system, and delete the alias (ALTER ALIAS, FREEZE ALIAS, THAW ALIAS, and DELETE ALIAS commands, respectively).

The original primary owner and the secondary owners of an alias authentication record can change the OWNER attribute to the user ID of any other user. That other user then has control of the ability of the alias to access the system. At any time, the new primary owner (or the secondary owners or the primary owner's group manager or the super ID) can transfer ownership to yet another user.

Note. If SUPER.SUPER is declared undeniable, any access denial by Safeguard ACLs is ignored. This applies to both aliases of SUPER.SUPER and the SUPER.SUPER user because all the checks are performed only on the User ID.

The ability to display the security attributes of an alias through the INFO ALIAS command is restricted to these users:

- The user who was assigned the alias
- The primary and secondary owners of the alias authentication record
- The group manager of the primary owner of the alias authentication record
- The super ID

Any alias of the user can execute the INFO USER command for any other alias of the user.

Table 6-1 shows who can use the user alias commands to display, add, modify, or delete an alias authentication record.

Table 6-1. Who Can Use the User Alias Commands (page 1 of 2)

ALIAS Command	Who Can Use
SET ALIAS LIKE	Any user, primary and secondary record owners, primary owner's group manager, and super ID
INFO ALIAS	User assigned the alias, primary and secondary record owners, primary owner's group manager, and super ID
ALTER ALIAS	Primary and secondary record owners, primary owner's group manager, and super ID

Table 6-1. Who Can Use the User Alias Commands (page 2 of 2)

ALIAS Command	Who Can Use
FREEZE ALIAS	Primary and secondary record owners, primary owner’s group manager, and super ID
THAW ALIAS	Primary and secondary record owners, primary owner’s group manager, and super ID
DELETE ALIAS	Primary and secondary record owners, primary owner’s group manager, and super ID
ADD ALIAS	See the description at the beginning of this subsection

Aliases and Access Control Lists

An alias name cannot appear on a Safeguard access control list. However, the Safeguard software still rules on access attempts by a user who is logged on with an alias name. When making access decisions for a user logged on as an alias, the Safeguard software checks access control lists for the user’s underlying user ID. For example, if user 16,24 is logged on under the alias PhilM, each attempt to access a protected object causes the Safeguard software to check that object’s access control list to determine if user 16,24 has the proper access authority.

In addition, if the user is logged on as an alias and attempts to add objects to the Safeguard database or attempts to manage protected objects, these actions are treated as if they are being performed by the underlying user ID.

User Attributes Applicable to an Alias

The user attributes applicable to a user authentication record are also applicable to an alias authentication record. For any particular attribute, the value can differ between a user’s user authentication record and that user’s alias authentication records. For example, different auditing specifications can be applied to the user and the alias, or different owners can be specified for the alias and user protection records.

User Alias Command Summary

Table 6-2 on page 6-4 summarizes each of the user alias commands.

Table 6-2. User Alias Command Summary

Command	Function
ADD ALIAS	Adds a user alias to the system and creates an authentication record for that alias with the user attribute values specified in the command. For any unspecified attributes, the current default values are used. (To set default values, use the SET ALIAS command.) After being added, a user can log on to the system with the user alias.
ALTER ALIAS	Changes the value of one or more user attributes in an alias authentication record.
DELETE ALIAS	Deletes a user alias from the system. The user cannot log on to the system with the deleted user alias.
FREEZE ALIAS	Temporarily suspends a user's ability to log on using the specified alias.
INFO ALIAS	Displays the existing attribute values defined for a specific user alias.
RESET ALIAS	Sets one or more current default user attributes to predefined values.
SET ALIAS	Sets one or more default user attribute values to specified values. When a user alias is added, the current default attributes are used for any attribute not specified in the ADD ALIAS command.
SHOW ALIAS	Displays the current default values of the user attributes.
THAW ALIAS	For a frozen alias, restores the user's ability to log on to the system with that alias.

Syntax of User Alias Commands

The rest of this section contains individual syntax descriptions for the user alias commands. Commands are presented in alphabetical order, and most of the descriptions contain these elements:

- A summary of the function performed by the command, including the restrictions on who can use the command
- Descriptions of the command parameters and variables
- The format for the command listing or report (for commands that produce displays or listings)
- Considerations for the use of the command
- Examples of command usage

ADD ALIAS Command

ADD ALIAS adds a user alias to the system and creates a Safeguard authentication record for that alias. Once a new alias is added to the system for a user, the user can log on to the system with that alias.

To execute the ADD ALIAS command, you must have the authority both to add the underlying user ID and to alter the authentication record for that user ID. For more information, see Who Can Manage User Aliases on page 6-1.

You can use the SET ALIAS command to set default values for the user attributes and then use ADD ALIAS to identify the user alias record to which the default values are to be assigned. You can also specify attribute values in your ADD ALIAS command. The current default values are used for any attributes left unspecified.

```
ADD ALIAS alias [ , ]

   { group-name.member-name | group-num,member-num }

   [ LIKE user | user-attribute ]

   [ , user-attribute ] ...
```

`ALIAS`

identifies ALIAS as the object class of the ADD command. Omit it if ALIAS is the assumed class. (For more information about assumed commands, see the ASSUME Command on page 4-3.)

`alias`

is the alias name to be added to the system. The name must be unique within the local system. *`alias`* is a case-sensitive text string that can be up to 32 alphanumeric characters in length. In addition to alphabetic and numeric characters, the characters period (.), hyphen (-), and underscore (_) are permitted within the text string. The first character of an alias name must be alphabetic or numeric.

An alias name cannot match an existing user name when the alias is converted to uppercase letters. For example, the alias Prog3.SueB is not valid if a user name PROG3.SUEB already exists.

`group-name.member-name`

is the user name of the user with which this alias is to be associated. The *`group-name.member-name`* must already exist.

`group-name` and *`member-name`* cannot contain wild-card characters.

group-num,member-num

is the user ID of the user with which this alias is to be associated. The *group-num,member-num* must already exist.

LIKE *user*

adopts the attribute values from an existing alias or user authentication record as the attribute values for the alias authentication record being added.

user is an existing user, specified in one of these formats:

```
alias
group-num,member-num
group-name.member-name
```

LIKE defines values for all attributes except:

```
PASSWORD [password]
REMOTEPASSWORD \system-name remote-password
GUARDIAN [DEFAULT] SECURITY ["]string["]
GUARDIAN [DEFAULT] [SUB]VOLUME [\system.]$vol.subvol
CI-NAME [process-name]
PRIMARY-GROUP [ [ NAME ] group-name | NUMBER group-num ]
INITIAL-DIRECTORY [dir-path]
INITIAL-PROGRAM [prog-path]
INITIAL-PROGTYPE [prog-type]
TEXT-DESCRIPTION "[text]"
```

user-attribute

defines a user attribute value for the alias being added. (The current default user values are used for any attributes not specified in the ADD ALIAS command.)

user-attribute can be:

```
OWNER [owner-id]
OWNER-LIST [[-]user-list]
PASSWORD [password]
USER-EXPIRES [date [, time]]
PASSWORD-MUST-CHANGE [EVERY num DAYS]
PASSWORD-EXPIRY-GRACE [num [DAYS]]
PASSWORD-EXPIRES [ date [ , time] ]
AUDIT-AUTHENTICATE-PASS [audit-spec]
AUDIT-AUTHENTICATE-FAIL [audit-spec]
AUDIT-MANAGE-PASS [audit-spec]
AUDIT-MANAGE-FAIL [audit-spec]
AUDIT-USER-ACTION-PASS [audit-spec]
AUDIT-USER-ACTION-FAIL [audit-spec]
TEXT-DESCRIPTION "[text]"
REMOTEPASSWORD \system-name remote-password
DEFAULT-PROTECTION [ obj-attr ]
                   [ ( obj-attr [ , obj-attr ] ... ) ]
GUARDIAN [DEFAULT] SECURITY ["]string["]
GUARDIAN [DEFAULT] [SUB]VOLUME [\system.]$vol.subvol
INITIAL-DIRECTORY [dir-path]
```

```
INITIAL-PROGRAM [prog-path]
INITIAL-PROGTYPE [prog-type]
CI-PROG [prog-filename]
CI-LIB [lib-filename]
CI-CPU [cpu-number | ANY]
CI-NAME [process-name]
CI-SWAP [$vol.[subvol.filename]]
CI-PRI [priority]
CI-PARAM-TEXT [startup-param-text]
```

Multiple *remote-password* entries are not allowed on the same line.

Because the CI-PARAM-TEXT, INITIAL-DIRECTORY, and INITIAL-PROGRAM attributes are text strings, each of them must be the last attribute specified in the command string. Therefore only one of these attributes can be specified in a single command.

For a complete description of each *user-attribute*, see the ALTER ALIAS Command on page 6-9 and the SET ALIAS Command on page 6-38.

Considerations

- The primary group for a new alias is set to the administrative group of the underlying user ID associated with the alias.

  When you add an alias, the administrative group for the underlying user ID becomes the primary group for that alias. To change the primary group, use the ALTER ALIAS command to alter the PRIMARY-GROUP attribute.

- PASSWORD-EXPIRES takes precedence over PASSWORD-MUST-CHANGE.

  If the PASSWORD-EXPIRES and PASSWORD-MUST-CHANGE attributes are set in the same ADD command, the setting of the PASSWORD-EXPIRES attribute takes precedence over the PASSWORD-EXPIRES date calculated as a result of setting the PASSWORD-MUST-CHANGE attribute.

Examples

1. The group manager for a marketing group (group name PRS) uses this command to add the alias Admin_Darlene for the user PRS.DARLENE:

   ```
   =ADD ALIAS Admin_Darlene, prs.darlene , PASSWORD BlueGill
   ```

   This command allows PRS.DARLENE to log on using the alias Admin_Darlene. When she logs on as Admin_Darlene, her password is BlueGill. The other user attributes for PRS.DARLENE have their default values.

2. The PRS group manager next adds aliases for two more group members.

   First, the group manager uses the SET command to create a pattern of attribute values—a useful procedure for adding a number of aliases that share attributes:

   ```
   =ASSUME ALIAS
   =SET USER-EXPIRES jun 28 2005 , &
   ```

```
=AUDIT-AUTHENTICATE-PASS all &
=PASSWORD-MUST-CHANGE EVERY 60 DAYS &
=OWNER-LIST 86,6 &
=TEXT-DESCRIPTION "Fred’s group"
```

These users must change passwords for their aliases every 60 days. Their ability to log on using the aliases expires at midnight on June 28, 2005. All successful authentication attempts using the aliases are audited by the Safeguard software. User 86,6 is added as a default secondary owner of the alias authentication records. That owner will have the same privileges as the records’ primary owner. Descriptive text is entered to identify the group manager.

The group manager then issues the SHOW command to check that the user attributes were entered correctly:

```
=SHOW
```

The report shows:

```
TYPE        OWNER
 ALIAS      86,255

 PASSWORD =
 USER-EXPIRES               = 28JUN05,  0:00
 PASSWORD-EXPIRES           =    * NONE *
 PASSWORD-MUST-CHANGE EVERY =    60 DAYS
 PASSWORD-EXPIRY-GRACE      =    * NONE *
 GUARDIAN DEFAULT SECURITY  = OOOO
 GUARDIAN DEFAULT VOLUME    = $SYSTEM.NOSUBVOL

 AUDIT-AUTHENTICATE-PASS  = ALL          AUDIT-MANAGE-PASS  = NONE
 AUDIT-AUTHENTICATE-FAIL  = NONE         AUDIT-MANAGE-FAIL  = NONE
 AUDIT-USER-ACTION-PASS   = NONE
 AUDIT-USER-ACTION-FAIL   = NONE

 TEXT-DESCRIPTION = "Fred’s group"

 CI-PROG = * NONE *
 CI-LIB  = * NONE *
 CI-NAME = * NONE *
 CI-SWAP = * NONE *
 CI-CPU  = ANY
 CI-PRI  = * NONE *
 CI-PARAM-TEXT =

 INITIAL-PROGTYPE      = PROGRAM
 INITIAL-PROGRAM       =
 INITIAL-DIRECTORY     =

SUBJECT DEFAULT-PROTECTION SECTION UNDEFINED!

SUBJECT OWNER-LIST SECTION

         86,6
```

To add aliases for the other users, the group manager uses the ADD command:

```
=ADD Mgr-Mabel, prs.mabel, OWNER 86,2, PASSWORD seaSide
=ADD Admin-Jack, prs.jack, OWNER 86,2, PASSWORD TROUT3
```

PRS.MABEL has the user alias Mgr-Mabel and the password seaSide, and PRS.JACK has the user alias Admin-Jack and the password TROUT3. The authentication records for both aliases belong to user 86,2.

3. To add the alias BENNY1 for the user 86,4, the PRS manager uses the LIKE clause with the ADD command:

```
=ADD ALIAS BENNY1, 86,4 , LIKE prs.darlene, &
=PASSWORD GoFish
```

This LIKE clause gives the new alias (BENNY1, who has the user ID 86,4) the same user attributes as the user PRS.DARLENE.

ALTER ALIAS Command

ALTER ALIAS changes one or more user attributes in an alias authentication record.

Only the primary owner of an alias authentication record, the secondary owners, the primary owner’s group manager, or the local super ID can use ALTER ALIAS to change the user attribute values in an alias authentication record.

For all attributes other than REMOTEPASSWORD, the ALTER ALIAS command replaces the current attribute value with the newly specified value. For the REMOTEPASSWORD attribute, ALTER ALIAS updates the remote password list by adding, deleting, or changing the corresponding remote password as indicated.

You cannot alter the underlying user ID for an alias.

```
ALTER ALIAS { alias | ( alias [ , alias ] ... ) }

   [ , ] { LIKE user | user-attribute }

   [ , user-attribute ] ... [ [,]

   WHERE expression ]
```

`ALIAS`

specifies ALIAS as the object type of the ALTER command. Omit it if ALIAS is the assumed type. (For more information about assumed types, see the ASSUME Command on page 4-3.)

`alias`

specifies the alias or aliases whose authentication records are to be changed. *`alias`* is a text-string as defined under the ADD ALIAS command. The *`alias`* can contain wild-card characters.

`LIKE` *`user`*

adopts the attribute values from an existing alias or user authentication record as the attribute values for the alias authentication record being added.

user is an existing user specified in one of these formats:

```
alias
group-num,member-num
group-name.member-name
```

LIKE changes the values of all attributes except:

```
PASSWORD [password]
REMOTEPASSWORD \system-name remote-password
GUARDIAN [DEFAULT] SECURITY ["]string["]
GUARDIAN [DEFAULT] [SUB]VOLUME [\system.]$vol.subvol
CI-NAME [process-name]
PRIMARY-GROUP [ [ NAME ] group-name | NUMBER group-num ]
INITIAL-DIRECTORY [dir-path]
INITIAL-PROGRAM [prog-path]
INITIAL-PROGTYPE [prog-type]
TEXT-DESCRIPTION "[text]"
```

user-attribute

changes the current value of the specified attribute. The attributes are:

```
OWNER [owner-id]
OWNER-LIST [[-]user-list]
PASSWORD [password]
USER-EXPIRES [ date [ , time ] ]
PASSWORD-MUST-CHANGE [EVERY num DAYS]
PASSWORD-EXPIRY-GRACE [num [DAYS]]
PASSWORD-EXPIRES [ date [ , time] ]
AUDIT-AUTHENTICATE-PASS [audit-spec]
AUDIT-AUTHENTICATE-FAIL [audit-spec]
AUDIT-MANAGE-PASS [audit-spec]
AUDIT-MANAGE-FAIL [audit-spec]
AUDIT-USER-ACTION-PASS [audit-spec]
AUDIT-USER-ACTION-FAIL [audit-spec]
TEXT-DESCRIPTION "[text]"
REMOTEPASSWORD [ \sys-name [ remote-password ] ]
DEFAULT-PROTECTION [ obj-attr ]
                   [ ( obj-attr [ , obj-attr ] ...) ]
GUARDIAN [DEFAULT] SECURITY ["]string["]
GUARDIAN [DEFAULT] [SUB]VOLUME [\system.]$vol.subvol
PRIMARY-GROUP [ [ NAME ] group-name | NUMBER group-num ]
INITIAL-DIRECTORY [dir-path]
INITIAL-PROGRAM [prog-path]
INITIAL-PROGTYPE [prog-type]
CI-PROG [prog-filename]
CI-LIB [lib-filename]
CI-CPU [cpu-number | ANY]
CI-NAME [process-name]
CI-SWAP [$vol.[subvol.filename]]
CI-PRI [priority]
CI-PARAM-TEXT [startup-param-text]
```

```
OWNER [owner-id]
```

transfers the primary ownership of an alias authentication record to the user whose user ID is specified as *owner-id*. For *owner-id*, specify either of:

```
[\*.]group-name.member-name
[\*.]group-num , member-num
```

If you omit *owner-id*, it is set to your user ID.

```
OWNER-LIST [[-]user-list]
```

changes the secondary ownership of alias authentication record by adding or deleting owners in the owner list. A minus sign (-) preceding *user-list* indicates that the specified users are to be deleted from the existing owner list. If the minus sign is omitted, the specified users are added to the owner list. If you omit *user-list*, the owner list is set to null. A maximum of 50 users can be specified in *user-list*. For *user-list*, specify either:

```
net-user-spec
(net-user-spec [, net-user-spec ...])

net-user-spec is either:


[\node-spec.]group-name.user-name
[\node-spec.]group-num , user-num

node-spec is one of:


*
node-name
node-number
```

```
node-name
```

specifies the system name.

```
node-number
```

specifies the Expand node number.

```
PASSWORD [password]
```

changes the logon password for this alias.

```
password
```

is a string of one to 64 characters. It can contain any alphanumeric characters except blanks, commas, semicolons, and the ASCII null character. The case of the letters is preserved. Lowercase letters remain lowercase, and uppercase remain uppercase.

If omitted, the value for *`password`* is set to null. In this case, the password is not required for the user to log on to the system.

The password is subject to the restrictions imposed by the configuration options described in Section 16, Safeguard Subsystem Commands.

▲ **WARNING.** Only the first eight characters of the password will be considered.

`USER-EXPIRES [ date [ , time] ]`

changes the user-expiration date to the specified date and time. (Both are local civil time.)

After USER-EXPIRES suspends a user's ability to log on to the system with this alias, changing the USER-EXPIRES attribute to some future date restores that ability to log on.

If you omit both *`date`* and *`time`*, the user-expiration attribute value is set to null, and the user's ability to log on to the system with this alias never expires.

If omitted, *`time`* is set to 0:00 (midnight).

The form of *`date`* [, *`time`*] is:

```
{ month-name day } year [,hour:min]
{ day month-name }
```

`month-name`

is the first three letters of the month name: JAN, FEB, MAR, and so on. You can use uppercase or lowercase letters.

`day`

is a 1-digit or 2-digit integer from 1 through 31, specifying the day of the month.

`year`

is a 4-digit integer, specifying the year.

`hour`

is an integer from 0 through 23, specifying the hour.

`min`

is an integer from 0 through 59, specifying the minute.

`PASSWORD-MUST-CHANGE [EVERY num DAYS]`

changes the maximum number of days that a user can use the same password. For *`num`*, specify an integer from 1 through 32,767.

Changing the PASSWORD-MUST-CHANGE attribute causes the Safeguard software to calculate a new PASSWORD-EXPIRES date. The PASSWORD-EXPIRES date is set to the current date, plus *num* days.

After PASSWORD-EXPIRES suspends a user's ability to log on to the system with this alias, extending the alias PASSWORD-MUST-CHANGE period can restore that ability. (For more information on how the PASSWORD-MUST-CHANGE operation works, see the SET ALIAS Command on page 6-38.)

Setting the PASSWORD-EXPIRES attribute after setting the PASSWORD-MUST-CHANGE attribute causes the explicit setting of the PASSWORD-EXPIRES attribute to override the date previously calculated as a result of setting PASSWORD-MUST-CHANGE.

Omitting "EVERY *num* DAYS" disables the PASSWORD-MUST-CHANGE mechanism, and the alias password never expires.

`PASSWORD-EXPIRY-GRACE [`*`num`*` [DAYS]]`

changes the number of days after password expiration during which the password for this alias can be changed during logon. For *num*, specify an integer from 0 through 32,767. If you omit *num*, the value for PASSWORD-EXPIRY-GRACE in the Safeguard configuration record is used. In this instance, the value *NONE* appears in this field of the alias protection record.

`PASSWORD-EXPIRES [ `*`date`*` [ , `*`time`*`] ]`

changes the date and time after which the password expires. Specify *date* and *time* in local civil time.

If you omit both *date* and *time*, no expiration is set for the password. (However, an expiration date is calculated and set if the PASSWORD-MUST-CHANGE period is subsequently specified or altered.)

If you omit only *time*, it is set to 0:00 (midnight).

The form of *date* [, *time*] is:

```
{ month-name day } year [,hour:min]
{ day month-name }
```

month-name

is the first three letters of the month name: JAN, FEB, MAR, and so on. You can use either uppercase or lowercase letters.

day

is a 1-digit or 2-digit integer from 1 through 31.

year

is a 4-digit integer.

hour

is an integer from 0 through 23.

min

is an integer from 0 through 59.

Setting the PASSWORD-MUST-CHANGE attribute after setting the PASSWORD-EXPIRES attribute causes the PASSWORD-EXPIRES date calculated as a result of setting PASSWORD-MUST-CHANGE to override the explicit setting of the PASSWORD-EXPIRES attribute.

AUDIT-AUTHENTICATE-PASS [*audit-spec*]

changes the *audit-spec* for successful user authentication (logon) attempts with this alias. The form of *audit-spec* is:

```
{ ALL | LOCAL | REMOTE | NONE }
```

For a description of *audit-spec*, see the SET ALIAS Command on page 6-38. Omitting *audit-spec* specifies NONE.

AUDIT-AUTHENTICATE-FAIL [*audit-spec*]

changes the *audit-spec* for unsuccessful user authentication (logon) attempts with this alias. The form of *audit-spec* is:

```
{ ALL | LOCAL | REMOTE | NONE }
```

For a description of *audit-spec*, see the SET ALIAS Command on page 6-38. Omitting *audit-spec* specifies NONE.

Note. In prior product versions of the Safeguard software, the AUDIT-AUTHENTICATE user attributes were called AUDIT-ACCESS. The user attribute name AUDIT-ACCESS is still supported, but HP discourages its use.

AUDIT-MANAGE-PASS [*audit-spec*]

changes the *audit-spec* for successful attempts to manage the alias authentication record. The form of *audit-spec* is:

```
{ ALL | LOCAL | REMOTE | NONE }
```

For a description of *audit-spec*, see the SET ALIAS Command on page 6-38. Omitting *audit-spec* specifies NONE.

AUDIT-MANAGE-FAIL [*audit-spec*]

changes the *audit-spec* for unsuccessful attempts to manage the alias authentication record. The form of *audit-spec* is:

```
{ ALL | LOCAL | REMOTE | NONE }
```

For a description of *audit-spec*, see the SET ALIAS Command on page 6-38. Omitting *audit-spec* specifies NONE.

AUDIT-USER-ACTION-PASS [*audit-spec*]

changes the *audit-spec* for successful events performed using this alias. The form of *audit-spec* is:

```
{ ALL | LOCAL | REMOTE | NONE }
```

For a description of *audit-spec*, see the SET ALIAS Command on page 6-38. Omitting *audit-spec* specifies NONE.

AUDIT-USER-ACTION-FAIL [*audit-spec*]

changes the *audit-spec* for unsuccessful events attempted using this alias. The form of *audit-spec* is:

```
{ ALL | LOCAL | REMOTE | NONE }
```

For a description of *audit-spec*, see the SET ALIAS Command on page 6-38. Omitting *audit-spec* specifies NONE.

TEXT-DESCRIPTION "[*text*]"

specifies a string of descriptive text to replace the existing description for this alias. All text between the quotation marks is considered descriptive text.

Because SAFECOM allows a maximum command length of 528 characters, the specified string must contain less than 528 characters. You can specify a longer descriptive text string by using the Safeguard SPI interface as described in the *Safeguard Management Programming Manual*.

If you specify TEXT-DESCRIPTION "" without any text, the description for this alias is removed.

RESET-TEXT-DESCRIPTION

resets the text description field to a null value (no descriptive text).

Note. The RESET-TEXT-DESCRIPTION field is supported only on systems running G06.27 and later G-series RVUs and H06.06 and later H-series.

RESET-BINARY-DESCRIPTION

resets the binary description field to zero length and null values. For more information about the binary description field, see the *Safeguard Management Programming Manual*.

Note. The RESET-BINARY-DESCRIPTION field is supported only on systems running H06.06 and later H-series RVUs and G06.27 and later G-series RVUs.

```
REMOTEPASSWORD [ \system-name [ remote-password] ]
```

adds a new remote password, changes the remote password currently defined for a particular system, or deletes a remote password. An alias can have zero, one, or many remote passwords (one for each remote system to which the alias is granted access, as well as one for the local system matching that remote system).

Specifying *system-name* without a *remote-password* deletes the remote password for the specified system.

△ **Caution.** Omitting both *system-name* and *remote-password* deletes all the remote passwords currently defined for the alias on this system.

system-name

specifies the system for which a remote password is to be assigned. For *system-name*, specify a valid system name.

remote-password

specifies a remote password to be associated with *system-name*. For *remote-password*, specify a string from one to eight characters long. You can use any alphanumeric characters except blanks, commas, semicolons, and the ASCII null character. The case of the letters is preserved. Lowercase letters remain lowercase, and uppercase letters remain uppercase. You cannot set multiple remote passwords with one command.

```
DEFAULT-PROTECTION [ obj-attr ]
                   [ ( obj-attr [ , obj-attr ] ...) ]
```

changes one or more attributes to be assigned immediately to new disk files created by this alias. If you omit *obj-attr*, new disk files remain under Guardian protection. If you specify any *obj-attr*, the attribute updates the current default protection record for the specified alias.

obj-attr

is one of:

```
OWNER [ owner-id ]
ACCESS [ access-spec [ ; access-spec ] ... ]
AUDIT-ACCESS-PASS [ audit-spec ]
AUDIT-ACCESS-FAIL [ audit-spec ]
AUDIT-MANAGE-PASS [ audit-spec ]
AUDIT-MANAGE-FAIL [ audit-spec ]
```

For more information about these object attributes as they apply to disk files, see Section 8, Disk-File Security Commands.

`GUARDIAN [DEFAULT] SECURITY ["]`*`string`*`["]`

changes the Guardian default disk file security string for the alias. The word DEFAULT is optional, as are the quotes that surround the security string specifier. You can include them in the command for readability. *`string`* is a four-character string that specifies the Guardian default security string. Each position in the string can contain one of these characters: O, U, G, C, A, or N.

For more information about Guardian default file-security string, see the *Safeguard User's Guide*.

`GUARDIAN [DEFAULT] [SUB]VOLUME [\`*`system`*`.]$`*`vol`*`.`*`subvol`*

changes the Guardian default subvolume. The word DEFAULT and the prefix SUB are optional. You can include them in the command for readability. `\`*`system`* is also optional. If you omit `\`*`system`*, the current system is assumed. `$`*`vol`* specifies the default volume, and *`subvol`* specifies the default subvolume.

`PRIMARY-GROUP [ [ NAME ]` *`group-name`* `| NUMBER` *`group-num`* `]`

changes the name or number of the primary group for the alias. The alias must already belong to this group. The word NAME is optional when you specify *`group-name`*. You can include it in the command for readability. *`group-name`* is the name of a group to which the alias already belongs. *`group-num`* is the number of a group to which the alias already belongs.

You can specify the primary group by group name or by group number, but not both. You cannot include PRIMARY-GROUP NAME and PRIMARY-GROUP NUMBER attributes in the same command. A PRIMARY-GROUP NAME replaces a previously specified PRIMARY-GROUP NUMBER, and vice versa.

The Safeguard software does not implicitly add this group to the alias group list if the alias does not already belong to this group. The previous primary group remains on the alias group list, but not as the primary group.

Without *`group-name`* or *`group-num`*, PRIMARY GROUP clears the primary group setting, and the administrative group for the user ID associated with this alias becomes the primary group. (See Considerations on page 6-20.)

`INITIAL-DIRECTORY [`*`dir-path`*`]`

changes the initial working directory within the OSS file system for the alias. *`dir-path`* is a case-sensitive text string of up to 256 characters. It must be a syntactically valid OSS pathname. If you specify the INITIAL-DIRECTORY attribute, it must be the last attribute in the command string.

If you omit *`dir-path`*, the string is set to null (no pathname).

`INITIAL-PROGRAM [`*`prog-path`*`]`

changes the initial program pathname within the OSS environment for the alias. *`prog-path`* is a case-sensitive text string of up to 256 characters. It must be a syntactically valid OSS pathname. If you specify the INITIAL-DIRECTORY attribute, it must be the last attribute in the command string.

If you omit *`prog-path`*, the string is set to null (no pathname).

`INITIAL-PROGTYPE [`*`prog-type`*`]`

changes the initial program type within the OSS environment for the alias.

`prog-type`

is one of these:

```
PROGRAM
SERVICE
WINDOW
```

This feature is not currently implemented on NonStop systems. It is reserved for future use.

If you omit *`prog-type`*, the initial program type is set to PROGRAM.

`CI-PROG [`*`prog-filename`*`]`

changes the command interpreter to be started after this alias is authenticated at a Safeguard terminal. *`prog-filename`* is the name of the command interpreter's object file. It must be a local file name.

If *`prog-filename`* is omitted, the other *`user-attribute`*s associated with CI-PROG *`prog-filename`* in this record are not meaningful.

`CI-LIB [`*`lib-filename`*`]`

changes the library file to be used with the command interpreter that is started when this alias is authenticated at a Safeguard terminal. *`lib-filename`* must be a local file name.

If you omit *`lib-filename`*, no library file is used.

`CI-CPU [`*`cpu-number`*` | ANY]`

changes the number of the CPU in which the command interpreter is to run. If you specify *ANY*, any CPU is used.

If you omit *`cpu-number`*, any CPU is used.

`CI-NAME [`*`process-name`*`]`

changes the process name to be assigned to the command interpreter specified by CI-PROG. *`process-name`* must be a local process name.

If you omit *process-name*, the Safeguard software generates a process name.

CI-SWAP [$*vol*[.*subvol*.*filename*]]

changes the name of the volume or file to be used as the swap volume or file for the command interpreter. $*vol* must be a local volume name. You can optionally supply a subvolume name and file name.

If you omit $*vol*, the same volume that contains the CI-PROG object file is used.

CI-PRI [*priority*]

changes the priority at which the command interpreter is to run.

If you omit *priority*, the value of CI-PRI in the Safeguard configuration record is used.

CI-PARAM-TEXT [*startup-param-text*]

changes the data to be supplied as the startup message text for the command interpreter specified by CI-PROG. If you specify the CI-PARAM-TEXT attribute, it must be the last attribute in the command string.

If you omit *startup-param-text*, the string is set to null. (No text is supplied in the startup message.)

WHERE *expression*

causes the command to apply to only those authentication records for aliases who belong to the groups specified by *expression*.

expression has the form:

group [{AND | OR} *group*] ...

group is one of:

```
GROUP [NAME]=group-name
GROUP NUMBER=group-num
PRIMARY-GROUP [NAME]=group-name
PRIMARY-GROUP NUMBER=group-num
```

Wild-card characters are not allowed in the group names or group numbers. Multiple groups within the *expression* can be enclosed within parentheses to change the order of evaluation of a complex expression, see Example 4.

group-name is case-sensitive. Therefore, you must enter alphabetic characters in an administrative group name in uppercase.

`RESET-TEXT-DESCRIPTION`

resets the text description field for this alias to a null value (the alias has no text description).

Note. The RESET-TEXT-DESCRIPTION field is supported only on systems running G06.27 and later G-series RVUs and H06.06 and later H-series.

`RESET-BINARY-DESCRIPTION`

resets the binary description field for this alias to zero length and null values. For more information about the binary description field, see the *Safeguard Management Programming Manual*.

Note. The RESET-BINARY-DESCRIPTION field is supported only on systems running G06.27 and later G-series RVUs and H06.06 and later H-series.

Considerations

- Changing your logon password

  Only the owner of an alias authentication record, the owner’s group manager and super ID can use the ALTER ALIAS command to change a password for an alias. However, users can change their alias password by logging on with the alias and changing the password during the logon dialog. (For more information, see the *Safeguard User’s Guide*.)

  A user logged on as an alias can also use the PASSWORD program to change the alias password.

  When you change your logon password, your Safeguard authentication record is automatically updated.

  Your password is subject to restrictions defined by the configuration attributes described in Section 16, Safeguard Subsystem Commands.

- Adding or deleting default protection while a user is logged on as an alias

  If you add or delete default protection for a user while that user is logged on as an alias, and the user subsequently creates a disk file during that session, the FUP INFO and TACL FILEINFO displays are not updated until the next time the disk file’s protection record is altered.

- Changing Guardian default security while an alias is logged on

  If you change the Guardian default disk file security for an alias while that alias is logged on, the change does not take effect until the next time the alias logs on or issues a Guardian VOLUME command.

Examples

1. The PRS group manager owns the alias authentication record for Admin_Darlene. The manager enters the following command to transfer ownership of that record to the user who has user ID 14,2 and to require that Darlene change the logon password for this alias every 35 days:

   ```
   =ALTER ALIAS Admin_Darlene, OWNER 14,2, &
   =PASSWORD-MUST-CHANGE EVERY 35 DAYS
   ```

   Because the OWNER attribute for Admin_Darlene was changed to a member of another group, PRS.MANAGER can no longer manage this authentication record.

2. The owner of the alias authentication record for fredX sets up Safeguard auditing for successful and failed authentication attempts (both local and remote) made under this alias:

   ```
   =ALTER ALIAS fredX, AUDIT-AUTHENTICATE-PASS all,&
   =AUDIT-AUTHENTICATE-FAIL all
   ```

3. The owner of the alias authentication record for BENNY1 alters the record so that the command interpreter EDITF starts automatically after BENNY1 logs on at a Safeguard terminal. The object program file for EDITF is $SALES.PROG2.EDITF.

   ```
   =ALTER ALIAS BENNY1, CI-PROG $sales.prog2.editf
   ```

4. The super ID alters the record of every alias that belongs to the group TEST1 and has a primary group number of 254 or 255. The AUDIT-USER-ACTION-FAIL attribute is set to ALL for each of these users.

   ```
   =ALTER ALIAS *, AUDIT-USER-ACT-FAIL all, &
   =WHERE GROUP=TEST1 AND (PRIMARY-GROUP NUMBER=254 &
   =OR PRIMARY-GROUP NUMBER=255)
   ```

DELETE ALIAS Command

DELETE ALIAS removes a user alias from a system and deletes the alias authentication record. After an alias is deleted, the user cannot log on to the system with that alias.

The primary owner of an alias authentication record, the secondary owners, the primary owner's group manager, and the super ID can delete an alias.

```
DELETE ALIAS { alias | ( alias [ , alias ] ... ) }

   [ [,] WHERE expression ]
```

ALIAS

specifies ALIAS as the object type of the DELETE command. Omit it if ALIAS is the assumed object type. (For more information about assumed types, see the ASSUME Command on page 4-3.)

`alias`

specifies the alias or aliases whose authentication records are to be deleted. *`alias`* is a text-string as defined under the ADD ALIAS command. The *`alias`* can contain wild-card characters.

`WHERE` *`expression`*

causes the DELETE command to apply only to authentication records for aliases who belong to the groups specified by *`expression`*. For a description of WHERE *`expression`*, see the ALTER ALIAS Command on page 6-9.

Considerations

- Deleting a user alias authentication record owner

  If the owner of an alias authentication record is deleted, only the group manager of the record owner or the local super ID can change the alias authentication record.

Examples

1. The following command deletes the alias clerk4:

   ```
   =DELETE ALIAS clerk4
   ```

2. The following command deletes all aliases that begin with Temp and are also members of the TEST group:

   ```
   =DELETE ALIAS Temp*, WHERE GROUP=TEST
   ```

3. The following commands assign the alias HeadBoss to a new underlying user ID 112,33. Because the underlying user ID cannot be altered directly in an alias authentication record, the alias must be deleted and re-created:

   ```
   =DELETE ALIAS HeadBoss
   =ADD ALIAS HeadBoss , 112,33, PASSWORD BiggiE
   ```

FREEZE ALIAS Command

FREEZE ALIAS temporarily suspends a user’s ability to log on to the system with the specified alias. You can later restore this ability through the THAW ALIAS command.

Only the primary owner of an alias authentication record, the secondary owners, the primary owner’s group manager, or the local super ID can freeze an alias authentication record. Depending on the value of the Safeguard AUTHENTICATE-FAIL-FREEZE configuration option, an alias can be automatically frozen. For details, refer to Section 16, Safeguard Subsystem Commands.

```
FREEZE ALIAS { alias | ( alias [ , alias ] ... ) }

   [ [,] WHERE expression ]
```

`ALIAS`

specifies ALIAS as the object type of the FREEZE command. Omit it if ALIAS is the assumed object type. (For more information about assumed types, see the ASSUME Command on page 4-3.)

`alias`

specifies the alias or aliases whose authentication records are to be frozen. *`alias`* is a text-string as defined under the ADD ALIAS command. The *`alias`* can contain wild-card characters.

`WHERE` *`expression`*

causes the FREEZE command to apply only to authentication records for aliases who belong to the groups specified by *`expression`*. For a description of WHERE *`expression`*, see the ALTER ALIAS Command on page 6-9.

Considerations

- Freezing an alias who is currently logged on

  Although an alias can be frozen while that user is logged on, freezing has no effect on the current command interpreter session of the alias. After logging off, however, the user cannot log on with that alias until the ability to log on is restored through the THAW ALIAS command.

- Logging on as a frozen alias

  Even if an alias is frozen, the local super ID (and the group manager of the alias) can log on as that alias. By default, the local super ID can log on as any user or alias defined for the system without supplying a password unless the configuration attribute PASSWORD-REQUIRED is set to ON. (See Section 16, Safeguard Subsystem Commands.)

Examples

1. The following FREEZE command suspends the alias clerk6 so that it cannot be used to log on to the system:

```
=FREEZE ALIAS clerk6
```

2. The following FREEZE command freezes all aliases that begin with Temp and are members of either the group TEST or the group Release:

```
=FREEZE ALIAS Temp*, WHERE GROUP=TEST OR GROUP=Release
```

INFO ALIAS Command

INFO ALIAS displays a report about the user attribute values currently stored in an alias authentication record.

Use of the INFO ALIAS command is limited to these users:

- The user assigned the alias
- The primary and secondary owners of the alias authentication record
- The primary owner’s group manager
- The super ID

Any alias of the user can execute the INFO USER command for any other alias of the user.

```
INFO [ / OUT listfile / ] ALIAS

   { alias | (  alias [ , alias ] ... ) }

   [ [ , ] option ] [ , option  ] ...
```

OUT *listfile*

directs SAFECOM output to *listfile* for the INFO report. (After executing the INFO command, SAFECOM redirects its output to the current OUT file.)

For *listfile*, specify any file name. SAFECOM opens *listfile* and appends the output text to it. If *listfile* does not exist, SAFECOM creates an EDIT-format file and writes the INFO report to that file.

ALIAS

specifies ALIAS as the object type of the INFO command. Omit it if ALIAS is the assumed type. (For more information about assumed types, see the ASSUME Command on page 4-3.)

alias

specifies the alias or aliases for whom an INFO report is to be produced. *alias* is a text-string as defined under the ADD ALIAS command. The *alias* can contain wild-card characters.

option

is one of:

```
GENERAL
DETAIL
AUDIT
CI
OSS
REMOTEPASSWORD
DEFAULT-PROTECTION
GROUP
OWNER-LIST
```

```
TEXT-DESCRIPTION
WHERE expression
```

`GENERAL`

displays the basic user attributes including UID, password settings, user expiration, Guardian security, and Guardian default volume.

`DETAIL`

displays all attributes, including those displayed by all other *`option`*s.

`AUDIT`

displays only attributes related to auditing.

`CI`

displays only attributes related to the default command interpreter.

`OSS`

displays only attributes related to OSS initial settings.

`REMOTEPASSWORD`

displays all remote passwords defined for this alias.

`DEFAULT-PROTECTION`

displays only attributes related to default disk file protection.

`GROUP`

displays only the group list and primary group for the alias.

`OWNER-LIST`

displays the secondary owners of the alias authentication record.

`TEXT-DESCRIPTION`

displays the descriptive text associated with the user’s authentication record.

`WHERE` *`expression`*

causes information to be displayed only for aliases who belong to the groups specified by *`expression`*. For a description of WHERE *`expression`*, see the ALTER ALIAS Command on page 6-9.

INFO ALIAS Brief Report

Figure 6-1shows the format of the brief INFO ALIAS report. A description of the user-attribute values and status fields immediately follows it.

Figure 6-1. INFO ALIAS Brief Report Format

```
NAME                          USER-ID  OWNER      STATUS
alias                         u-id     o-id   [+] status
```

NAME
alias

is the user alias whose current user attributes are being displayed.

USER-ID
u-id

is the structured view of the user ID of the user associated with this alias.

OWNER
o-id

is the user ID of the user who is the primary owner of this alias authentication record. If *o-id* is the network form of a user ID, the primary owner is a network user.

[+]

indicates the existence of an OWNER-LIST for the alias authentication record. The plus sign does not appear if no OWNER-LIST exists. You can specify the OWNER-LIST keyword to display a list of owners.

STATUS
status

indicates the current status of this alias. *status* can be any of:

USER-EXP The user's ability to log on to the system with this alias has expired. Until the USER-EXPIRES date is changed to some future date, the user cannot log on to the system with this alias.

PSWD-EXP The password for this alias has expired. Until the password is changed or until the PASSWORD-MUST-CHANGE period is extended (through the ALTER ALIAS command), the user cannot log on to the system with this alias. The PASSWORD-EXPIRES attribute can also be changed directly with the ALTER ALIAS command.

FROZEN The user's ability to log on to the system with this alias has been frozen. Until the owner of the alias authentication record or the owner's group manager restores this ability through the THAW ALIAS command, the user cannot log on to the system with this alias.

THAWED The user can log on to the system with this alias.

The values of the *status* field are listed in the order of their priority. When two or more of the conditions described by a *status* value apply to a user alias, only the highest priority is displayed. For example, if a password is expired and the alias is frozen, *status* is displayed as PSWD-EXP.

INFO ALIAS Detailed Report

Figure 6-2 on page 6-28 shows the format of the detailed INFO ALIAS report.

Figure 6-2. INFO ALIAS Detailed Report Format

```
NAME                          USER-ID  OWNER     STATUS
alias                         u-id     o-id  [+] status

  UID                        = u-id
  USER-EXPIRES               = date, time   [-- EXPIRED --]
  PASSWORD-EXPIRES           = date, time   [-- EXPIRED --]
  PASSWORD-MAY-CHANGE        = date, time   [-- EXPIRED --]
  PASSWORD-MUST-CHANGE EVERY = num  DAYS
  PASSWORD-EXPIRY-GRACE      = num  DAYS
  LAST-LOGON                 = date, time
  LAST-UNSUCESSFUL-ATTEMPT   =   * NONE *
  LAST-MODIFIED              = date, time
  CREATION-TIME              = date, time
  FROZEN/THAWED              = FROZEN | THAWED
  STATIC FAILED LOGON COUNT  = count
  STATIC-FAILED-LOGON-RESET  =   * NONE *
  GUARDIAN DEFAULT SECURITY  = string
  GUARDIAN DEFAULT VOLUME    = $vol.subvol

  CREATOR-USER-NAME          = user-name/alias-name
  CREATOR-USER-TYPE          = USER/ALIAS ( uid )
  CREATOR-NODENUMBER         = num


  AUDIT-AUTHENTICATE-PASS  = a-spec     AUDIT-MANAGE-PASS  = a-spec
  AUDIT-AUTHENTICATE-FAIL  = a-spec     AUDIT-MANAGE-FAIL  = a-spec
  AUDIT-USER-ACTION-PASS   = a-spec
  AUDIT-USER-ACTION-FAIL   = a-spec

  TEXT-DESCRIPTION = ["text"]

  BINARY-DESCRIPTION-LENGTH = length

  CI-PROG = [prog-filename]
  CI-LIB  = [lib-filename]]
  CI-NAME = [process-name
  CI-SWAP = [$vol[.subvol.filename]]
  CI-CPU  = {num | ANY}
  CI-PRI  = [num]
  CI-PARAM-TEXT = [text]

  INITIAL-PROGTYPE      = prog-type
  INITIAL-PROGRAM       = [prog-path]
  INITIAL-DIRECTORY     = [dir-path]

  PRIMARY-GROUP  = group
  GROUP          = group

 [REMOTEPASSWORD = \system remotepassword]

SUBJECT DEFAULT PROTECTION SECTION

SUBJECT OWNER-LIST SECTION
```

In addition to the user attributes and status fields displayed in the brief INFO ALIAS report, the detailed INFO ALIAS report also displays these user attributes and status fields:

`UID =` *`u-id`*

is the scalar view of the user ID of the user associated with this alias.

`LAST-LOGON =` *`date`*`,`*`time`*

is the time and date when the user last logged onto the system with this alias (in local civil time).

`LAST-MODIFIED =` *`date`*`,`*`time`*

is the time and date when this alias authentication record was last changed (in local civil time).

`USER-EXPIRES =` *`date`*`,` *`time`*

is the date and time when the user’s ability to log on to the system with this alias will be suspended (in local civil time). After the user’s ability to log on to the system with this alias has expired, changing the USER-EXPIRES attribute to some future date restores that ability.

`PASSWORD-EXPIRES =` *`date`*`,` *`time`*

is the date and time when the password expires for this alias. Whenever the password is changed through the ALTER ALIAS command or the Safeguard logon dialog, the Safeguard software calculates a new PASSWORD-EXPIRES date by adding the number of days specified in the alias PASSWORD-MUST-CHANGE attribute to the date of the password change. The Safeguard software also calculates a new PASSWORD-EXPIRES date whenever the alias PASSWORD-MUST-CHANGE attribute is changed.

Immediate expiration of the alias password can also be specified with PASSWORD-EXPIRES attribute.

After the password expires, the user cannot log on to the system with this alias until either the password is changed or the alias PASSWORD-MUST-CHANGE period is extended. (If the alias has an extension period established with PASSWORD-EXPIRY-GRACE, the user can log on with that alias to change the expired password.)

`date` and *`time`* are given in local civil time.

`PASSWORD-MAY-CHANGE =`*`date`*`,` *`time`*

specifies the date and time that this alias password can be changed.

`PASSWORD-MUST-CHANGE EVERY =` *`num`* `DAYS`

specifies the maximum number of days that this alias can use the same password.

`PASSWORD-EXPIRY-GRACE =` *`num`* ` DAYS`

specifies the number of days after password expiration that the alias password can be changed during logon.

`FROZEN/THAWED =` *`frozen`* `|` *`thawed`*

indicates whether or not a user's access to the system with this alias has been frozen. While the alias is frozen, the user cannot log on to the system with this alias.

`CREATION-TIME  = date, time`

specifies the date and time when the user was created.

`STATIC FAILED LOGON COUNT =` *`count`*

is the number of total unsuccessful logon attempts made with this alias since it was created.

`STATIC-FAILED-LOGON-RESET = * NONE *`

specifies the last time when the value of the attribute, STATIC FAILED LOGON COUNT, was reset.

`GUARDIAN DEFAULT SECURITY =` *`string`*

is the Guardian default security string for this alias.

`GUARDIAN DEFAULT VOLUME = $`*`vol.subvol`*

is the Guardian default subvolume for this alias.

`CREATOR-USER-NAME = user-name/alias-name`

specifies the username of the user who created the user.

`CREATOR-USER-TYPE  = USER/ALIAS ( uid )`

identifies if the creator is an alias or a user, followed by the user ID of the creator.

`CREATOR-NODENUMBER = num`

specifies the system number where the user is created.

```
AUDIT-AUTHENTICATE-PASS = a-spec   AUDIT-MANAGE-PASS = a-spec
AUDIT-AUTHENTICATE-FAIL = a-spec   AUDIT-MANAGE-FAIL = a-spec
AUDIT-USER-ACTION-PASS  = a-spec
AUDIT-USER-ACTION-FAIL  = a-spec
```

indicate the conditions under which the Safeguard software is to audit attempts to log on with this alias name, attempts to manage the alias authentication record, and attempts by the user to perform an event while logged on as this alias. *a-spec* can be:

`{ ALL | LOCAL | REMOTE | NONE }`

For a full description of *a-spec*, see the *audit-spec* for the SET ALIAS command.

`TEXT-DESCRIPTION = ["`*text*`"]`

is the descriptive text associated with the alias.

`BINARY-DESCRIPTION-LENGTH = `*length*

is the length in bytes of the binary description field for the alias. If no binary description was specified for the alias, *length* is 0.

`CI-PROG = [ `*prog-filename*` ]`

is the object file name of the command interpreter started after the user logs on at a Safeguard terminal with this alias.

`CI-LIB = [ `*lib-filename*` ]`

is the file name of the library file used with the command interpreter.

`CI-NAME = [ `*process-name*` ]`

is the process name assigned to the command interpreter.

`CI-SWAP = [ $`*vol*`[`*.subvol.filename*`] ]`

is the swap volume or file used with the command interpreter.

`CI-CPU = `*num*` | ANY`

is the number of the CPU in which the command interpreter runs. ANY indicates any CPU.

`CI-PRI = [ `*num*` ]`

is the priority at which the command interpreter runs.

`CI-PARAM-TEXT = [ `*text*` ]`

is the startup parameter text supplied to the command interpreter.

`INITIAL-PROGTYPE = `*prog-type*

is the initial program type: PROGRAM, WINDOW, or SERVICE.

`INITIAL-PROGRAM = [ `*prog-path*` ]`

is the initial program pathname for the OSS file system. It is blank if no pathname is specified.

`INITIAL-DIRECTORY = [` *`dir-path`* `]`

is the initial directory pathname. It is blank if no pathname is defined.

`PRIMARY-GROUP =` *`group`*

is the group name of the primary group for this alias.

`GROUP =` *`group`*

is the group name of each group in the alias group list. Groups in this list are specified by the MEMBER attribute of the ADD or ALTER GROUP commands.

`[ REMOTEPASSWORD = \`*`system-name  remotepassword`* `]`

is a remote password defined for the specified system name.

When \system-name appears as \??????, the remote password is defined for a system number that is no longer assigned to a system on the network.

When *remotepassword* appears as ;;;;;;;;, the remote password contains one or more unprintable characters.

REMOTEPASSWORD does not appear if no remote passwords are defined for the alias.

`SUBJECT DEFAULT PROTECTION SECTION`

shows the default protection assigned to the user's disk files when they are added to the Safeguard database.

`OWNER-LIST-SECTION`

is a list of the secondary owners of the user's authentication record.

Examples

1. This example of the INFO ALIAS command displays the user attributes for the alias fredX before and after the alias is frozen:

```
=INFO ALIAS fredX
```

```
NAME                           USER-ID  OWNER     STATUS
fredX                           86,21   86,255    THAWED
```

```
=FREEZE ALIAS fredX
=INFO ALIAS fredX
```

```
NAME                           USER-ID  OWNER     STATUS
fredX                           86,21   86,255    FROZEN
```

2. This command displays the group list for each alias that is a member of the group Rev40:

   ```
   =INFO ALIAS *, GROUP, WHERE GROUP=Rev40
   ```

RESET ALIAS Command

RESET ALIAS resets the current default values for user attributes to predefined values. (The predefined reset values are the values of the default user attributes when you begin a SAFECOM session.)

When you add an alias to your system, the current default user attribute values are used for any attributes you do not specify in the ADD ALIAS command. (Use the SET ALIAS command to set the default user attribute values to specific values.)

```
RESET ALIAS [ [ , ] user-attribute-keyword ]

   [ , user-attribute-keyword ] ...
```

`RESET ALIAS`

entered with no *`user-attribute-keyword`*, resets all current default user attribute values to their predefined values.

`ALIAS`

specifies ALIAS as the object type of the RESET command. Omit it if ALIAS is the assumed type. (For more information, see the ASSUME Command on page 4-3.)

`user-attribute-keyword`

sets the current default value of the specified user attribute to a predefined value. *`user-attribute-keyword`* can be:

```
OWNER
OWNER-LIST
PASSWORD
USER-EXPIRES
PASSWORD-MUST-CHANGE
PASSWORD-EXPIRY-GRACE
PASSWORD-EXPIRES
AUDIT-AUTHENTICATE-PASS
AUDIT-AUTHENTICATE-FAIL
AUDIT-MANAGE-PASS
AUDIT-MANAGE-FAIL
AUDIT-USER-ACTION-PASS
AUDIT-USER-ACTION-FAIL
TEXT-DESCRIPTION
REMOTEPASSWORD
DEFAULT-PROTECTION [obj-attr]
GUARDIAN [DEFAULT] SECURITY
GUARDIAN [DEFAULT] [SUB]VOLUME
INITIAL-DIRECTORY
```

```
INITIAL-PROGRAM
INITIAL-PROGTYPE
CI-PROG
CI-LIB
CI-CPU
CI-NAME
CI-SWAP
CI-PRI
CI-PARAM-TEXT
```

The predefined values for the attributes are:

`OWNER`

`owner-id` is set to the user ID of the current SAFECOM user.

`OWNER-LIST`

`user-list` is set to null (no secondary owners).

`PASSWORD`

`password` is set to null (no password required to log on).

`USER-EXPIRES`

`date`,*`time`* are set to null (no expiration date).

`PASSWORD-MUST-CHANGE`

`num` days is set to null. (The alias password never has to be changed.)

`PASSWORD-EXPIRY-GRACE`

`num` days is set to null. (There is no extension period during which to change an expired alias password.)

`PASSWORD-EXPIRES`

`date`,*`time`* are set to null (no expiration date).

`AUDIT-AUTHENTICATE-PASS`

`audit-spec` is set to NONE.

`AUDIT-AUTHENTICATE-FAIL`

`audit-spec` is set to NONE.

`AUDIT-MANAGE-PASS`

`audit-spec` is set to NONE.

`AUDIT-MANAGE-FAIL`

`audit-spec` is set to NONE.

`AUDIT-USER-ACTION-PASS`

`audit-spec` is set to NONE.

`AUDIT-USER-ACTION-FAIL`

`audit-spec` is set to NONE.

`TEXT-DESCRIPTION`

`text` is set to null (no descriptive text).

`REMOTEPASSWORD`

The remote password list is set to null (no remote passwords).

`DEFAULT-PROTECTION`

The default protection record is set to null. (New files remain under Guardian protection until explicitly added to the Safeguard database.)

`GUARDIAN [DEFAULT] SECURITY`

`string` is set to “OOOO.”

`GUARDIAN [DEFAULT] [SUB]VOLUME`

$*`vol.subvol`* is set to $SYSTEM.NOSUBVOL.

`INITIAL-DIRECTORY`

`dir-path` is set to null (no pathname).

`INITIAL-PROGRAM`

`prog-path` is set to null (no pathname).

`INITIAL-PROGTYPE`

`prog-type` is set to PROGRAM.

`CI-PROG`

`prog-filename` is set to null (no command interpreter in this alias record).

`CI-LIB`

`lib-filename` is set to null (no library file).

CI-CPU

cpu-number is set to ANY.

CI-NAME

process-name is set to null. (The Safeguard software generates a name.)

CI-SWAP

$*vol* is set to null. (Use same volume as CI-PROG object file).

CI-PRI

priority is set to null. (Use the value of CI-PRI in the Safeguard configuration record.)

CI-PARAM-TEXT

startup-param-text is set to null. (No data is supplied in startup message text.)

Examples

To restore the current default user attributes (set in previous SET ALIAS commands) to their predefined values, you can first enter the SHOW ALIAS commands to display the current user attributes:

```
=SHOW ALIAS
```

The report shows:

```
TYPE        OWNER
 ALIAS      86,2

  PASSWORD =
  USER-EXPIRES               =    * NONE *
  PASSWORD-EXPIRES           =    * NONE *
  PASSWORD-MUST-CHANGE EVERY =    30 DAYS
  PASSWORD-EXPIRY-GRACE      =    * NONE *
  GUARDIAN DEFAULT SECURITY  = OOOO
  GUARDIAN DEFAULT VOLUME    = $DATA2.FRED

  AUDIT-AUTHENTICATE-PASS  = ALL          AUDIT-MANAGE-PASS  = REMOTE
  AUDIT-AUTHENTICATE-FAIL  = ALL          AUDIT-MANAGE-FAIL  = NONE
  AUDIT-USER-ACTION-PASS   = NONE
  AUDIT-USER-ACTION-FAIL   = NONE

  TEXT-DESCRIPTION = "ALIAS FOR USER1"

  CI-PROG = * NONE *
  CI-LIB  = * NONE *
  CI-NAME = * NONE *
  CI-SWAP = * NONE *
  CI-CPU  = ANY
  CI-PRI  = * NONE *
  CI-PARAM-TEXT =

  INITIAL-PROGTYPE      = PROGRAM
  INITIAL-PROGRAM       =
  INITIAL-DIRECTORY     =

 SUBJECT DEFAULT-PROTECTION SECTION UNDEFINED!

 SUBJECT OWNER-LIST SECTION UNDEFINED!
```

Then execute the following RESET command:

```
=RESET ALIAS
```

Finally, enter another SHOW ALIAS command to display the attributes that have been RESET:

```
=SHOW ALIAS
```

The report shows:

```
TYPE        OWNER
 ALIAS      86,255

  PASSWORD =
  USER-EXPIRES                 =    * NONE *
  PASSWORD-EXPIRES             =    * NONE *
  PASSWORD-MUST-CHANGE EVERY =    * NONE *
  PASSWORD-EXPIRY-GRACE        =    * NONE *
  GUARDIAN DEFAULT SECURITY    = OOOO
  GUARDIAN DEFAULT VOLUME      = $SYSTEM.NOSUBVOL

  AUDIT-AUTHENTICATE-PASS   = NONE         AUDIT-MANAGE-PASS   = NONE
  AUDIT-AUTHENTICATE-FAIL   = NONE         AUDIT-MANAGE-FAIL   = NONE
  AUDIT-USER-ACTION-PASS    = NONE
  AUDIT-USER-ACTION-FAIL    = NONE

  TEXT-DESCRIPTION =

  CI-PROG = * NONE *
  CI-LIB  = * NONE *
  CI-NAME = * NONE *
  CI-SWAP = * NONE *
  CI-CPU  = ANY
  CI-PRI  = * NONE *
  CI-PARAM-TEXT =

  INITIAL-PROGTYPE      = PROGRAM
  INITIAL-PROGRAM       =
  INITIAL-DIRECTORY     =

 SUBJECT DEFAULT-PROTECTION SECTION UNDEFINED!

 SUBJECT OWNER-LIST SECTION UNDEFINED!
```

SET ALIAS Command

SET ALIAS establishes current default values for one or more user attributes. Later, when you add an alias to your system, these current default values are used for any attribute you do not specify in the ADD ALIAS command.

After you set the current default values, use the SHOW ALIAS command to display them before you add an alias through ADD ALIAS.

```
SET ALIAS [ , ] { LIKE user | user-attribute }

   [ , user-attribute ] ...
```

ALIAS

specifies ALIAS as the object type of the SET command. Omit it if ALIAS is the assumed type. (For more information, see the ASSUME Command on page 4-3.)

LIKE *user*

sets the current default user attribute values to the same as those currently defined for the user or alias specified with *user*.

user is an existing user specified in one of these formats:

```
alias
group-num,member-num
group-name.member-name
```

LIKE sets the current default values for all user attributes except:

```
PASSWORD [password]
REMOTEPASSWORD \system-name remote-password
GUARDIAN [DEFAULT] SECURITY ["]string["]
GUARDIAN [DEFAULT] [SUB]VOLUME [\system.]$vol.subvol
CI-NAME [process-name]
PRIMARY-GROUP [ [ NAME ] group-name | NUMBER group-num ]
INITIAL-DIRECTORY [dir-path]
INITIAL-PROGRAM [prog-path]
INITIAL-PROGTYPE [prog-type]
TEXT-DESCRIPTION "[text]"
```

user-attribute

establishes a current default user-attribute value to be used in subsequent ADD ALIAS commands. The user attributes are:

```
OWNER [owner-id]
OWNER-LIST [[-]user-list]
PASSWORD [password]
USER-EXPIRES [ date [ , time] ]
PASSWORD-MUST-CHANGE [EVERY num DAYS]
PASSWORD-EXPIRY-GRACE [num [DAYS]]
PASSWORD-EXPIRES [ date [ , time] ]
AUDIT-AUTHENTICATE-PASS [audit-spec]
AUDIT-AUTHENTICATE-FAIL [audit-spec]
AUDIT-MANAGE-PASS [audit-spec]
AUDIT-MANAGE-FAIL [audit-spec]
AUDIT-USER-ACTION-PASS [audit-spec]
AUDIT-USER-ACTION-FAIL [audit-spec]
TEXT-DESCRIPTION "[text]"
REMOTEPASSWORD \system-name remote-password
DEFAULT-PROTECTION [ obj-attr ]
                   [ ( obj-attr [ , obj-attr ] ...) ]
GUARDIAN [DEFAULT] SECURITY ["]string["]
GUARDIAN [DEFAULT] [SUB]VOLUME [\system.]$vol.subvol
INITIAL-DIRECTORY [dir-path]
INITIAL-PROGRAM [prog-path]
INITIAL-PROGTYPE [prog-type]
CI-PROG [prog-filename]
CI-LIB [lib-filename]
CI-CPU [cpu-number | ANY]
CI-NAME [process-name]
CI-SWAP [$vol.[subvol.filename]]
CI-PRI [priority]
CI-PARAM-TEXT [startup-param-text]
```

```
OWNER [owner-id]
```

specifies the primary owner of an alias authentication record. For *owner-id*, specify either of:

```
[\*.]group-name.member-name
[\*.]group-num , member-num
```

If you omit *owner-id*, your user ID becomes the current *owner-id*.

```
OWNER-LIST [[-]user-list]
```

changes the secondary ownership of an alias authentication record by adding or deleting owners in the owner list. A minus sign (-) preceding *user-list* indicates that the specified users are to be deleted from the existing owner list. If the minus sign is omitted, the specified users are added to the owner list. If *user-list* is omitted, the owner list is set to null (no secondary owners) A maximum of 50 users can be specified in *user-list*. For *user-list*, specify either:

```
net-user-spec
(net-user-spec [, net-user-spec ...])

net-user-spec is either:


[\node-spec.]group-name.user-name
[\node-spec.]group-num , user-num

node-spec is one of:


*
node-name
node-number
```

node-name

specifies the system name.

node-number

specifies the Expand node number.

```
PASSWORD [password]
```

specifies a logon password for the alias. Typically, users must enter their alias and a password to log on to a system.

For *password*, specify the logon password, which can be one to eight characters long. Use any alphanumeric characters except blanks, commas, semicolons, and the ASCII null character. The case of letters in a password is preserved. Lowercase letters remain lowercase, and uppercase letters remain uppercase.

If you omit *password*, the value for *password* is set to null. (No password is required for logon.)

```
USER-EXPIRES [ date [ , time] ]
```

establishes a date and time after which a user cannot log on to the system with this alias. Specify *date* and *time* as local civil time.

If you omit both *date* and *time*, the user-expiration attribute value is set to null (no expiration date).

If omitted, *time* is set to 0:00 (midnight).

The form of *date* [, *time*] is:

```
{ month-name day } year [,hour:min]
{ day month-name }
```

month-name

is the first three letters of the month name: JAN, FEB, MAR, and so on. You can use either uppercase or lowercase letters.

day

is a 1-digit or 2-digit integer from 1 through 31.

year

is a 4-digit integer.

hour

is an integer from 0 through 23.

min

is an integer from 0 through 59.

```
PASSWORD-MUST-CHANGE [EVERY num DAYS]
```

specifies the maximum number of days that the same password can be used. For *num*, specify an integer from 1 through 32,767.

When you add an alias with a PASSWORD-MUST-CHANGE attribute, the Safeguard software calculates a PASSWORD-EXPIRES date by adding *num* days to the current date. If the password is not changed before the PASSWORD-EXPIRES date, the user cannot log on to the system with that alias after that date. Each time the password is changed, the Safeguard software calculates a new PASSWORD-EXPIRES date by adding *num* days to the date of the password change.

Omitting the EVERY *num* DAYS clause disables PASSWORD-MUST-CHANGE. (That is, the password never expires unless the PASSWORD-EXPIRES attribute is set.)

`PASSWORD-EXPIRY-GRACE` *`num`* `[DAYS]`

specifies the number of days after password expiration during which the password for this alias can be changed during logon. For *num*, specify an integer from 0 through 32,767. A value of 0 means no extension period.

Omitting *num* specifies that the value of PASSWORD-EXPIRY-GRACE in the Safeguard configuration record is to be used to determine the extension period. In this instance, the value *NONE* appears in this field of the alias protection record.

`PASSWORD-EXPIRES [` *`date`* `[ ,` *`time`*`] ]`

establishes a date and time after which the password expires for this alias. Specify *date* and *time* as local civil time.

If you omit both *date* and *time*, no expiration is set for the password. (However, an expiration date is calculated and set if a PASSWORD-MUST-CHANGE period is subsequently specified or altered.)

If omitted, *time* is set to 0:00 (midnight).

The form of *date* [, *time*] is:

```
{ month-name day } year [,hour:min]
{ day month-name }
```

month-name

is the first three letters of the month name: JAN, FEB, MAR, and so on. You can use either uppercase or lowercase letters.

day

is a 1-digit or 2-digit integer from 1 through 31.

year

is a 4-digit integer.

hour

is an integer from 0 through 23.

min

is an integer from 0 through 59.

`AUDIT-AUTHENTICATE-PASS [`*`audit-spec`*`]`

establishes an *`audit-spec`* for successful user authentication attempts. The *`audit-spec`* specifies the conditions under which the Safeguard software writes an audit record to the audit file when the user successfully logs on to the system with this alias.

The form of *`audit-spec`* is:

```
{ ALL | LOCAL | REMOTE | NONE }
```

`ALL`

All successful logons are audited.

`LOCAL`

Only successful logons from the local system are audited.

`REMOTE`

This form has no effect. Remote authentication is not supported.

`NONE`

No successful logons are audited.

Omitting *`audit-spec`* specifies NONE.

Note. In prior product versions of the Safeguard software, the AUDIT-AUTHENTICATE user attributes were called AUDIT-ACCESS. The user attribute name AUDIT-ACCESS is still supported, but HP discourages its use.

`AUDIT-AUTHENTICATE-FAIL [`*`audit-spec`*`]`

establishes an *`audit-spec`* for unsuccessful user authentication attempts. The *`audit-spec`* specifies the conditions under which an audit record is written to the audit file if the user fails to log on properly using this alias.

The form of *`audit-spec`* is:

```
{ ALL | LOCAL | REMOTE | NONE }
```

`ALL`

All unsuccessful logons are audited.

`LOCAL`

Only unsuccessful logons from the local system are audited.

`REMOTE`

This form has no effect. Remote authentication is not supported.

`NONE`

No unsuccessful logons are audited.

Omitting *`audit-spec`* specifies NONE.

`AUDIT-MANAGE-PASS [`*`audit-spec`*`]`

establishes an *`audit-spec`* for successful attempts to manage the alias authentication record. The *`audit-spec`* specifies the conditions under which an audit record is written to the audit file when the alias authentication record is managed.

The form of *`audit-spec`* is:

```
{ ALL | LOCAL | REMOTE | NONE }
```

`ALL`

All successful management attempts are audited.

`LOCAL`

Only successful management attempts from the local system are audited.

`REMOTE`

Only successful management attempts from remote systems are audited.

`NONE`

No successful management attempts are audited.

Omitting *`audit-spec`* specifies NONE.

`AUDIT-MANAGE-FAIL [`*`audit-spec`*`]`

establishes an *`audit-spec`* for unsuccessful attempts to manage the alias authentication record. The *`audit-spec`* specifies the conditions under which an audit record is written to the audit file when somebody tries, but fails, to manage the alias authentication record.

The form of *`audit-spec`* is:

```
{ ALL | LOCAL | REMOTE | NONE }
```

`ALL`

All unsuccessful management attempts are audited.

`LOCAL`

Only unsuccessful management attempts from the local system are audited.

`REMOTE`

Only unsuccessful management attempts from a remote system are audited.

`NONE`

No unsuccessful management attempts are audited.

Omitting *`audit-spec`* specifies NONE.

`AUDIT-USER-ACTION-PASS [`*`audit-spec`*`]`

establishes an *`audit-spec`* for successful events performed by the user logged on with this alias, including attempts to access objects and attempts to create or manage Safeguard protection records.

The *`audit-spec`* specifies the conditions under which the Safeguard software writes an audit record to the audit file when the alias successfully performs an event.

Note. When the SAFEGUARD global configuration attributes AUDIT-CLIENT-OSS and AUDIT-OSS-FILTER are enabled, the AUDIT-USER-ACTION-PASS attribute takes effect for OSS auditing.

The form of *`audit-spec`* is:

```
{ ALL | LOCAL | REMOTE | NONE }
```

`ALL`

All successful events are audited.

`LOCAL`

Only successful events on the local system are audited.

`REMOTE`

Only successful events by a remote user are audited.

`NONE`

No successful events are audited.

Omitting *`audit-spec`* specifies NONE.

`AUDIT-USER-ACTION-FAIL [`*`audit-spec`*`]`

establishes an *`audit-spec`* for unsuccessful events attempted by the user logged on with this alias, including attempts to access objects and attempts to create or manage Safeguard protection records.

The *`audit-spec`* specifies the conditions under which the Safeguard software writes an audit record to the audit file when the alias unsuccessfully attempts to perform an event.

Note. When the SAFEGUARD global configuration attributes AUDIT-CLIENT-OSS and AUDIT-OSS-FILTER are enabled, the AUDIT-USER-ACTION-FAIL attribute takes effect for OSS auditing.

The form of *`audit-spec`* is:

`{ ALL | LOCAL | REMOTE | NONE }`

`ALL`

All unsuccessful events are audited.

`LOCAL`

Only unsuccessful events on the local system are audited.

`REMOTE`

Only unsuccessful events by a remote user are audited.

`NONE`

No unsuccessful events are audited.

Omitting *`audit-spec`* specifies NONE.

`TEXT-DESCRIPTION "[`*`text`*`]"`

specifies a string of descriptive text to be associated with the alias. The text must consist of printable characters. This attribute is provided for documentation purposes only and has no effect on the alias record. All text between the quotation marks is considered to be descriptive text.

Because SAFECOM allows a maximum command length of 528 characters, the specified string must contain less than 528 characters. You can specify a longer descriptive text string by using the Safeguard SPI command interface, as described in the *Safeguard Management Programming Manual*.

If you omit *`text`*, no descriptive text is included in the alias authentication record.

`REMOTEPASSWORD \`*`system-name remote-password`*

establishes a remote password for a local alias:

`\`*`system-name`*

is the system for which the following remote password is to be assigned. `\`*`system-name`* must be a valid system name.

remote-password

is the remote password assigned to \system-name. For *remote-password*, specify a string of one to eight characters. Any character can be used in a remote password except blanks, commas, semicolons, and the ASCII null character. The case of letters is preserved. Lowercase letters remain lowercase, and uppercase letters remain uppercase. Only one remote password can be set with a SET command.

Note. Use RESET ALIAS REMOTEPASSWORD to clear a default remote password that you previously established with the SET command.

```
DEFAULT-PROTECTION [ obj-attr ]
                   [ ( obj-attr [ ,obj-attr ] ...) ]
```

specifies one or more attributes to be assigned immediately to new disk files created by this alias. If you omit *obj-attr*, new disk files remain under Guardian protection. If any *obj-attr* is specified, the attribute updates the current default protection record.

obj-attr

is one of:

```
OWNER [ owner-id ]
ACCESS [ access-spec [ ; access-spec ] ... ]
AUDIT-ACCESS-PASS [ audit-spec ]
AUDIT-ACCESS-FAIL [ audit-spec ]
AUDIT-MANAGE-PASS [ audit-spec ]
AUDIT-MANAGE-FAIL [ audit-spec ]
```

For more information about these object attributes as they apply to disk files, see Section 8, Disk-File Security Commands.

```
GUARDIAN [DEFAULT] SECURITY ["]string["]
```

specifies the Guardian default disk file security string for the alias. The word DEFAULT is optional, as are the quotes that surround the security string specifier. You can include them in the command for readability. *string* is a four-character string that specifies the Guardian default security string. Each position in the string can contain one of these characters: O, U, G, C, A, or N.

If no GUARDIAN SECURITY is specified, the default Guardian security string is set to “OOOO.”

For more information about Guardian default file-security string, see the *Safeguard User’s Guide*.

```
GUARDIAN [DEFAULT] [SUB]VOLUME [\system.]$vol.subvol
```

specifies the Guardian default subvolume. The word DEFAULT and the prefix SUB are optional. You can include them in the command for readability. \system is also optional. If you omit \system, the current system is assumed.

$*vol* specifies the default volume, and *subvol* specifies the default subvolume.

If no GUARDIAN VOLUME is specified, the default subvolume is set to $SYSTEM.NOSUBVOL.

`INITIAL-DIRECTORY [`*`dir-path`*`]`

specifies the initial working directory within the OSS file system for the alias. *dir-path* is a case-sensitive text string of up to 256 characters. It must be a syntactically valid OSS pathname. If you specify the INITIAL-DIRECTORY attribute, it must be the last attribute in the command string.

If you omit *dir-path*, no pathname is used.

`INITIAL-PROGRAM [`*`prog-path`*`]`

specifies the initial program pathname within the OSS environment for the alias. *prog-path* is a case-sensitive text string of up to 256 characters. It must be a syntactically valid OSS pathname. If you specify the INITIAL-DIRECTORY attribute, it must be the last attribute in the command string.

If you omit *prog-path*, no pathname is used.

`INITIAL-PROGTYPE [`*`prog-type`*`]`

specifies the initial program type within the OSS environment for the alias.

prog-type

is one of these:

```
PROGRAM
SERVICE
WINDOW
```

If you omit *prog-type*, PROGRAM is used.

This feature is not currently implemented on NonStop systems. It is reserved for future use.

`CI-PROG [`*`prog-filename`*`]`

specifies the command interpreter to be started after this alias is authenticated at a Safeguard terminal. *prog-filename* is the name of the command interpreter’s object file. It must be a local file name.

If you omit *prog-filename*, the other user attributes associated with CI-PROG in this record are not meaningful.

If CI-PROG *prog-filename* is omitted in the alias authentication record, the Safeguard software starts the PROG (with associated parameters) in the definition record for the terminal at which the alias logs on. If no PROG is specified in the terminal definition record, the Safeguard software starts the CI-

PROG (with associated parameters) specified in the Safeguard configuration record.

`CI-LIB [`*`lib-filename`*`]`

specifies the library file to be used with the command interpreter started when this alias is authenticated at a Safeguard terminal. *`lib-filename`* must be a local file name.

If you omit *`lib-filename`*, no library file is used.

`CI-CPU [`*`cpu-number`*` | ANY]`

specifies the number of the CPU in which the command interpreter is to run. If you specify *ANY*, any CPU is used.

If you do not specify *`cpu-number`*, any CPU is used.

`CI-NAME [`*`process-name`*`]`

specifies the process name to be assigned to the command interpreter specified by CI-PROG.

If you omit *`process-name`*, Safeguard assigns a process name. *`process-name`* must be a local process name.

`CI-SWAP [$`*`vol.`*`[`*`subvol.filename`*`]]`

specifies the name of the volume or file to be used as the swap volume or file for the command interpreter. $*`vol`* must be a local volume name. You can optionally supply a subvolume name and file name.

Omitting $*`vol`* results in use of the same volume that contains the CI-PROG object file.

`CI-PRI [`*`priority`*`]`

specifies the priority at which the command interpreter is to run.

If you omit *`priority`*, the value of CI-PRI in the Safeguard configuration record is used.

`CI-PARAM-TEXT [`*`startup-param-text`*`]`

specifies the data to be supplied as the startup message text for the command interpreter specified by CI-PROG. If you specify the CI-PARAM-TEXT attribute, it must be the last attribute in the command string.

If you omit *`startup-param-text`*, no startup parameter text is used.

Considerations

- An expired alias cannot be used to log on.

When access for an alias expires, the user cannot log on to the system with that alias, but the alias authentication record is not deleted.

- GUARDIAN DEFAULT attributes are equivalent to using the Guardian DEFAULT command.

  Setting the Guardian default file security or default subvolume with the GUARDIAN SECURITY or GUARDIAN VOLUME attributes is equivalent to using the Guardian DEFAULT command. Similarly, the DEFAULT command can be used to change these attributes when Safeguard is running.

- PASSWORD-EXPIRES takes precedence over PASSWORD-MUST-CHANGE.

  If the PASSWORD-EXPIRES and PASSWORD-MUST-CHANGE attributes are set in the same command, the setting of the PASSWORD-EXPIRES attribute takes precedence over the PASSWORD-EXPIRES date calculated as a result of setting the PASSWORD-MUST-CHANGE attribute.

Examples

The group manager for group 14 enters the following SET and ADD commands to add a temporary alias for user 14,22. The user cannot log on with this alias after noon on September 21, 1993.

First the group manager issues the SET ALIAS commands:

```
=SET ALIAS PASSWORD PeaNut
=SET ALIAS USER-EXPIRES 21 SEP 2005, 12:00
=SET ALIAS AUDIT-AUTHENTICATE-PASS all
=SET ALIAS GUARDIAN VOLUME $data2.arthur
=SET ALIAS TEXT-DESCRIPTION "Alias for user 14,22"
```

Then the group manager issues the SHOW ALIAS command to display the changes that have taken place:

```
=SHOW ALIAS
```

The report displays:

```
 TYPE         OWNER
  ALIAS       14,255

   PASSWORD = PeaNut
   USER-EXPIRES               = 21SEP05, 12:00
   PASSWORD-EXPIRES           =    * NONE *
   PASSWORD-MUST-CHANGE EVERY =    * NONE *
   PASSWORD-EXPIRY-GRACE      =    * NONE *
   GUARDIAN DEFAULT SECURITY  = OOOO
   GUARDIAN DEFAULT VOLUME    = $DATA2.ARTHUR

   AUDIT-AUTHENTICATE-PASS  = ALL         AUDIT-MANAGE-PASS  = NONE
   AUDIT-AUTHENTICATE-FAIL  = NONE        AUDIT-MANAGE-FAIL  = NONE
   AUDIT-USER-ACTION-PASS   = NONE
   AUDIT-USER-ACTION-FAIL   = NONE

   TEXT-DESCRIPTION = "Alias for user 14,22"

   CI-PROG = * NONE *
   CI-LIB  = * NONE *
   CI-NAME = * NONE *
   CI-SWAP = * NONE *
   CI-CPU  = ANY
   CI-PRI  = * NONE *
   CI-PARAM-TEXT =

   INITIAL-PROGTYPE      = PROGRAM
   INITIAL-PROGRAM       =
   INITIAL-DIRECTORY     =

  SUBJECT DEFAULT-PROTECTION SECTION UNDEFINED!

  SUBJECT OWNER-LIST SECTION UNDEFINED!
```

Next the group manager issues the ADD ALIAS command to add the alias Temp3 for user 14,22:

```
=ADD ALIAS Temp3, 14,22
```

Finally the group manager issues the INFO ALIAS command to confirm that the alias was added:

```
=INFO ALIAS Temp3
```

The report displays:

```
NAME                          USER-ID  OWNER     STATUS
Temp3                          14,22   14,255    THAWED
```

SHOW ALIAS Command

SHOW ALIAS displays the current default values for user attributes.

When you add an alias to the system, the current default values are used for any attribute you do not specify in the ADD ALIAS command.

To set the default values for user attributes to specific values, use SET ALIAS.

```
SHOW [ / OUT listfile / ] ALIAS
```

`OUT listfile`

directs SAFECOM output to *listfile* for the SHOW report.

For *listfile*, specify any file name. SAFECOM opens *listfile* and appends the output text to the file. If *listfile* does not exist, SAFECOM creates an EDIT-format file and writes the SHOW report to it.

`ALIAS`

identifies ALIAS as the object class of the SHOW command. Omit it if ALIAS is the assumed type. (For more information, see the ASSUME Command on page 4-3.)

SHOW ALIAS Report Format

The SHOW ALIAS command displays the default user attributes and their current values in the format shown in Figure 6-3 on page 6-53.

Figure 6-3. SHOW ALIAS Report Format

```
 TYPE         OWNER
  ALIAS       gn,un

   PASSWORD = [password]
   USER-EXPIRES                =  { date,time | * NONE * }
   PASSWORD-EXPIRES            =    * NONE *
   PASSWORD-MUST-CHANGE EVERY =  { n DAYS | * NONE * }
   PASSWORD-EXPIRY-GRACE       =  { n DAYS | * NONE * }
   GUARDIAN DEFAULT SECURITY  = string
   GUARDIAN DEFAULT VOLUME     = $vol.subvol

   AUDIT-AUTHENTICATE-PASS = a-spec      AUDIT-MANAGE-PASS = a-spec
   AUDIT-AUTHENTICATE-FAIL = a-spec      AUDIT-MANAGE-FAIL = a-spec
   AUDIT-USER-ACTION-PASS  = a-spec
   AUDIT-USER-ACTION-FAIL  = a-spec

   TEXT-DESCRIPTION = ["text"]

   CI-PROG = { prog-filename | * NONE * }
   CI-LIB  = { lib-filename | * NONE * }
   CI-NAME = { process-name | * NONE * }
   CI-SWAP = { $vol[.subvol.filename] | * NONE * }
   CI-CPU  = { num | ANY }
   CI-PRI  = { num | * NONE * }
   CI-PARAM-TEXT = [ param-text ]

   INITIAL-PROGTYPE       = prog-type
   INITIAL-PROGRAM        = [prog-path]
   INITIAL-DIRECTORY      = [dir-path]

  SUBJECT DEFAULT-PROTECTION SECTION

  SUBJECT OWNER-LIST SECTION
```

These user attribute values are:

`OWNER` *`gn,un`*

is the user ID (group number and member number) of the user who will own the alias authentication record.

`PASSWORD = [`*`password`*`]`

identifies the currently assigned logon password. If no password has been assigned, *`password`* does not appear.

`USER-EXPIRES = {` *`date,time`* `| * NONE * }`

either gives the date and time when the alias will expire or indicates that no expiration date has been specified.

```
PASSWORD-EXPIRES = { date,time | * NONE * }
```

either gives the date and time when the password expires or indicates that no expiration date has been specified.

```
PASSWORD-MUST-CHANGE EVERY = { n DAYS | * NONE * }
```

either gives the maximum number of days that the alias can retain the same password or indicates that no limit has been set.

```
PASSWORD-EXPIRY-GRACE = { n DAYS | * NONE * }
```

either gives the number of days after password expiration that the password for this alias can be changed during logon or indicates that no extension period is allowed.

```
GUARDIAN DEFAULT SECURITY = string
```

gives the Guardian default disk file security string.

```
GUARDIAN DEFAULT VOLUME = $vol.subvol
```

gives the Guardian default subvolume.

```
AUDIT-AUTHENTICATE-PASS = a-spec    AUDIT-MANAGE-PASS = a-spec
AUDIT-AUTHENTICATE-FAIL = a-spec    AUDIT-MANAGE-FAIL = a-spec
AUDIT-USER-ACTION-PASS  = a-spec
AUDIT-USER-ACTION-FAIL  = a-spec
```

indicate the conditions under which attempts to authenticate the alias, attempts to manage the alias authentication record, and attempts to perform an operation using the alias are audited by the Safeguard software. For more information about these fields, see the SET ALIAS Command on page 6-38.

```
TEXT-DESCRIPTION = [ "text" ]
```

is the descriptive text associated with the alias record.

```
CI-PROG = { prog-filename | * NONE * }
```

either gives the object file name of the command interpreter started after the user logs on at a Safeguard terminal with this alias or indicates no command interpreter.

```
CI-LIB = { lib | * NONE * }
```

either gives the file name of the library file used with the command interpreter or indicates no library file.

```
CI-NAME = { process-name | * NONE * }
```

either gives the process name assigned to the command interpreter or indicates no process name.

`CI-SWAP = { $`*`vol`*`[`*`.subvol.filename`*`] | * NONE * }`

either gives the swap volume or file used with the command interpreter or indicates no swap volume or file is specified.

`CI-CPU = {` *`num`* `| ANY }`

either gives the number of the CPU in which the command interpreter runs or indicates any CPU will be used.

`CI-PRI = {` *`num`* `| * NONE * }`

either gives the priority at which the command interpreter runs or indicates that no priority is assigned in the user record.

`CI-PARAM-TEXT = [` *`text`* `]`

either gives the startup parameter text supplied to the command interpreter or indicates no parameter is supplied.

`INITIAL-PROGTYPE =` *`prog-type`*

gives the initial program type: PROGRAM, WINDOW, or SERVICE.

`INITIAL-PROGRAM = [` *`prog-path`* `]`

either gives the initial program pathname or appears blank to indicate that no pathname is defined.

`INITIAL-DIRECTORY = [` *`dir-path`* `]`

either gives the initial directory pathname or appears blank to indicate that no pathname is defined.

`SUBJECT DEFAULT-PROTECTION SECTION`

either gives the default protection to be assigned to disk files created by the alias or indicates that no default protection is defined.

`SUBJECT OWNER-LIST SECTION`

lists the secondary owners of the alias authentication record.

Examples

1. This sample SHOW ALIAS command displays the predefined user-attribute settings for the user who has user ID 86,2:

   ```
   =SHOW ALIAS
   ```

   The report displays:

```
 TYPE        OWNER
  ALIAS      86,2

   PASSWORD =
   USER-EXPIRES               =    * NONE *
   PASSWORD-EXPIRES           =    * NONE *
   PASSWORD-MUST-CHANGE EVERY =    * NONE *
   PASSWORD-EXPIRY-GRACE      =    * NONE *
   GUARDIAN DEFAULT SECURITY  = OOOO
   GUARDIAN DEFAULT VOLUME    = $SYSTEM.NOSUBVOL

   AUDIT-AUTHENTICATE-PASS  = NONE        AUDIT-MANAGE-PASS  = NONE
   AUDIT-AUTHENTICATE-FAIL  = NONE        AUDIT-MANAGE-FAIL  = NONE
   AUDIT-USER-ACTION-PASS   = NONE
   AUDIT-USER-ACTION-FAIL   = NONE

   TEXT-DESCRIPTION =

   CI-PROG = * NONE *
   CI-LIB  = * NONE *
   CI-NAME = * NONE *
   CI-SWAP = * NONE *
   CI-CPU  = ANY
   CI-PRI  = * NONE *
   CI-PARAM-TEXT =

   INITIAL-PROGTYPE      = PROGRAM
   INITIAL-PROGRAM       =
   INITIAL-DIRECTORY     =

  SUBJECT DEFAULT-PROTECTION SECTION UNDEFINED!

  SUBJECT OWNER-LIST SECTION UNDEFINED!
```

2. In this example, the PRS group manager (group 86) enters the SHOW ALIAS command after using the SET ALIAS command to establish these user values:

   - A logon password of Pasta
   - A USER-EXPIRES date of December 15, 2005, at midnight
   - A PASSWORD-MUST-CHANGE requirement of 30 days
   - An AUDIT-AUTHENTICATE-PASS specification to audit all successful attempts to authenticate the alias
   - An AUDIT-MANAGE-FAIL specification to audit all unsuccessful attempts to manage the alias authentication record
   - A TEXT-DESCRIPTION of “Alias for user 86,2”
   - A GUARDIAN SECURITY of NUNU and a GUARDIAN VOLUME of $tops.harry

The PRS group manager enters:

```
=SHOW ALIAS
```

The report displays:

```
TYPE        OWNER
 ALIAS      86,255

  PASSWORD = Pasta
  USER-EXPIRES               = 15DEC05, 0:00
  PASSWORD-EXPIRES           =    * NONE *
  PASSWORD-MUST-CHANGE EVERY =    30 DAYS
  PASSWORD-EXPIRY-GRACE      =    * NONE *
  GUARDIAN DEFAULT SECURITY  = NUNU
  GUARDIAN DEFAULT VOLUME    = $TOPS.HARRY

  AUDIT-AUTHENTICATE-PASS  = ALL         AUDIT-MANAGE-PASS  = NONE
  AUDIT-AUTHENTICATE-FAIL  = NONE        AUDIT-MANAGE-FAIL  = ALL
  AUDIT-USER-ACTION-PASS   = NONE
  AUDIT-USER-ACTION-FAIL   = NONE

  TEXT-DESCRIPTION = "Alias for user 86,2"

  CI-PROG = * NONE *
  CI-LIB  = * NONE *
  CI-NAME = * NONE *
  CI-SWAP = * NONE *
  CI-CPU  = ANY
  CI-PRI  = * NONE *
  CI-PARAM-TEXT =

  INITIAL-PROGTYPE      = PROGRAM
  INITIAL-PROGRAM       =
  INITIAL-DIRECTORY     =

 SUBJECT DEFAULT-PROTECTION SECTION UNDEFINED!

 SUBJECT OWNER-LIST SECTION UNDEFINED!
```

THAW ALIAS Command

THAW ALIAS restores a user’s ability to log on to a system with a frozen alias. (After a FREEZE ALIAS command, the user cannot log on to the system using that alias until a THAW ALIAS command is executed for the alias.)

THAW ALIAS has no effect on an alias whose access is not currently frozen.

The primary owner of an alias authentication record, the secondary owners, the primary owner’s group manager, and the super ID can thaw an alias.

```
THAW ALIAS { alias | ( alias [ , alias ] ... ) }

   [ [,] WHERE expression ]
```

ALIAS

identifies ALIAS as the object type of the THAW command. Omit it if ALIAS is the assumed object type. (For more information, see the ASSUME Command on page 4-3.)

`alias`

specifies the alias (or aliases) whose ability to log on is to be restored. *`alias`* can contain wild-card characters.

`WHERE` *`expression`*

causes the THAW command to apply to only authentication records for aliases who belong to the groups specified by *`expression`*. For a description of WHERE *`expression,`* see the ALTER ALIAS Command on page 6-9.

Examples

1. The following command restores the ability to log on with the aliases Temp3 and Temp6:

   ```
   =THAW ALIAS (Temp3, Temp6)
   ```

2. The following command restores the logon ability for all aliases that begin with the letters *`Ca`* and are also members of the group admin4:

   ```
   =THAW USER Ca* WHERE GROUP=admin4
   ```

7 Group Commands

The GROUP commands allow a security administrator to define user groups and manage the membership of those groups. User groups created explicitly with the ADD GROUP command can exist independently of user definitions. The groups created in this manner usually serve as file-sharing groups rather than as administrative groups. Typically, an administrative group is created implicitly with the ADD USER command, as described in Section 5, User Security Commands.

Note. In prior product versions, GROUP commands were used to manage Safeguard security groups. GROUP commands are now used to manage file-sharing groups, as described in this section. Security groups are now managed with the SECURITY-GROUP commands, as described in Section 13, Security Group Commands.

The attributes in a group definition record allow you to specify the group's name and numeric ID, a text description, and a list of group members. Group names and IDs can be mentioned in the Access clause of ACLs defined in protection records. Unlike the Safeguard security groups described in Section 13, Security Group Commands, the groups defined with GROUP commands have no inherent privileges or restrictions associated with them.

Use the MEMBER attribute in a group definition record to specify the users who are members of the group. You can make a single user or alias a member of up to 32 groups. A single group can contain more than 256 members for file sharing.

Who Can Manage User Groups

If no ACL has been defined for OBJECTTYPE USER, use of the ADD GROUP command is restricted to super-group members. If an ACL exists for OBJECTTYPE USER, only users with create (C) authority on that access control list can use the ADD GROUP command.

By default, the OWNER attribute in a group definition record contains the user ID of the user who first created the group and who therefore owns that group definition record. This record owner can change the attributes in the group record. In addition, the original owner can set the OWNER attribute to the user ID of any other user. That other user then has control of the group record. At any time, the new owner (or the owner's group manager or the super ID) can transfer ownership to yet another user.

The owner of a group created implicitly with the ADD USER command is the user who executes that command to add the first user to the group. This group owner can subsequently use the GROUP commands to manage the group definition record.

Only the record owner, that owner's group manager, and the super ID user can change a group definition record or delete the record.

Only the record owner, that owner's group manager, and the super ID user can view their group details by executing the INFO GROUP command.

Group Names and Access Control Lists

Currently, only administrative group names and numbers are allowed on Safeguard ACLs. File-sharing group names and numbers are not permitted in ACLs.

However, the Safeguard software's method of evaluating ACLs recognizes extended group membership. An ACL entry in the form *`group-name.*`* is now interpreted to include all members of the specified group, not just users who have the specified group as their administrative group. Similarly, an entry in the form *`group-number,*`* is interpreted to include all members of the specified group number.

For example, assume that the ALTER GROUP command has been used to add the user with the user name GROUPB.JOE to the group named GROUPA. An ACL that specifies GROUPA.* now includes the user GROUPB.JOE as well as all users whose GROUPA is their administrative group.

Guardian file security also recognizes group membership. The Guardian file-security settings G and C encompass all users whose group list includes the file owner's administrative group.

The Super Group and File-Sharing Membership

Although GROUP commands can be used to add file-sharing members to the super group (group number 255), it is generally not advisable to do so.

Making a user a file-sharing member of the super group does not allow that user to assume all privileges of super-group membership. A file-sharing member is granted a super-group privilege only when granting that privilege is based on the evaluation of a Guardian security string, a Safeguard access control list, or an OSS file permission code. For example, if a Safeguard OBJECTTYPE USER record exists with an entry that grants all super group members (SUPER.*) the authority to execute the ADD USER command, file-sharing members of the super group are also granted this authority.

However, many super-group privileges are based on a check of the specific user ID of the user attempting to execute the privileged command. File-sharing members of the super group are not allowed to perform these types of operations.

Group Command Summary

Table 7-1 on page 7-3 lists the group commands and gives a brief description of each. The remainder of this section describes these commands in detail.

Table 7-1. Group Command Summary

Command	Description
ADD GROUP	Adds a group definition record with the specified group attribute values.
ALTER GROUP	Changes one or more attribute values in a group definition record.
DELETE GROUP	Deletes a group definition record.
GROUP	Displays the existing attribute values in a group definition record.

Syntax of Group Commands

The remainder of this section describes each group command in detail. Commands are presented in alphabetical order, and descriptions contain these elements:

- A summary of the command's function, including the restrictions on who can use the command
- The command syntax, including descriptions of the command parameters and variables
- The format for any command listing or report
- Considerations for the use of the command
- Examples of command usage

ADD GROUP Command

The ADD GROUP command adds a group definition record for a specified group. You can specify only one group name in an ADD GROUP command, and you cannot use wild-card characters in the group name.

If no access control list exists for OBJECTTYPE USER, only local super-group members can use the ADD GROUP command. If an access control list exists for OBJECTTYPE USER, only users with create (C) authority on that access control list can use the ADD GROUP command.

```
ADD GROUP [ NAME ] group-name [ , ] NUMBER group-num ,

   [ group-attribute [ , group-attribute ] ] ...
```

`[ NAME ]` *`group-name`*

specifies the name of the group to be added to the system. The name must be unique within the local system. *`group-name`* is a case-sensitive text string that can be up to 32 alphanumeric characters in length. In addition to alphabetic and numeric characters, the characters period (.), hyphen (-), and underscore (_) are permitted within the text string. The first character of a group name must be alphabetic or numeric. The *group-name* must not already exist as an administrative group name.

Note. If you want to define a group that can be subsequently used as an administrative group, the group name and group number must meet the syntactical requirements for administrative groups:

- The group name must be from one to eight alphabetic or numeric characters, the first of which must be alphabetic. Because the ADD GROUP command recognizes the case of alphabetic characters, the alphabetic characters in an administrative group name must be entered in uppercase.
- The group number for an administrative group must be from 0 through 255.

`NUMBER` *`group-num`*

specifies the numeric ID of the group to be added to the system. The *group-num* is a number from 0 through 65535. It must be unique within the local system. *group-num* must not already exist as an administrative group number.

Note. Safeguard also supports group numbers 65536 through 65567. However, these group numbers are reserved for HP internal use only.

Group number 65536 is reserved for a file sharing group called the SECURITY-ENCRYPTION-ADMIN group. The SECURITY-ENCRYPTION-ADMIN group members are authorized to perform volume level data encryption.

The following considerations govern this group:

- Only super group members can be a part of this group.
- This group is not supported on Safeguard ACLs.

Note. The group numbers are supported only by systems running J06.08 and later J-series RVUs and H06.18 and later H-series RVUs.

`group-attribute`

defines a group attribute value for the group being added. (Default values are used for any attributes not specified in the ADD GROUP command.)

The group attributes are:

```
OWNER [ owner-id ]
MEMBER member-list
DESCRIPTION [ text ]
```

`OWNER [`*`owner-id`*`]`

specifies the owner of a group record. For *owner-id*, specify either of:

```
[\*.]group-name.member-name
[\*.]group-num , member-num
```

If you omit *owner-id*, your user ID becomes the current *owner-id*.

MEMBER *member-list*

specifies users who are granted membership in this group. *member-list* can be either of:

net-user-spec

(*net-user-spec* [, *net-user-spec* ...])

net-user-spec can be either of:

alias
group-name.*member-name*

Note. *net-user-spec* can include wild-card characters (? or *) only on systems running J06.08 and later J-series RVUs and H06.19 and later H-series RVUs.

The *group number,member number* form of a user ID is not allowed in a MEMBER list.

You can specify up to 32 members to be added to the list in a single ADD GROUP command.

DESCRIPTION [*text*]

specifies up to 255 characters of descriptive text. All text following the keyword DESCRIPTION to the end of the command is considered to be descriptive text. Therefore, if you specify a description, it must appear last in the command string.

If you omit *text*, no descriptive text is included in the group record.

OWNER-LIST [[-]*user-list*]

changes the secondary ownership of a group record by adding or deleting owners in the owner list. A minus sign (-) preceding user-list indicates that the specified users are to be deleted from the existing owner list.

If the minus sign is omitted, the specified users are added to the owner list. If user-list is omitted, the owner list is set to null (no secondary owners). A maximum of 50 users can be specified in user-list. For user-list, specify either:

```
net-user-spec
(net-user-spec [;net-user-spec ...])
```

`net-user-spec` is either:

```
[\node-spec.]group-name.member-name
[\node-spec.]group-num , member-num
```

node-spec is one of:

```
*
node-name
node-number
```

```
node-name
specifies the system name.
```

```
node-number
specifies the Expand node number.
```

```
group-name
specifies the name of any group.
```

```
group-num
specifies the group number of any group.
```

Considerations

- There is no restriction on creating groups named SECURITY-ADMINISTRATOR and SYSTEM-OPERATOR. However, such groups have no effect on the execution of restricted commands protected by the Safeguard security groups, as described in Section 13, Security Group Commands.
- If you create a group that qualifies syntactically as an administrative group, ownership of that group does not qualify you to use the ADD USER command to add members to the group. The ADD USER command does not check for group

ownership. It is subject to different restrictions, as described in Section 5, User Security Commands.

- Although it is syntactically valid to create a group with a name that consists of all numbers, HP does not recommend this practice. A numeric name might cause confusion between the group name and group number even though the Safeguard software can distinguish between them.
- HP recommends that group numbers from 0 through 255 be reserved for administrative groups.
- Group ownership does not imply administrative control over the group members. To alter a user or alias authentication record, you must own that record or be the owner's group manager.
- You need not own user or alias authentication records to add them to a group's member list.
- A single user or alias can be a member of up to 32 groups.
- There is no restriction on the number of members in a group. An administrative group is limited to 256 members for administration purposes, but it can have additional members for file sharing.

Examples

1. The following command adds a group definition record for the group named shift1-admin. The group is assigned group number 656. The command includes a description that identifies this group as first-shift administrators. No members are added to the group.

```
=ADD GROUP shift1-admin NUMBER 656, DESCRIPTION All &
=first-shift system administrators
```

2. The following command adds a group definition record for the group named ADMIN, which is assigned the group number 120. No descriptive text is included. This group can later be activated as an administrative group because its name and number meet the syntactical requirements for administrative groups.

```
=ADD GROUP NAME ADMIN, NUMBER 120
```

3. The following command adds a group definition record for the group named Target, which is assigned the group number 320. Three users are added to the group: the user names DEV.PAT and DEV.JAN, and the alias ArthurD. No descriptive text is included.

```
=ADD GROUP Target, NUMBER 320, MEMBER (dev.pat,&
=dev.jan, ArthurD)
```

4. The following command adds a group definition record for the group named MYGROUP1, which is assigned the group number 101. Wild-card character, *, is used to add all users and aliases to the group. No descriptive text is included.

```
 =ADD GROUP MYGROUP1, NUMBER 101, MEMBER *
```

5. The following command adds a group definition record for the group named MYGROUP2, which is assigned the group number 102. Wild -card character, *.*, is used to add all users and aliases in the Guardian user name format to the group. No descriptive text is included.

```
=ADD GROUP MYGROUP2, NUMBER 102, MEMBER *.*
```

6. The following command adds a group definition record for the group named MYGROUP3, which is assigned the group number 103. Wild-card character, ?, is used to add users and aliases whose names are three characters long, begin with the letter t, and end with the letter r, to the group. No descriptive text is included.

```
=ADD GROUP MYGROUP3, NUMBER 103, MEMBER t?r
```

7. The following command adds a group with OWNER-LIST which contains only one local user in the OWNER-LIST. The group does not exist.

```
=ADD GROUP TEST, NUM 100, OWNER-LIST 81,1
```

8. The following command adds a group with OWNER and OWNER-LIST attribute. The group does not exist.

```
=ADD GROUP TEST, NUM 100, OWNER 40,5,OWNER-LIST 81,1;81,2
```

9. The following command adds a group with OWNER-LIST which contains both local users and remote users and also includes node-num in the user specification. The group does not exist.

```
=ADD GROUP TEST, NUM 100, OWNER-LIST 81,1;\*.81,1;\ABC.81,2
```

Note. The OWNER-LIST attribute is only supported by systems running J06.14 and later J-series RVUs and H06.25 and later H-series RVUs.

ALTER GROUP Command

The ALTER GROUP command changes one or more attributes in an existing group definition record.

The owner of a group definition record, the owner's group manager, and the super ID can alter that group record.

```
ALTER GROUP { [ NAME ] name-list | NUMBER num-list } [ , ]

   [ group-attribute [ , group-attribute ] ] ...
```

`[ NAME ]` *`name-list`*

specifies the name of the group or groups to be altered.

`name-list`

specifies one or more groups for which definition records are to be altered. The *`name-list`* can contain up to 32 entries.

name-list can be either of:

group-name

(*group-name* [, *group-name*] ...)

group-name

can be any group name. The name can contain wild-card characters.

NUMBER *num-list*

specifies the numeric ID of the group or groups to be altered. The *num-list* can contain up to 32 entries.

num-list

specifies one or more groups for which definition records are to be altered.

num-list can be either of:

group-num

(*group-num* [, *group-num*] ...)

group-num

can be any group number. Wild-card characters are not valid in a *group-num*.

group-attribute

changes the current value of the specified attribute.

The group attributes are:

OWNER [*owner-id*]
MEMBER [+ | -] *member-list*
DESCRIPTION [*text*]

OWNER [*owner-id*]

transfers ownership of the group definition record to the user whose user ID is specified as *owner-id*. For *owner-id*, specify either of:

[*.]*group-name*.*member-name*
[*.]*group-num* , *member-num*

If you omit *owner-id*, your user ID becomes the current *owner-id*.

MEMBER [+ | -] *member-list*

specifies users to be added to or removed from the group. *member-list* preceded by no sign or by a plus sign (+) indicates that the list is to be added to the group. *member-list* preceded by a minus sign (-) indicates that the list is to be removed from the group. *member-list* can either of these forms:

```
  net-user-spec

( net-user-spec [ , net-user-spec ...] )
```

net-user-spec can be either of:

```
alias
group-name.member-name
```

Note. *net-user-spec* can include wild-card characters (? and *).This feature is supported only on systems running J06.08 and later J-series RVUs and H06.19 and later H-series RVUs.

`The group number.member number form of a user ID is not allowed in a MEMBER list.`

You can specify up to 32 members to be added to the list and up to 32 members to deleted from the list in a single command.

If necessary, you can use more than one MEMBER clause in a single ALTER GROUP command.

To add and delete members in the same command, use separate MEMBER clauses, as Example 3 shows.

`DESCRIPTION [ text ]`

specifies up to 255 characters of descriptive text to replace the existing description for this group. All text following the keyword DESCRIPTION to the end of the command is considered descriptive text.

If you specify DESCRIPTION without any text, the description for this group is removed.

`OWNER-LIST [[-]user-list]`

changes the secondary ownership of a group record by adding or deleting owners in the owner list. A minus sign (-) preceding user-list indicates that the specified users are to be deleted from the existing owner list.

If the minus sign is omitted, the specified users are added to the owner list. If user-list is omitted, the owner list is set to null (no secondary owners). A maximum of 50 users can be specified in user-list. For user-list, specify either:

```
net-user-spec
(net-user-spec [;net-user-spec ...])
```

`net-user-spec` is either:

```
[\node-spec.]group-name.member-name
[\node-spec.]group-num , member-num
```

node-spec is one of:

```
*
node-name
node-number
```

```
node-name
specifies the system name.
```

```
node-number
specifies the Expand node number.
```

```
group-name
specifies the name of any group.
```

```
group-num
specifies the group number of any group.
```

Considerations

- You need not own user or alias authentication records to add those users or aliases to a group's member list.
- A single user or alias can be a member of up to 32 groups.
- There is no restriction on the number of members in a group. An administrative group is limited to 256 members for administration purposes, but it can have additional members for file-sharing purposes.

Examples

1. The following command changes the description of the group assigned group number 656:

```
=ALTER GROUP NUMBER 656 DESCRIPTION All first- and &
=second-shift system administrators
```

2. The following command changes the description of the group Rel20 and all groups whose names begin with *Test*:

```
=ALTER GROUP NAME (Test*, Rel20) DESCRIPTION Temporary &
=group for system test purposes
```

3. The following command adds the user PROG4.SUE to group number 120 and deletes the aliases Joe3 and HotDog from that group:

```
=ALTER GROUP NUMBER 120 MEMBER PROG4.SUE, &
=MEMBER -(Joe3, HotDog)
```

4. The following command alters a group with OWNER-LIST which contains only one local user in the OWNER-LIST:

```
=ALTER GROUP TEST, NUM 100, OWNER-LIST 81, 1
```

5. The following command alters a group with OWNER and OWNER-LIST attribute:

```
=ALTER GROUP TEST, NUM 100, OWNER 40, 5, OWNER-LIST 81, 1;
                                                  81, 2
```

6. The following command alters a group with OWNER-LIST, which contains both local users and remote users and also includes node-num in the user specification:

```
=ALTER GROUP TEST, NUM 100, OWNER-LIST 81,1;\*.81,1;\ABC.81,2
```

7. The following command alters a group so that a user is deleted from the OWNER-LIST:

```
=ALTER GROUP TEST, NUM 100, OWNER-LIST - 81, 1
```

DELETE GROUP Command

The DELETE GROUP command deletes group definition records for specified groups. You can specify groups by either group name or group number in a DELETE GROUP command, but you cannot mix names and numbers within the same command. If you specify group names, they can contain wild-card characters.

A group cannot be deleted if it contains members. You must remove all members from a group before you can delete that group record.

The owner of a group definition record, the owner’s group manager, and the super ID can delete that group record.

```
DELETE GROUP { [ NAME ] name-list | NUMBER num-list }
```

`[ NAME ] name-list`

specifies the name of the group or groups to be altered.

`name-list`

specifies one or more groups for which definition records are to be deleted. The *`name-list`* can contain up to 32 entries.

`name-list` can be either of:

```
  group-name

( group-name [ , group-name ] ... )
```

`group-name`

can be any group name. The name can contain wild-card characters.

`NUMBER num-list`

specifies the numeric ID of the group or groups to be altered. The *`num-list`* can contain up to 32 entries.

`num-list`

specifies one or more groups for which definition records are to be altered.

`num-list` can be either of:

```
  group-num

( group-num[ , group-num] ... )
```

`group-num`

can be any group number. Wild-card characters are not valid in a *`group-num`*.

Considerations

- Unlike an administrative group that is created implicitly with the ADD USER command, a group created with the ADD GROUP command is not deleted automatically when its last member is deleted. (For additional details, see the INFO GROUP Detailed Report on page 7-17.)

Examples

The following command deletes the group definition record for group Firefly2:

```
=DELETE GROUP Firefly2
```

INFO GROUP Command

The INFO GROUP command shows the group attributes stored in a specified group definition record.

Only the record owner, that owner’s group manager, and the super ID user can view their group details by executing the INFO GROUP command.

```
INFO GROUP { [ NAME ] name-list | NUMBER num-list }

   [ [ , ] DETAIL ][, OWNER-LIST]
```

[NAME] *name-list*

specifies the name of the group or groups for which information is to be displayed.

name-list

specifies one or more groups for which definition records are to be displayed. The *name-list* can contain up to 32 entries.

name-list can be either of:

group-name

(*group-name* [, *group-name*] ...)

group-name

can be any group name. The name can contain wild-card characters.

NUMBER *num-list*

specifies the numeric ID of the group or groups for which definition records are to be displayed. The *num-list* can contain up to 32 entries.

num-list

specifies one or more groups for which definition records are to be altered.

num-list can be either of:

group-num

(*group-num* [, *group-num*] ...)

group-num

can be any group number. Wild-card characters are not valid in a *group-num*.

`OWNER-LIST [[-]user-list]`

changes the secondary ownership of a group record by adding or deleting owners in the owner list. A minus sign (-) preceding user-list indicates that the specified users are to be deleted from the existing owner list.

If the minus sign is omitted, the specified users are added to the owner list. If user-list is omitted, the owner list is set to null (no secondary owners). A maximum of 50 users can be specified in user-list. For user-list, specify either:

```
net-user-spec

(net-user-spec [;net-user-spec ...])

net-user-spec is either:


[\node-spec.]group-name.member-name

[\node-spec.]group-num , member-num


node-spec is one of:

*

node-name

node-number


node-name

specifies the system name.


node-number

specifies the Expand node number.


group-name

specifies the name of any group.


group-num

specifies the group number of any group.
```

INFO GROUP Brief Report

Figure 7-1 shows the format of the brief INFO GROUP report. A description of the group attribute values and status fields immediately follows it.

Figure 7-1. INFO GROUP Brief Report Format

```
GROUP NAME                      NUMBER     OWNER    LAST-MODIFIED
group-name                      groupnum   o-id     date,time
```

Figure 7-1 contains the following group attribute values and status fields:

`GROUP NAME`
`group-name`

is the name of the group whose attributes are being displayed.

`NUMBER`
`group-num`

is the group number of the group.

`OWNER`
`o-id`

is the user ID of the user who owns this group definition record.

`LAST-MODIFIED`
`date,time`

is the time and date when this group definition record was last changed (in local civil time).

INFO GROUP Detailed Report

Figure 7-2 on page 7-17 shows the format of the detailed INFO GROUP report. A description of the group attribute values and status fields immediately follows it.

Figure 7-2. INFO GROUP Detailed Report Format

```
GROUP NAME                         NUMBER     OWNER     LAST-MODIFIED
group-name                         groupnum   o-id      date,time

CREATION-TIME      = date,time
CREATOR-USER-NAME  = user-name/alias-name
CREATOR-USER-TYPE  = type (UID)
CREATOR-NODENUMBER = num
AUTO-DELETE = {ON | OFF}
DESCRIPTION = [text]
MEMBER = [member]
OWNER-LIST = [[-]user-list]
```

In addition to the attributes and status fields displayed in the brief INFO GROUP report, the detailed INFO GROUP report also displays these attributes:

`CREATION-TIME  = date, time`

specifies the date and time when the user was created.

`CREATOR-USER-NAME = user-name/alias-name`

specifies the username of the user who created the user.

`CREATOR-USER-TYPE  = USER/ALIAS ( uid )`

identifies if the creator is an alias or a user, followed by the user ID of the creator.

`CREATOR-NODENUMBER = num`

specifies the system number where the user is created.

`AUTO-DELETE = { ON/OFF }`

is the AUTO-DELETE group attribute, which is a read-only attribute. It cannot be set through SAFECOM. The Safeguard software automatically sets this attribute. Groups added with the ADD GROUP command have AUTO-DELETE set to OFF. Groups added implicitly with the ADD USER command have this AUTO-DELETE set to ON.

If AUTO-DELETE is set to ON, the group is deleted automatically when the last member of that group is removed from the group.

`DESCRIPTION = [` *`text`* `]`

is the description of the group whose attributes are being displayed.

`MEMBER = [` *`member`* `]`

is a list of the group members.

`OWNER-LIST [[-]user-list]`

changes the secondary ownership of a group record by adding or deleting owners in the owner list. A minus sign (-) preceding user-list indicates that the specified users are to be deleted from the existing owner list.

If the minus sign is omitted, the specified users are added to the owner list. If user-list is omitted, the owner list is set to null (no secondary owners). A maximum of 50 users can be specified in user-list. For user-list, specify either:

```
net-user-spec

(net-user-spec [;net-user-spec ...])

net-user-spec is either:


[\node-spec.]group-name.member-name

[\node-spec.]group-num , member-num


node-spec is one of:

*

node-name

node-number


node-name

specifies the system name.


node-number

specifies the Expand node number.


group-name

specifies the name of any group.


group-num
   specifies the group number of any group.
```

Considerations

- If the group name contains wild-card characters, and if the user does not have sufficient privileges to retrieve information about any of the group records that match the specified wild-card expression, Safeguard does not display `Security Violation` error. Instead, it displays the records it has access to.

- If the group name does not contain wild-card characters and if the user does not have sufficient privileges to retrieve information about the specified group, a `Security Violation` error is displayed.

Examples

1. The following command displays the group attributes for the group name Day_Shift:

```
=INFO GROUP Day_Shift, DETAIL
```

```
GROUP NAME                       NUMBER   OWNER   LAST-MODIFIED
Day_Shift                          520    12,125  14MAY94, 13:43

 CREATION-TIME      = 14MAY94, 23:38
 CREATOR-USER-NAME  = SUPER.SUPER
 CREATOR-USER-TYPE  = USER  (255,255)
 CREATOR-NODENUMBER =     86
 AUTO-DELETE = OFF
 DESCRIPTION = All first-shift operators
 MEMBER = HarveyJ
 MEMBER = Shift-Super
 MEMBER = OPS.SUE
 MEMBER = OPS.FRED
  GROUP OWNER-LIST SECTION UNDEFINED!
```

2. The following command displays the secondary owners of the group record in brief format by specifying OWNER-LIST option and users are present in OWNER-LIST:

```
=INFO GROUP TEST, OWNER-LIST

Display shall be as shown below:


GROUP NAME       NUMBER         OWNER LAST-MODIFIED

TEST     5      255,255+          8AUG11, 14:10


GROUP OWNER-LIST SECTION

      081,001

      \*.081,002
```

3. The following command displays the secondary owners of the group record in brief format by specifying OWNER-LIST option when no users are present in OWNER-LIST:

```
=INFO GROUP TEST, OWNER-LIST
```

Display shall be as shown below:

```
GROUP NAME        NUMBER     OWNER          LAST-MODIFIED
TEST                 5      255,255            8AUG11, 14:10

GROUP OWNER-LIST SECTION UNDEFINED!
```

To display detailed report of the group. The secondary owners exist for the group.

```
= INFO GROUP TEST, DETAIL

GROUP NAME       NUMBER     OWNER     LAST-MODIFIED
TEST              5      255,255      8AUG11, 14:10

  CREATION-TIME      =  8AUG11, 14:10
  CREATOR-USER-NAME  = SUPER.SUPER
  CREATOR-USER-TYPE  = USER  (255,255)
  CREATOR-NODENUMBER =    167
  AUTO-DELETE        = ON
  DESCRIPTION =
  MEMBER      = testuse1
  MEMBER      = TEST.MGR
  MEMBER      = TEST.USER1
  MEMBER      = TEST.USER2
  MEMBER      = TEST.USER5

  GROUP OWNER-LIST SECTION
      081,001
      \*.081,002
```

1. The following displays the detailed report of the group.The secondary owners do not exist for the group.

```
  = INFO GROUP TEST, DETAIL

GROUP NAME        NUMBER     OWNER       LAST-MODIFIED

TEST               5        255,255       8AUG11, 14:10

  CREATION-TIME      =  8AUG11, 14:10
  CREATOR-USER-NAME  = SUPER.SUPER
  CREATOR-USER-TYPE  = USER  (255,255)
  CREATOR-NODENUMBER =    167
  AUTO-DELETE        = ON
  DESCRIPTION =
  MEMBER      = testuse1
  MEMBER      = TEST.MGR
  MEMBER      = TEST.USER1
  MEMBER      = TEST.USER2
  MEMBER      = TEST.USER5

GROUP OWNER-LIST SECTION UNDEFINED!
```

8

Disk-File Security Commands

The SAFECOM disk file security commands give disk-file owners access control of protected disk files and the ability to specify when to audit attempts to access and manage the authorization records for these files.

By default, only the disk file's owner, the owner's group manager, or the super ID can add a Safeguard authorization record unless a list of users is specified by the OBJECTTYPE DISKFILE. (For more information, see Section 12, OBJECTTYPE Security Commands.) After a record is added, all attempts to access that file are subject to a Safeguard authorization check and optionally to Safeguard auditing. Access to a disk file includes all the standard Guardian access modes: READ, WRITE, EXECUTE, and PURGE. The Safeguard software also supports two additional access authorities: CREATE and OWNER.

Ownership is transferable at the discretion of the initial owner to any user ID or group. Also, owners can create and modify an access control list (ACL) for the file. The ACL selectively grants or denies access to the file.

This section describes disk-file ownership and disk-file authorization records and then summarizes the DISKFILE and DISKFILE-PATTERN security commands. A detailed description of each command follows the command summary.

Disk-File Ownership

The owner ID associated with the disk file indicates who owns the file. When a disk file is created, the owner ID is set to that of the user ID of the file creator, and the file is protected by the standard Guardian security system.

To place a file under Safeguard control, the file owner (or another properly authorized user) creates a disk-file authorization record through the ADD DISKFILE command. A file can also be placed automatically under Safeguard control through the use of a default protection record. For more information, see Section 5, User Security Commands. In each disk file authorization record, the file owner is identified by the user ID stored as the OWNER attribute. The owner identified in this manner is known as the file's primary owner.

The file's primary owner, the owner's group manager, or the super ID can transfer ownership of the file to another user through the ALTER DISKFILE command. Additional ownership is defined by the OWNER authority code for ACL entries and is an independent extension of the initial owner. Any user with OWNER authority on the ACL can manage the file's authorization record by altering, freezing, thawing, or deleting it. This feature allows multiple groups or individuals to administer the security of a particular file.

Any user with OWNER authority on the ACL can explicitly deny a local super ID any of the authorities (including OWNER) implicitly granted to that user ID and have this denial actively enforced all of the time.

The primary owner can also set the PROGID attribute through the ALTER DISKFILE command. The PROGID attribute is controlled by the super ID, primary owners, and secondary owners, and is not transferable.

When a disk file is under Safeguard protection, the Safeguard software controls all security attributes. The FUP GIVE, LICENSE, REVOKE, and SECURE commands are superseded by Safeguard protection. Also, FUP INFO displays **** in the RWEP column indicating that access to the file is controlled by the Safeguard software. The owner must use SAFECOM DISKFILE commands to manage Safeguard access controls for the file. If a file is placed automatically under the Safeguard control using the DEFAULT-PROTECTION or PERSISTENT PROTECTION record, FUP INFO displays **** in the RWEP column.

For example, a disk-file owner can use the ALTER DISKFILE command to change the defined ACL entries. A file owner can also use the FREEZE DISKFILE command to temporarily suspend access by other users and can later enter a THAW DISKFILE command to restore access.

OWNER authority can be specified for all disk files protected by the Safeguard software. OWNER is automatically included whenever the * authority code is used. It can be abbreviated as O.

With the Safeguard software, the owner of a disk file can also be defined as a network user. A network user who owns a protected file can use the Safeguard software from a remote node to control access to that file (provided the user has remote passwords set up between the two systems).

For more information about controlling the class of objects, see DISKFILE on page 12-2.

You can also use diskfile patterns to secure disk files. For more information, see the *Safeguard User’s Guide*.

Disk-File Access Authorities

The ACL defined for a disk file can grant any combination of these access authorities to users and user groups:

READ	Read the contents of a disk file
WRITE	Modify the contents of a disk file
EXECUTE	Run a program object disk file as a process
PURGE	Purge a disk file
CREATE	Create a disk file
OWNER	Manage the authorization record

The Safeguard software can also control the creation of disk files on specific volumes or subvolumes. For a description of the SAFECOM commands that control file-creation authority, see Section 9, Disk Volume and Subvolume Security Commands.

Disk-File Access Authorization

When a process attempts to access a protected disk file, the Safeguard software checks the processes group list and the disk file ACL to see if the user identified by the process accessor ID (PAID) of that process has the required access authority. If that user lacks the authority, the access attempt is rejected with a security violation error (file error 48). For more information on process and creator accessor IDs, see the *Security Management Guide*.

When the Safeguard software authorizes access to a disk file, it also determines whether the requesting process was started by a user authenticated on a remote system. If so, the user identified by the PAID of that process must be identified as a network user on the disk file ACL, or the Safeguard software rejects the access attempt with a security violation error (file error 48).

Processes use system procedure calls to access disk files. The Safeguard software must authorize any attempts to access protected disk files made through Guardian procedures:

To create a process
: The owner must have EXECUTE authority for the program object disk file. (Creating a process is also subject to authorizations from the PROCESS object type.)

To open a file
: The owner must have either READ or WRITE authority.

To purge a file
: The owner must have PURGE authority.

The Safeguard software must also authorize attempts to rename a protected disk file. Table 8-1 shows the access authorities required to rename a disk file on a system protected by the Safeguard software.

Table 8-1. Access Authority Required to Rename a File

Current File Name			**New File Name**		**Result**
Safeguard Record Exists?	**Safeguard Purge Allowed?**	**Guardian Purge Allowed?**	**Safeguard Vol/Subvol/ Disk File Record Exist?**	**Safeguard Create Allowed?**	**Rename Allowed?**
No	-	Yes	No	-	Yes
No	-	Yes	Yes	Yes	Yes
No	-	Yes	Yes	No	No
No	-	No	-	-	No
Yes	Yes	-	No	-	Yes

Table 8-1. Access Authority Required to Rename a File

Current File Name			New File Name		Result
Yes	Yes	-	Yes	Yes	Yes
Yes	Yes	-	Yes	No	No
Yes	No	-	-	-	No

Note. If a persistent protection record exists for the new file name, the renamed file assumes that persistent ACL. If the current file has a Safeguard ACL and the new file name does not have a persistent protection record, the renamed file assumes the ACL of the current file. However, if the PERSISTENT flag is ON in the current file's protection record, that ACL is not transferred to the renamed file.

An open request that passes the Safeguard authorization check can nevertheless fail. For example, if a process attempts to open a file that is already open with exclusive access, the open attempt fails with file error 12 (file in use). (For more information, see the *Guardian Procedure Calls Reference Manual*.)

Disk-File Security Command Summary

Table 8-2 gives a brief description of the disk-file security commands. The remainder of this section describes these commands in detail.

Table 8-2. Disk-File Security Command Summary (page 1 of 2)

Command	Description
ADD DISKFILE*	Adds a disk-file authorization record with the specified attributes. Current default disk-file attribute values are used for any attributes not specified in the ADD DISKFILE command.
ADD DISKFILE-PATTERN*	Adds a diskfile pattern for files in specified location. Current default diskfile-pattern attribute values are used for any attributes not specified in the ADD DISKFILE command.
ALTER DISKFILE*	Changes one or more attribute values in an authorization record. For all disk-file attributes except ACCESS, ALTER DISKFILE replaces the current attribute value with the specified value. For the ACCESS attribute, ALTER DISKFILE changes the existing ACL to incorporate *access-spec*.
ALTER DISKFILE-PATTERN*	Changes one or more of the security attributes in the diskfile-pattern authorization record.
DELETE DISKFILE*	Deletes a disk-file authorization record. After deletion, all accesses to the file are subject to standard Guardian security checking. The original security is restored for the deleted file.
DELETE DISKFILE-PATTERN*	Removes a diskfile pattern from the Safeguard database by deleting the disk-file authorization record.

Table 8-2. Disk-File Security Command Summary (page 2 of 2)

Command	Description
FREEZE DISKFILE*	Temporarily suspends access to a disk file. (Only the file owner, the owner’s group manager, and the super ID can access a frozen disk file.)
FREEZE DISKFILE-PATTERN*	Suspends access authority to a diskfile pattern. No one except an owner, the primary owner's group manager, and the super ID can gain access to the frozen pattern.
INFO DISKFILE*	Displays current values for the specified disk file.
INFO DISKFILE-PATTERN*	Displays current values for the specified diskfile pattern.
RESET DISKFILE	Resets one or more default attribute values to predefined values.
RESET DISKFILE-PATTERN	Resets one or more default diskfile-pattern attributes to values predefined by the Safeguard software. Any subsequent ADD DISKFILE-PATTERN commands use these predefined defaults for attributes not specified in the ADD DISKFILE-PATTERN command.
SET DISKFILE	Sets one or more default attributes to specified values. When an authorization record is added, the current default disk-file attribute values are used for any attributes not specified in the ADD DISKFILE command.
SET DISKFILE-PATTERN	Establishes default diskfile-pattern attributes that you specify. Any subsequent ADD DISKFILE-PATTERN commands use these defaults for attributes not specified in the ADD DISKFILE command.
SHOW DISKFILE	Displays the current default values of the attributes.
SHOW DISKFILE-PATTERN	Displays the current default attributes for diskfile-patterns. Any subsequent ADD DISKFILE-PATTERN commands use these defaults for attributes not specified in the ADD DISKFILE-PATTERN command.
THAW DISKFILE*	For frozen disk files, restores the access authorities granted to users through the disk file ACL.
THAW DISKFILE-PATTERN*	Restores diskfile-pattern access authorities for users on the access control list.

* The ADD, ALTER, DELETE, FREEZE,THAW, and INFO commands used with wild cards, when there is no existing DISKFILE matching the given pattern, will display the "Record not found" error.

Syntax of Disk-File Security Commands

This section includes the individual syntax descriptions for the SAFECOM disk-file security commands. Commands are in alphabetical order and contain these elements:

- A summary of the function performed by the command, including the restrictions on who can use the command

- The syntax of the command, including descriptions of the command parameters and variables
- The format for the command listing or report (for commands that produce displays or listings)
- Considerations for the use of the command
- Examples of command usage

ADD DISKFILE Command

ADD DISKFILE creates a Safeguard authorization record for one or more existing disk files. After an authorization record is created for a disk file, all attempts to access the disk file are subject to a Safeguard authorization check and optionally to Safeguard auditing.

Only the owner of a disk file, the owner’s group manager, or the local super ID can add an authorization record for a disk file.

You can use SET DISKFILE to establish default disk-file attribute values and then use ADD DISKFILE simply to name the disk files to which the default attributes are to be applied. You can also specify values for the disk-file attributes in your ADD DISKFILE command. The current default values are used for any attributes not specified in your ADD DISKFILE command.

```
ADD DISKFILE filename-list [ , ]

   [ LIKE disk-file-name | disk-file-attribute ]

   [ , disk-file-attribute ] ...
```

`DISKFILE`

specifies DISKFILE as the object type of the ADD command. Omit this option if DISKFILE is the assumed object type. (For more information on assumed object types, see the ASSUME Command on page 4-3.)

`filename-list`

specifies one or more disk files for which authorization records are to be added. (Authorization records can be added only for disk files that already exist.)

`filename-list` can be either of:

```
  disk-file-name

( disk-file-name [ , disk-file-name ] ... )
```

`disk-file-name`

can be any disk-file name. The name can contain wild-card characters.

`LIKE` *`disk-file-name`*

adopts the existing attribute values of *`disk-file-name`* as the *`disk-file-attribute`* values to be used for the authorization record or records being added.

`disk-file-name`

identifies the disk file whose current *`disk-file-attribute`* values are to be assigned to the disk-file authorization record or records being added. *`disk-file-name`* can be any disk-file name.

`disk-file-attribute`

defines a disk-file attribute value for the disk-file authorization record or records being added. The disk-file attributes are:

```
OWNER [owner-id]
ACCESS access-spec [ ; access-spec ] ...
LICENSE {ON|OFF}
PROGID {ON|OFF}
CLEARONPURGE {ON|OFF}
PERSISTENT {ON|OFF}
OBJECT-TEXT-DESCRIPTION "[any-text]"
AUDIT-ACCESS-PASS [audit-spec]
AUDIT-ACCESS-FAIL [audit-spec]
AUDIT-MANAGE-PASS [audit-spec]
AUDIT-MANAGE-FAIL [audit-spec]
WARNING-MODE {ON|OFF}
TRUST {ME|SHARED|OFF} (H-series only)
AUDIT-PRIV-LOGON { ON | OFF}
PRIV-LOGON { ON | OFF}
```

Note. The attributes, AUDIT-PRIV-LOGON and PRIV-LOGON, are supported only on systems running H06.11 and later H-series RVUs and G06.32 and later G-series RVUs.

`OWNER [`*`owner-id`*`]`

specifies the new owner of the disk file or files. *`owner-id`* can be either of:

```
[\node-spec.]group-name.member-name
[\node-spec.]group-num , member-num
```

If you omit *`owner-id`*, *`owner-id`* is set to your user ID.

`ACCESS` *`access-spec`* `[ ;` *`access-spec`* `] ...`

changes the ACL for *`filename-list`* by adding or deleting ACL entries or by changing the authority list of a current ACL entry.

`PROCESS-ACCESS access-spec [ ; access-spec ] ...`

is used to set the default process access list for a process launched from a specific diskfile. It changes the process ACL for filename-list by adding or deleting entries, or changing the authority list of a current entry.

An ACL contains as many as 50 entries that grant or deny access authorities to users and user groups.

access-spec has the form:

```
user-list  [-] [DENY] authority-list
group-list [-] [DENY] authority-list
```

user-list

specifies users who are granted (or denied) the access authorities specified with the following *authority-list*. *user-list* can be either of:

```
  net-user-spec

( net-user-spec [ , net-user-spec ] ... )
```

net-user-spec can be any of:

```
[\node-spec.]adm-group-name.user-name
[\node-spec.]adm-group-num , user-num
[\node-spec.]adm-group-name.*
[\node-spec.]adm-group-num , *
[\node-spec.]*.*
[\node-spec.]*,*
```

-

(minus-sign) operates on existing ACL entries. The minus-sign form of *access-spec* modifies the current default ACL. The *authority* entries are removed from the default ACL entries for the users specified with *user-list*.

group-list

can take either of these forms:

```
  net-group-spec

( net-group-spec [ , net-user-spec ] ... )
```

net-group-spec

can take any of these forms:

```
GROUP [NAME][\node-spec.] group-name

GROUP NUMBER [\node-spec.]
```

node-spec

has the form:

```
* | node-name | node-number
```

node-name

specifies the system name.

node-number

specifies the Expand node number.

adm-group-name

specifies the name of the administrative group.

admin-group-name

specifies the group number of an administrative group.

group-name

specifies the name of any group.

group-num

specifies the group number of any group.

`-`

(minus-sign) operates on existing ACL entries. The minus-sign form of *access-spec* modifies the current default ACL. The *authority* entries are removed from the default ACL entries for the users specified with *user-list*.

Note. Specifying ACCESS *access-spec* through the ADD command does not override the current default ACL (established through the SET command). Instead, any ACL entries specified with the ADD command are added to the current default ACL, and the entire ACL is defined for the disk file whose authorization record is being added.

`DENY`

denies the users or groups specified by *user-list* the access authorities specified by *authority-list*.

authority-list

specifies the access authorities to be granted (or denied) to *user-list*. *authority-list* can be any one of:

```
  authority
( authority [ , authority ] ... )
  *
```

authority

is any one of:

```
R[EAD]
W[RITE]
E[XECUTE]
P[URGE]
C[REATE]
O[WNER]
```

`*`

(asterisk) specifies all the disk-file access authorities except CREATE authority (R, W, E, P, and O).

`LICENSE {ON|OFF}`

either licenses a program object file or revokes the license of a currently licensed program object file. (For more information about the LICENSE attribute, see SET DISKFILE Command on page 8-57.)

`LICENSE ON`

licenses all program object files specified with *filename-list*.

`LICENSE OFF`

revokes the license of all program object files specified with *filename-list*.

`PROGID {ON|OFF}`

changes the PROGID attribute of a program object file. When the PROGID attribute is set ON, the process accessor ID (PAID) of a process that is executed from that object file is set to the user ID of the primary owner of the object file. When PROGID is OFF, the PAID of a process run from the object file is set to the user ID of the user who runs the process.

`PROGID ON`

indicates the PROGID attribute is set ON for all program object files specified with *filename-list*.

`PROGID OFF`

indicates the PROGID attribute is set OFF for all program object files specified with subsequent ADD DISKFILE commands.

`CLEARONPURGE {ON|OFF}`

changes the CLEARONPURGE attribute for all the disk files in *filename-list*. The CLEARONPURGE attribute specifies whether the data pages of a disk file are physically cleared when the file is purged. (For more information about the CLEARONPURGE, see SET DISKFILE Command on page 8-57.)

`CLEARONPURGE ON`

indicates that when a disk file is purged, its entry in the volume directory is deleted, and its data pages are physically cleared.

`CLEARONPURGE OFF`

indicates that when a disk file is purged, its entry in the volume directory is deleted.

`PERSISTENT {ON|OFF}`

changes the PERSISTENT attribute for all the disk files in *`filename-list`*. The PERSISTENT attribute specifies whether the authorization record for a disk file is retained if the disk file is purged.

`PERSISTENT ON`

indicates that the authorization record for the disk file is retained if the file is purged. If you purge a file with PERSISTENT ON and later create a file with the same name, that file assumes the authorization record associated with the old file.

`PERSISTENT OFF`

indicates the authorization record for the disk file is deleted if the file is purged.

`OBJECT-TEXT-DESCRIPTION “[any-text]”`

allows you to store printable characters as comments. These comments are associated with the objects and are used to manage the object authorization record.

The text description field can accommodate 255 bytes of text data.

Note. When the LIKE clause is used with the ADD DISKFILE command, the OBJECT-TEXT-DESCRIPTION field is not copied with other object authorization record attributes.

The OBJECT-TEXT-DESCRIPTION attribute is supported only on systems running J06.05 and later J-series RVUs, H06.16 and later H-series RVUs, and G06.32 and later G-series RVUs.

`AUDIT-ACCESS-PASS [`*`audit-spec`*`]`

changes the *`audit-spec`* for successful attempts to access the disk file. The form of *`audit-spec`* is:

`{ ALL | LOCAL | REMOTE | NONE }`

For a description of *`audit-spec`*, see the SET DISKFILE Command on page 8-57. Omitting *`audit-spec`* specifies NONE.

`AUDIT-ACCESS-FAIL [`*`audit-spec`*`]`

changes the *`audit-spec`* for unsuccessful attempts to access the disk file. The form of *`audit-spec`* is:

```
{ ALL | LOCAL | REMOTE | NONE }
```

For a description of *`audit-spec`*, see the SET DISKFILE Command on page 8-57. Omitting *`audit-spec`* specifies NONE.

`AUDIT-MANAGE-PASS [`*`audit-spec`*`]`

changes the *`audit-spec`* for successful attempts to manage (change or read) a disk-file authorization record. The form of *`audit-spec`* is:

```
{ ALL | LOCAL | REMOTE | NONE }
```

For a description of *`audit-spec`*, see the SET DISKFILE Command on page 8-57. Omitting *`audit-spec`* specifies NONE.

`AUDIT-MANAGE-FAIL [`*`audit-spec`*`]`

changes the *`audit-spec`* for unsuccessful attempts to manage (change or read) a disk-file authorization record. The form of *`audit-spec`* is:

```
{ ALL | LOCAL | REMOTE | NONE }
```

For a description of *`audit-spec`*, see the SET DISKFILE Command on page 8-57. Omitting *`audit-spec`* specifies NONE.

`WARNING-MODE { ON | OFF }`

defines whether warning mode is enabled for the specified disk file. The value is required. For details on warning mode, see the *Safeguard Administrator's Manual*.

ON enables warning mode for the specified disk file. The initial value is OFF, which disables warning mode for the specified disk file.

`TRUST { ME | SHARED | OFF }`

sets the TRUST attribute for the specified disk file. The disk file must be a program object file. The initial value is OFF. This attribute is valid only on systems running H-series RVUs. Only the super ID can set this attribute.

`PRIV-LOGON { ON | OFF}`

establishes whether the program file (object disk file) can request additional logon related sensitive features. When set to ON, a process created from this program file can request a logon without specifying a password.

A process originated from a program file calling USER_AUTHENTICATE_ with a 2 and 15 bit set to ON, the requesting user for authentication need not give a password. Even with wrong password the user will be able to logon successfully as bit 2 and 15 in the options field. In case of only bit 2 set to 1 and bit 15 as 0; no fail delay will take place. That is, no failure delay will be imposed even after three attempts with wrong password. The authentication will not be successful but there will be no delay imposed.

Also establishes whether the program file (object disk file) can request a delay to be imposed for failed logon attempts. When set to ON, a process created from this program file is not subjected to logon failure delays.

OFF is the initial value.

PRIV-LOGON may also be used in the WHERE expression of a command to restrict scope of that command to files with PRIV-LOGON ON.

Considerations

- Attributes in an ADD command affect only the record added.

  Any attribute specifications in an ADD DISKFILE command affect only the authorization record being created and do not change the current default disk-file attribute values. This condition is also true for a LIKE clause in an ADD DISKFILE command.

- Disk-file security can be managed from a remote node.

  An authorization record for a disk file can be added by only the local owner of the file, the owner’s group manager, or the super ID. However, if a disk-file authorization record is added that specifies a network user ID for the OWNER attribute, the authorization record can be altered, frozen, thawed, and deleted by that network user from a remote or local node.

- Relationship between ADD DISKFILE and the FUP GIVE, SECURE, LICENSE, and REVOKE commands

  After you create an authorization record for a disk file, the FUP GIVE, SECURE, LICENSE, and REVOKE commands no longer work for the disk file. You must use the ALTER DISKFILE command to perform the equivalent operations. (For a list of equivalent FUP and SAFECOM commands, see the Considerations for ALTER DISKFILE Command on page 8-21.)

  However, the super ID can use the FUP SECURE, LICENSE, and REVOKE commands on a disk file that has a Safeguard protection record. Even though this usage is allowed, restrict it to emergency situations. It can result in access mediation problems and inconsistencies in Safeguard protection records.

- Using LIKE *disk-file-name*

You can use the LIKE *disk-file-name* clause to define all the disk-file attribute values for a disk file, and then change one or more of the attribute values by specifying new values after the LIKE keyword. For example, this command adds an authorization record for MEMO1 that has the same disk-file attribute values as MEMO2 except for the OWNER attribute:

```
=ADD DISKFILE memo1, LIKE memo2, OWNER sales.kidd
```

Using the LIKE clause with an ADD DISKFILE command does not change any of the current default disk-file attribute values.

- Securing partitioned files

  To secure a partitioned disk file completely, add a separate disk-file authorization record for each partition. Adding an authorization record for only the primary partition protects the partitioned file from any accesses made by opening the primary partition but does not prevent the secondary partitions from being opened individually.

- Renaming a file with persistent protection

  If you rename a file that has persistent protection, the persistent protection is lost because it remains associated with the source file name. However, if the target file name in a rename operation has persistent protection, the new file assumes that protection record.

- CREATE authority is meaningless without persistent protection.

  CREATE authority for a disk file has no meaning unless the PERSISTENT attribute is ON for that file.

- The OWNER attribute and persistent protection

  When a file with persistent protection is created, the OWNER attribute is not changed to match the user ID of the user who creates the file. The OWNER attribute remains set to the owner of the file’s protection record.

- Persistent protection takes precedence over default protection.

  When a file with persistent protection is created, the persistent protection takes precedence over any default protection specified for the user who creates the file.

- The PROGID, LICENSE, and CLEARONPURGE attributes in the persistent protection records.

  When a file with persistent protection is purged, the PROGID, LICENSE, and CLEARONPURGE attributes are set to OFF.

- If disk-file persistence is enabled, the ADD DISKFILE command accepts disk-file protection records for non-existent files.

- Licensed program object file requires local super ID.

Only a local super ID can add an authorization record for a licensed program object file and retain the license attribute in the newly added authorization record.

△ **Caution.** When adding an authorization record for a licensed program object file, set the LICENSE attribute value to ON. If the LICENSE attribute is OFF (the default value), the license for that object file is revoked.

When you add an authorization record for a disk file that currently has the PROGID or CLEARONPURGE options set to ON, set the corresponding disk file attribute to ON before adding the authorization record. (The default attribute value for PROGID and CLEARONPURGE is OFF.)

Examples

- The owner of the disk file $DATA.KEEP.INFO uses these commands to add a Safeguard authorization record for the file, provide its description, and give ownership of the file to a member of group 86:

```
=SET DISKFILE ACCESS 86,2 (r,w,e,p); 86,* (r,e)
=SET DISKFILE CLEARONPURGE ON, AUDIT-ACCESS-PASS all,&
=AUDIT-MANAGE-PASS all
=ADD DISKFILE $DATA.KEEP.INFO,OBJECT-TEXT-DESCRIPTION “ACL &
and Record Created”,OWNER 86,2
```

  The first SET command establishes an ACL that grants all four access privileges (RWEP) to user ID 86,2 and allows every member of the group 86 to read and execute the file. Next, the CLEARONPURGE attribute is set to ON, and the Safeguard software is instructed to audit all successful attempts to access this file or its authorization record. Finally, the ADD command adds a Safeguard record, allows you to add information on the object, and sets the OWNER attribute to user ID 86,2.

- Following example creates a process from a diskfile and grants create permissions to a specific user or group.

```
add diskfile $data.vol.test, process-access  x.y C
```

  A process is created from object *test* such that create permission is granted to the user x.y.

ADD DISKFILE-PATTERN Command

ADD DISKFILE-PATTERN creates a Safeguard authorization record for one or more disk files. After a diskfile-pattern authorization record is created, all attempts to access the disk files described by that pattern are subject to a Safeguard authorization check and optionally to Safeguard auditing.

You can use SET DISKFILE-PATTERN to establish default disk-file attribute values and then use ADD DISKFILE-PATTERN simply to name the disk files to which the default attributes are to be applied. You can also specify values for the disk-file attributes in your ADD DISKFILE-PATTERN command. The current default values are used for any attributes not specified in your ADD DISKFILE-PATTERN command.

```
ADD DISKFILE-PATTERN pattern-spec-list [ , ]
[ LIKE pattern-spec | pattern-attribute ]
[ , pattern-attribute ] ...
```

pattern-spec-list

is the same as the corresponding non-pattern object types. That is, a PATTERN-SPEC-LIST is a comma-separated list of one or more PATTERN-SPEC attributes. (*pattern-spec* [, *pattern-spec*] . . .)

LIKE *pattern-spec*

adopts the existing attribute values of *pattern-spec* as the *pattern-attribute* values to be used for the authorization record or records being added.

pattern-spec

are the characters that define the pattern that describe a set of objects. The PATTERN-SPEC for a diskfile pattern is a fully qualified diskfile name that contains at least one wildcard in either the subvolume or file name component that includes the following components:

- A volume name, which will include only valid volume characters. For the ADD command, wildcards are not valid in the volume name component of the pattern-spec when used for a LIKE operation. One wildcard character is required in either the subvolume or filename.
- A subvolume name, which might include wildcard characters and valid subvolume characters.
- A file name, which might include wildcard characters and valid file name characters.

pattern-attribute

defines a pattern attribute value for the diskfile-pattern authorization record or records being added. The pattern attributes are:

```
OWNER [owner-id]
ACCESS access-spec [ ; access-spec ] ...
AUDIT-ACCESS-PASS [audit-spec]
AUDIT-ACCESS-FAIL [audit-spec]
AUDIT-MANAGE-PASS [audit-spec]
AUDIT-MANAGE-FAIL [audit-spec]
WARNING-MODE {ON|OFF}
```

`OWNER [`*`owner-id`*`]`

specifies the new owner of the diskfile pattern. *`owner-id`* can be either of:

```
[\*.]group-name.member-name
[\*.]group-num , member-num
```

If you omit *`owner-id`*, *`owner-id`* is set to your user ID.

`ACCESS` *`access-spec`* `[ ;` *`access-spec`* `] ...`

changes the ACL for *`filename-list`* by adding or deleting ACL entries or by changing the authority list of a current ACL entry.

An ACL contains as many as 50 entries that grant or deny access authorities to users and user groups.

`access-spec` has the form:

```
user-list  [-] [DENY] authority-list

group-list [-] [DENY] authority-list
```

`user-list`

specifies users who are granted (or denied) the access authorities specified with the following *`authority-list`*. *`user-list`* can be either of:

```
  net-user-spec

( net-user-spec [ , net-user-spec ] ... )
```

`net-user-spec` can be any of:

```
[\node-spec.]adm-group-name.user-name
[\node-spec.]adm-group-num , user-num
[\node-spec.]adm-group-name.*
[\node-spec.]adm-group-num , *
[\node-spec.]*.*
[\node-spec.]*,*
```

`-`

(minus-sign) operates on existing ACL entries. The minus-sign form of *`access-spec`* modifies the current default ACL. The *`authority`* entries are removed from the default ACL entries for the users specified with *`user-list`*.

`group-list`

can take either of these forms:

```
  net-group-spec

( net-group-spec [ , net-user-spec ] ... )
```

net-group-spec

can take any of these forms:

`GROUP [NAME][\`*`node-spec`*`.]` *`group-name`*e

`GROUP NUMBER [\`*`node-spec`*`.]`

node-spec

has the form:

`*` `|` *`node-name`* `|` *`node-number`*

node-name

specifies the system name.

node-number

specifies the Expand node number.

adm-group-name

specifies the name of the administrative group.

admin-group-name

specifies the group number of an administrative group.

group-name

specifies the name of any group.

group-num

specifies the group number of any group.

`-`

(minus-sign) operates on the existing ACL entries. The minus-sign form of *`access-spec`* modifies the current default ACL. The *`authority`* entries are removed from the default ACL entries for the users specified with *`user-list`*.

Note. Specifying ACCESS *`access-spec`* through the ADD command does not override the current default ACL (established through the SET command). Instead, any ACL entries specified with the ADD command are added to the current default ACL, and the entire ACL is defined for the disk file whose authorization record is being added.

`DENY`

denies the users or groups specified by *`user-list`* the access authorities specified by *`authority-list`*.

authority-list

specifies the access authorities to be granted (or denied) to *user-list*. *authority-list* can be any one of:

```
  authority
( authority [ , authority ] ... )
  *
```

authority

is any one of:

```
R[EAD]
W[RITE]
E[XECUTE]
P[URGE]
C[REATE]
O[WNER]
```

*

(asterisk) specifies all the disk-file access authorities (R, W, E, P, C, and O).

AUDIT-ACCESS-PASS [*audit-spec*]

changes the *audit-spec* for successful attempts to access the diskfile pattern. The form of *audit-spec* is:

```
{ ALL | LOCAL | REMOTE | NONE }
```

For a description of *audit-spec*, see the SET DISKFILE Command on page 8-57. Omitting *audit-spec* specifies NONE.

AUDIT-ACCESS-FAIL [*audit-spec*]

changes the *audit-spec* for unsuccessful attempts to access the diskfile pattern. The form of *audit-spec* is:

```
{ ALL | LOCAL | REMOTE | NONE }
```

For a description of *audit-spec*, see the SET DISKFILE Command on page 8-57. Omitting *audit-spec* specifies NONE.

AUDIT-MANAGE-PASS [*audit-spec*]

changes the *audit-spec* for successful attempts to manage (change or read) a diskfile-pattern authorization record. The form of *audit-spec* is:

```
{ ALL | LOCAL | REMOTE | NONE }
```

For a description of *audit-spec*, see the SET DISKFILE Command on page 8-57. Omitting *audit-spec* specifies NONE.

`AUDIT-MANAGE-FAIL [`*`audit-spec`*`]`

changes the *`audit-spec`* for unsuccessful attempts to manage (change or read) a diskfile-pattern authorization record. The form of *`audit-spec`* is:

```
{ ALL | LOCAL | REMOTE | NONE }
```

For a description of *`audit-spec`*, see the SET DISKFILE Command on page 8-57. Omitting *`audit-spec`* specifies NONE.

`WARNING-MODE { ON | OFF }`

defines whether the warning mode is enabled for the specified diskfile pattern. The value is required. For more information on the warning mode, see the *Safeguard Administrator's Manual*.

ON enables warning mode for the specified diskfile pattern. The initial value is OFF, which disables warning mode for the specified diskfile pattern.

Considerations

- Attributes in an ADD command affect only the record added.

  Any attribute specifications in an ADD DISKFILE command affect only the authorization record being created and do not change the current default disk-file attribute values. This condition is also true for a LIKE clause in an ADD DISKFILE command.

- Disk-file security can be managed from a remote node.

  An authorization record for a disk file can be added by only the local owner of the file, the owner’s group manager, or the super ID. However, if a disk-file authorization record is added that specifies a network user ID for the OWNER attribute, the authorization record can be altered, frozen, thawed, and deleted by that network user from a remote or local node.

- Relationship between ADD DISKFILE and the FUP GIVE, SECURE, LICENSE, and REVOKE commands

  After you create an authorization record for a disk file, the FUP GIVE, SECURE, LICENSE, and REVOKE commands no longer work for the disk file. You must use the ALTER DISKFILE command to perform the equivalent operations. (For a list of equivalent FUP and SAFECOM commands, see the Considerations for ALTER DISKFILE Command on page 8-21.)

  However, the super ID can use the FUP SECURE, LICENSE, and REVOKE commands on a disk file that has a Safeguard protection record. Even though this usage is allowed, restrict it to emergency situations. It can result in access mediation problems and inconsistencies in Safeguard protection records.

- Using LIKE *`disk-file-name`*

You can use the LIKE *disk-file-name* clause to define all the disk-file attribute values for a disk file, and then change one or more of the attribute values by specifying new values after the LIKE keyword. For example, this command adds an authorization record for MEMO1 that has the same disk-file attribute values as MEMO2 except for the OWNER attribute:

```
=ADD DISKFILE memo1, LIKE memo2, OWNER sales.kidd
```

Using the LIKE clause with an ADD DISKFILE command does not change any of the current default disk-file attribute values.

- Securing partitioned files

  To secure a partitioned disk file completely, add a separate disk-file authorization record for each partition. Adding an authorization record for only the primary partition protects the partitioned file from any accesses made by opening the primary partition but does not prevent the secondary partitions from being opened individually.

Examples

1. To add a protection record that describes all production data base files that reside on $DATA with subvolume names that begin with PROD:

   ```
   ADD DISKFILE-PATTERN $DATA.PROD*.*, &
   ACCESS PROD.* (R,W)
   ```

2. To add a diskfile pattern for all files in subvolume $A.B:

   ```
   ADD DISKFILE-PATTERN $A.B.*, ACCESS *.* (R,W)
   ```

3. To add a diskfile-pattern protection record for every disk file named FILE followed by one alphanumeric character that is in a subvolume REPORT on every volume beginning with $DATA:

   ```
   ADD DISKFILE-PATTERN $DATA*.REPORT.FILE?
   ```

ALTER DISKFILE Command

ALTER DISKFILE changes one or more disk file attribute values in an existing disk file authorization record. The primary owner of a disk file, the primary owner’s group manager, the local super ID, and any user with OWNER authority on the ACL can change a disk file authorization record.

Except for the ACCESS attribute, new attribute values specified in an ALTER DISKFILE command replace the existing attribute value with the specified value. Using ALTER DISKFILE to specify a new ACCESS *access-spec* adds the new *access-*

spec to the existing ACL. To remove authorities previously granted to users, use the minus-sign (-) form of *access-spec*.

```
ALTER DISKFILE filename-list [ , ]

    { LIKE disk-file-name | disk-file-attribute }

    [ , disk-file-attribute ] ...
```

DISKFILE

specifies DISKFILE as the object type of the ALTER command. Omit it if DISKFILE is the assumed object type. (For more information on assumed object types, see the ASSUME Command on page 4-3.)

filename-list

specifies one or more disk files whose existing *disk-file-attribute*s are to be changed. All disk files specified must already have Safeguard authorization records (created with the ADD DISKFILE command).

filename-list can be either of:

```
  disk-file-name

( disk-file-name [ , disk-file-name ] ... )
```

disk-file-name

can be any disk-file name. The name can contain wild-card characters.

LIKE *disk-file-name*

changes the attribute values of the disk files specified with *filename-list* to the same as the existing attribute values for *disk-file-name*. For the ACCESS attribute, LIKE only adds ACL entries or adds authorities to existing entries. It does not replace or delete ACL entries or authorities.

disk-file-name

identifies the disk file whose existing attribute values are to be assigned to the disk-file authorization record or records being altered. *disk-file-name* can be any disk-file name.

disk-file-attribute

changes the existing value of the specified disk file attribute for the disk file or files being altered. The disk file attributes are:

```
OWNER [owner-id]
ACCESS access-spec [ ; access-spec ] ...
LICENSE {ON|OFF}
PROGID {ON|OFF}
```

```
CLEARONPURGE {ON|OFF}
PERSISTENT {ON|OFF}
OBJECT-TEXT-DESCRIPTION "[any-text]"
RESET-OBJECT-TEXT-DESCRIPTION
AUDIT-ACCESS-PASS [audit-spec]
AUDIT-ACCESS-FAIL [audit-spec]
AUDIT-MANAGE-PASS [audit-spec]
AUDIT-MANAGE-FAIL [audit-spec]
WHERE option-list
WARNING-MODE {ON|OFF}
TRUST {ME|SHARED|OFF} (H-series only)
AUDIT-PRIV-LOGON { ON | OFF}
PRIV-LOGON { ON | OFF}
```

Note. The attributes, AUDIT-PRIV-LOGON and PRIV-LOGON, are supported only on systems running H06.11 and later H-series RVUs and G06.32 and later G-series RVUs.

`OWNER [`*`owner-id`*`]`

specifies the new owner of the disk file or files. *`owner-id`* can be either of:

```
[\node-spec.]group-name.member-name
[\node-spec.]group-num , member-num
```

If you omit *`owner-id`*, *`owner-id`* is set to your user ID.

`ACCESS `*`access-spec`*` [ ; `*`access-spec`*` ] ...`

changes the ACL for *`filename-list`* by adding or deleting ACL entries or by changing the authority list of a current access-control-list entry.

`PROCESS-ACCESS access-spec [ ; access-spec ] ...`

is used to set the default process access list for a process launched from a specific diskfile. It changes the process ACL for filename-list by adding or deleting entries, or changing the authority list of a current entry.

An ACL contains as many as 50 entries that grant or deny access authorities to users and user groups.

`access-spec` has the form:

```
user-list  [-] [DENY] authority-list

group-list [-] [DENY] authority-list
```

`user-list`

specifies users who are granted (or denied) the access authorities specified with the following *`authority-list`*. *`user-list`* can be either of:

```
  net-user-spec

( net-user-spec [ , net-user-spec ] ... )
```

net-user-spec can be any of:

```
[\node-spec.]adm-group-name.user-name
[\node-spec.]adm-group-num , user-num
[\node-spec.]adm-group-name.*
[\node-spec.]adm-group-num , *
[\node-spec.]*.*
[\node-spec.]*,*
```

-

(minus-sign) operates on the existing ACL entries. The minus-sign form of *access-spec* modifies the current default ACL. The *authority* entries are removed from the default ACL entries for the users specified with *user-list*.

group-list

can be either of:

net-group-spec

(*net-group-spec* [, *net-user-spec*] ...)

net-group-spec can take any of these forms:

GROUP [NAME][\node-spec.] *group-nam*e

GROUP NUMBER [\node-spec.]

node-spec

takes this form:

```
* | node-name | node-number
```

node-name

specifies the system name.

node-number

specifies the Expand node number.

adm-group-name

specifies the name of the administrative group.

admin-group-name

specifies the group number of an administrative group.

group-name

specifies the name of any group.

group-num

specifies the group number of any group.

-

(minus-sign) operates on the existing ACL entries. The minus-sign form of *access-spec* modifies the current default ACL. The *authority* entries are removed from the default ACL entries for the users specified with *user-list*.

DENY

denies the users or groups specified by *user-list* the access authorities specified by *authority-list*.

authority-list

specifies the access authorities to be granted (or denied) to *user-list*. *authority-list* can be any of:

```
  authority

( authority [ , authority ] ... )

  *
```

authority

is any one of:

```
R[EAD]
W[RITE]
E[XECUTE]
P[URGE]
C[REATE]
O[WNER]
```

*

indicates all the disk-file access authorities except CREATE authority (R, W, E, P, and O).

LICENSE {ON|OFF}

either licenses a program object file or revokes the license of a currently licensed program object file. (For more information about the LICENSE attribute, see SET DISKFILE Command on page 8-57.)

LICENSE ON

licenses all program object files specified with *filename-list*.

`LICENSE OFF`

revokes the license of all program object files specified with *`filename-list`*.

`PROGID {ON|OFF}`

changes the PROGID attribute of a program object file. When the PROGID attribute is set to ON, the process accessor ID (PAID) of a process that is executed from that object file is set to the user ID of the primary owner of the object file. When PROGID is OFF, the PAID of a process run from the object file is set to the user ID of the user who runs the process.

`PROGID ON`

the PROGID attribute is set to ON for all program object files specified with *`filename-list`*.

`PROGID OFF`

the PROGID attribute is set to OFF for all program object files specified with *`filename-list`*.

`CLEARONPURGE {ON|OFF}`

changes the CLEARONPURGE attribute for all the disk files in *`filename-list`*. The CLEARONPURGE attribute specifies whether the data pages of a disk file are physically cleared when the file is purged. (For a complete description of CLEARONPURGE, see SET DISKFILE Command on page 8-57.)

`CLEARONPURGE ON`

when a disk file is purged, its entry in the volume directory is deleted, and its data pages are physically cleared.

`CLEARONPURGE OFF`

when a disk file is purged, its entry in the volume directory is deleted.

`PERSISTENT {ON|OFF}`

changes the PERSISTENT attribute for all the disk files in *`filename-list`*. The PERSISTENT attribute specifies whether the authorization record for a disk file is retained if the disk file is purged.

`PERSISTENT ON`

indicates that the authorization record for the disk file is retained if the file is purged. If you purge a file with PERSISTENT ON and later create a file with the same name, that file assumes the authorization record associated with the old file.

`PERSISTENT OFF`

indicates that the authorization record for the disk file is deleted if the file is purged.

`OBJECT-TEXT-DESCRIPTION "[any-text]"`

allows you to store printable characters as comments. These comments are associated with the objects and are used to manage the object authorization record.

The text description field can accommodate 255 bytes of text data.

Note. The text specified in the text description field overwrites existing data, if any. Also, when LIKE clause is used with ALTER DISKFILE command, the OBJECT-TEXT-DESCRIPTION field is not copied with other object authorization record attributes. Also, if you specify OBJECT-TEXT-DESCRIPTION without any text in the quotation marks, the object text description for this record is removed.

The OBJECT-TEXT-DESCRIPTION attribute is supported only on systems running J06.05 and later J-series RVUs, H06.16 and later H-series RVUs, and G06.32 and later G-series RVUs.

`RESET-OBJECT-TEXT-DESCRIPTION`

Resets the object description to Null.

Note. The RESET-OBJECT-TEXT-DESCRIPTION attribute is supported only on systems running J06.05 and later J-series RVUs, H06.16 and later H-Series RVUs, and G06.32 and later G-series RVUs.

`AUDIT-ACCESS-PASS [`*`audit-spec`*`]`

changes the *audit-spec* for successful attempts to access the disk file. The form of *audit-spec* is:

```
{ ALL | LOCAL | REMOTE | NONE }
```

For a description of *audit-spec*, see the SET DISKFILE Command on page 8-57. Omitting *audit-spec* specifies NONE.

`AUDIT-ACCESS-FAIL [`*`audit-spec`*`]`

changes the *audit-spec* for unsuccessful attempts to access the disk file. The form of *audit-spec* is:

```
{ ALL | LOCAL | REMOTE | NONE }
```

For a description of *audit-spec*, see the SET DISKFILE Command on page 8-57. Omitting *audit-spec* specifies NONE.

`AUDIT-MANAGE-PASS [`*`audit-spec`*`]`

changes the *`audit-spec`* for successful attempts to change or read a disk file authorization record. The form of *`audit-spec`* is:

```
{ ALL | LOCAL | REMOTE | NONE }
```

For a description of *`audit-spec`*, see the SET DISKFILE Command on page 8-57. Omitting *`audit-spec`* specifies NONE.

`AUDIT-MANAGE-FAIL [`*`audit-spec`*`]`

changes the *`audit-spec`* for unsuccessful attempts to change or read a disk file authorization record. The form of *`audit-spec`* is:

```
{ ALL | LOCAL | REMOTE | NONE }
```

For a description of *`audit-spec`*, see the SET DISKFILE Command on page 8-57. Omitting *`audit-spec`* specifies NONE.

`WHERE` *`option-list`*

specifies that only disk files in *`filename-list`* that have LICENSE, PROGID, WARNING-MODE, TRUST ME, or TRUST SHARED set are to be altered.

`option-list` `has the form:`

```
[ ( ] option [ OR option ] [ ) ]
```

`option`

can be one of:

```
PROGID
LICENSE
WARNING-MODE
TRUSTME (H-series only)
TRUSTSHARED (H-series only)
PRIV-LOGON { ON | OFF }
```

`WARNING-MODE { ON | OFF }`

defines whether the warning mode is enabled for the specified disk file. The value is required. For more information on warning mode, see the *Safeguard Administrator's Manual*.

ON enables warning mode for the specified disk file. The initial value is OFF, which disables warning mode for the specified disk file.

`TRUST { ME | SHARED | OFF }`

sets the TRUST attribute for the specified disk file. The disk file must be a program object file. This attribute is valid only on systems running H-series RVUs. Only the super ID can set this attribute.

`PRIV-LOGON { ON | OFF}`

establishes whether the program file (object disk file) can request additional logon related sensitive features. When set to ON, a process created from this program file can request a logon without specifying a password.

A process originated from a program file calling USER_AUTHENTICATE_ with a 2 and 15 bit set to ON, the requesting user for authentication need not give a password. Even with wrong password the user will be able to logon successfully as bit 2 and 15 in the options field. In case of only bit 2 set to 1 and bit 15 as 0; no fail delay will take place. That is, no failure delay will be imposed even after three attempts with wrong password. The authentication will not be successful but there will be no delay imposed.

Also establishes whether the program file (object disk file) can request a delay to be imposed for failed logon attempts. When set to ON, a process created from this program file is not subjected to logon failure delays.

OFF is the initial value.

PRIV-LOGON may also be used in the WHERE expression of a command to restrict scope of that command to files with PRIV-LOGON ON.

Considerations

- The relationship between ALTER DISKFILE and FUP commands

  After a disk-file authorization record is created, the FUP GIVE, SECURE, LICENSE, and REVOKE commands no longer work for that disk file. You must use the SAFECOM ALTER DISKFILE command to perform the equivalent operations.

  The following list of command pairs shows the equivalent SAFECOM commands to use for the disabled FUP commands:

```
FUP GIVE filename-list , group-num,member-num
ALTER DISKFILE filename-list , OWNER owner-id

FUP SECURE filename-list , PROGID
ALTER DISKFILE filename-list , PROGID ON

FUP REVOKE filename-list , PROGID
ALTER DISKFILE filename-list , PROGID OFF

FUP SECURE filename-list , CLEARONPURGE
ALTER DISKFILE filename-list , CLEARONPURGE ON

FUP REVOKE filename-list , CLEARONPURGE
ALTER DISKFILE filename-list , CLEARONPURGE OFF

FUP LICENSE filename-list
ALTER DISKFILE filename-list , LICENSE ON

FUP REVOKE filename-list
ALTER DISKFILE filename-list , LICENSE OFF
```

The following two commands perform similar functions but are not strictly equivalent:

```
FUP SECURE filename-list , "security-string"
ALTER DISKFILE filename-list , ACCESS access-spec
   [ ; access-spec ] ...
```

An *access-spec* can include or deny specific users or groups of users to which the owner does not belong. A *security-string* does not have this flexibility.

- Altering a disk file that is currently open

  Using ALTER DISKFILE to change one or more attributes for a disk file has no effect on any users currently accessing the disk file. (That is, changing file security attributes has no effect on processes that currently have the disk file open.)

  For example, if you change a disk-file ACL to deny Read access to a user who is running a process that is currently accessing the file, the process can continue accessing the file until it closes the file. However, when the process attempts to reopen the file for Read access, the Safeguard software returns a security violation (file error 48).

Examples

- The owner of the file $DATA.KEEP.INFO adds three ACL entries, provides their description, and changes another entry:

  ```
  =ALTER DISKFILE $data.keep.info,OBJECT-TEXT-DESCRIPTION&
  “Record Created”, ACCESS 86,8 (r,w,e) ; &
  =86,10 (r,w,e); prs.darlene DENY (w,e,p) ; 86,* - e
  ```

  Now the users who have user IDs 86,8 and 86,10 can read, write, and execute this file, and user PRS.DARLENE cannot write, execute, or purge the file. The ACL entry for group 86 is changed so that members of group 86 no longer have EXECUTE authority for the file.

- The super ID uses this command to alter authorization records for all files on the volume $DATA that have either the PROGID or LICENSE attribute set ON. The Safeguard authorization records specify auditing for all attempts to access or manage these files.

  ```
  =ALTER DISKFILE $data1.*.*, AUDIT ALL &
  =WHERE (PROGID OR LICENSE)
  ```

- This example sets WARNING-MODE to OFF in files where WARNING-MODE is ON in the `$data.lawsuit` subvolume.

  ```
  =ALTER DISKFILE $data.lawsuit.*, WARNING-MODE OFF WHERE &
  =WARNING-MODE
  ```

- This example creates a process and grants purge permissions to a specific user or group.

  ```
  alter diskfile $data.vol.test,process-access x.y p
  ```

A process is created from the object `test` such that purge permission is granted to user `x.y`.

ALTER DISKFILE-PATTERN Command

ALTER DISKFILE-PATTERN changes one or more diskfile-pattern attribute values in an existing diskfile-pattern authorization record. The primary owner of a pattern, the primary owner’s group manager, the local super ID, and any user with OWNER authority on the ACL can change a pattern authorization record.

Except for the ACCESS attribute, new attribute values specified in an ALTER DISKFILE command replace the existing attribute value with the specified value. Using ALTER DISKFILE to specify a new ACCESS *access-spec* adds the new *access-spec* to the existing ACL. To remove authorities previously granted to users, use the minus-sign (-) form of *access-spec*.

```
ALTER DISKFILE-PATTERN pattern-spec-list [ , ][ ALL ]
[ WHERE option-list ] [ , ]
[ LIKE pattern-spec | pattern-attribute ]
[ , pattern-attribute ] ...
```

pattern-spec-list

is the same as the corresponding non-pattern object types. That is, a PATTERN-SPEC-LIST is a comma-separated list of one or more PATTERN-SPEC attributes. (*pattern-spec* [, *pattern-spec*] . . .)

`ALL`

instructs Safeguard to use all the wildcard characters as a part of the search string, not as part of the pattern.

`LIKE` *pattern-spec*

adopts the existing attribute values of *pattern-spec* as the *pattern-attribute* values to be used for the authorization record or records being added.

pattern-spec

are the characters that define the pattern that describe a set of objects. The PATTERN-SPEC for a diskfile pattern is a fully qualified diskfile name that contains at least one wildcard in either the subvolume or file name component that includes the following components:

- A volume name, which will include only valid volume characters; that is, wildcard characters are not part of the pattern, and if present, imply a one-dimensional search. No wildcard characters are allowed in the volume when used for a LIKE operation.

- A subvolume name, which might include wildcard characters and valid subvolume characters.
- A file name, which might include wildcard characters and valid file name characters.

`WHERE` *`option-list`*

specifies that only disk files in *`filename-list`* that have WARNING-MODE set are to be altered.

`option-list` has the form:

```
[ ( ] option [ OR option ] [ ) ]
```

`option`

can be:

```
WARNING-MODE
```

`pattern-attribute`

defines a pattern attribute value for the diskfile-pattern authorization record or records being added. The pattern attributes are:

```
OWNER [owner-id]
ACCESS access-spec [ ; access-spec ] ...
AUDIT-ACCESS-PASS [audit-spec]
AUDIT-ACCESS-FAIL [audit-spec]
AUDIT-MANAGE-PASS [audit-spec]
AUDIT-MANAGE-FAIL [audit-spec]
WARNING-MODE {ON|OFF}
```

`OWNER [`*`owner-id`*`]`

specifies the new owner of the disk file or files. *`owner-id`* can be either of:

```
[\*.]group-name.member-name
[\*.]group-num , member-num
```

If you omit *`owner-id`*, *`owner-id`* is set to your user ID.

`ACCESS` *`access-spec`* `[ ;` *`access-spec`* `] ...`

changes the ACL for *`filename-list`* by adding or deleting ACL entries or by changing the authority list of a current ACL entry.

An ACL contains as many as 50 entries that grant or deny access authorities to users and user groups.

access-spec has the form:

user-list [-] [DENY] *authority-list*

group-list [-] [DENY] *authority-list*

user-list

specifies users who are granted (or denied) the access authorities specified with the following *authority-list*. *user-list* can be either of:

net-user-spec

(*net-user-spec* [, *net-user-spec*] ...)

net-user-spec can be any of:

[\node-spec.]*adm-group-name*.*user-name*
[\node-spec.]*adm-group-num* , *user-num*
[\node-spec.]*adm-group-name*.*
[\node-spec.]*adm-group-num* , *
[\node-spec.]*.*
[\node-spec.]*,*

-

(minus-sign) operates on existing ACL entries. The minus-sign form of *access-spec* modifies the current default ACL. The *authority* entries are removed from the default ACL entries for the users specified with *user-list*.

group-list

can take either of these forms:

net-group-spec

(*net-group-spec* [, *net-user-spec*] ...)

net-group-spec

can take any of these forms:

GROUP [NAME][\node-spec.] *group-nam*e

GROUP NUMBER [\node-spec.]

node-spec

has the form:

* | *node-name* | *node-number*

node-name

specifies the system name.

node-number

specifies the Expand node number.

adm-group-name

specifies the name of the administrative group.

admin-group-name

specifies the group number of an administrative group.

group-name

specifies the name of any group.

group-num

specifies the group number of any group.

-

(minus-sign) operates on the existing ACL entries. The minus-sign form of *access-spec* modifies the current default ACL. The *authority* entries are removed from the default ACL entries for the users specified with *user-list*.

Note. Specifying ACCESS *access-spec* through the ADD command does not override the current default ACL (established through the SET command). Instead, any ACL entries specified with the ADD command are added to the current default ACL, and the entire ACL is defined for the disk file whose authorization record is being added.

DENY

denies the users or groups specified by *user-list* the access authorities specified by *authority-list*.

authority-list

specifies the access authorities to be granted (or denied) to *user-list*. *authority-list* can be any one of:

```
  authority
( authority [ , authority ] ... )
  *
```

authority

is any one of:

```
R[EAD]
W[RITE]
E[XECUTE]
P[URGE]
C[REATE]
O[WNER]
```

`*`

(asterisk) specifies all the disk-file access authorities (R, W, E, P, C, and O).

`AUDIT-ACCESS-PASS [`*`audit-spec`*`]`

changes the *audit-spec* for successful attempts to access the diskfile pattern. The form of *audit-spec* is:

```
{ ALL | LOCAL | REMOTE | NONE }
```

For a description of *audit-spec*, see the SET DISKFILE Command on page 8-57. Omitting *audit-spec* specifies NONE.

`AUDIT-ACCESS-FAIL [`*`audit-spec`*`]`

changes the *audit-spec* for unsuccessful attempts to access the diskfile pattern. The form of *audit-spec* is:

```
{ ALL | LOCAL | REMOTE | NONE }
```

For a description of *audit-spec*, see the SET DISKFILE Command on page 8-57. Omitting *audit-spec* specifies NONE.

`AUDIT-MANAGE-PASS [`*`audit-spec`*`]`

changes the *audit-spec* for successful attempts to manage (change or read) a diskfile-pattern authorization record. The form of *audit-spec* is:

```
{ ALL | LOCAL | REMOTE | NONE }
```

For a description of *audit-spec*, see the SET DISKFILE Command on page 8-57. Omitting *audit-spec* specifies NONE.

`AUDIT-MANAGE-FAIL [`*`audit-spec`*`]`

changes the *audit-spec* for unsuccessful attempts to manage (change or read) a diskfile-pattern authorization record. The form of *audit-spec* is:

```
{ ALL | LOCAL | REMOTE | NONE }
```

For a description of *audit-spec*, see the SET DISKFILE Command on page 8-57. Omitting *audit-spec* specifies NONE.

```
WARNING-MODE { ON | OFF }
```

defines whether the warning mode is enabled for the specified diskfile pattern. The value is required. For more information on warning mode, see the *Safeguard Administrator's Manual*.

ON enables warning mode for the specified diskfile pattern. The initial value is OFF, which disables warning mode for the specified diskfile pattern.

Considerations

- Altering a disk file that is currently open

  Using ALTER DISKFILE-PATTERN to change one or more attributes for a disk file described by that pattern has no effect on any users currently accessing the disk file. (That is, changing file security attributes has no effect on processes that currently have the disk file open.)

  For example, if you change a disk-file ACL to deny Read access to a user who is running a process that is currently accessing the file, the process can continue accessing the file until it closes the file. However, when the process attempts to reopen the file for Read access, the Safeguard software returns a security violation (file error 48).

Examples

1. To alter a diskfile pattern to give SUPER.SUPER read and write access:

   ```
   ALTER DISKFILE-PATTERN $DATA.APLOGS.LOG*, &
   ACCESS SUPER.SUPER (R,W)
   ```

2. To alter all diskfile pattern that match $DATA*.APLOGS.LOG*:

   ```
   ALTER DISKFILE-PATTERN $DATA*.APLOGS.LOG*, ALL, &
   ACCESS SUPER.SUPER (R,W)
   ```

   The above command would alter the diskfile patterns for all matching patterns. For example, if the following patterns exist, they are altered as:

   - $DATA01.APLOGS.LOGAPR*
   - $DATA1.APLOGS.LOG*
   - $DATA123.APLOGS.LOG?
   - $DATA.APLOGS.LOG????
   - $DATABLE.APLOGS.LOGON?1A

DELETE DISKFILE Command

The DELETE DISKFILE command deletes the authorization record for a disk file. After a file-authorization record is deleted, the file is placed under the control of the standard

Guardian security system and is no longer subject to Safeguard authorization checks or Safeguard auditing.

Using DELETE DISKFILE to delete a disk-file authorization record does not delete the file. To delete a file, use the FUP PURGE command, the PURGE command in the command interpreter, or the PURGE procedure. When a disk file is purged, its authorization record is automatically deleted.

The owner of a disk file, the primary owner's group manager, the local super ID, and any user with OWNER authority on the ACL can delete a disk file authorization record.

After deleting the authorization record for the disk file, the Safeguard software displays an OBSERVE message telling the user that the files in *filename-list* are returned to Guardian protection.

```
DELETE DISKFILE filename-list [ [ , ] WHERE option-list ]
```

DISKFILE

specifies DISKFILE as the object type of the DELETE command. Omit it if DISKFILE is the assumed object type. (For more information on assumed object types, see the ASSUME Command on page 4-3.)

filename-list

specifies one or more disk files whose authorization records are to be deleted. *filename-list* can be either:

disk-file-name

(*disk-file-name* [, *disk-file-name*] ...)

disk-file-name

can be any disk-file name. The name can contain wild-card characters.

WHERE *option-list*

specifies that only disk files in *filename-list* that have LICENSE, PROGID, WARNING-MODE, TRUST ME, or TRUST SHARED set are to be deleted.

option-list has the form:

```
[ ( ] option [ OR option ] [ ) ]
```

option

can be one of:

```
PROGID
LICENSE
WARNING-MODE
```

```
TRUSTME (H-series only)
TRUSTSHARED (H-series only)
```

Considerations

- Deleting a disk-file authorization record places the file under standard Guardian security.

  When you delete a disk-file authorization record, the disk file is no longer subject to Safeguard authorization checks and auditing. All subsequent accesses to the disk file are subject to standard Guardian disk-file security checks.

  When the Safeguard software deletes a disk-file authorization record, the *security-string* for the file is reset to the original value. (That is, the file receives the security it had before being added to Safeguard protection.)

  Deleting a disk-file authorization record does not change these disk-file attributes:

LICENSE ON	The disk file remains licensed.
PROGID ON	The PROGID security option remains in effect.
CLEARONPURGE ON	The CLEARONPURGE security option remains in effect.

- Users who have PURGE authority can delete a disk file’s authorization record.

  Only an owner of a disk file, the primary owner’s group manager, and the super ID can delete a disk file authorization record through DELETE DISKFILE. However, any user who can purge the disk file itself can delete the file authorization record unless PERSISTENT protection is specified for the file. (Purging a file that does not have PERSISTENT protection deletes the Safeguard record for the file.)

Example

The PRS group manager (who has user ID 86,255) deletes the authorization record for the file $DATA.KEEP.INFO:

```
=DELETE DISKFILE $data.keep.info
```

DELETE DISKFILE-PATTERN Command

DELETE DISKFILE-PATTERN deletes the pattern authorization record for a diskfile pattern.

The owner of a diskfile pattern, the primary owner’s group manager, the local super ID, and any user with OWNER authority on the ACL can delete a diskfile-pattern authorization record.

```
DELETE DISKFILE-PATTERN pattern-spec-list
[ , ] [ ALL ] [ WHERE option-list ]
```

pattern-spec-list

is the same as the corresponding non-pattern object types. That is, a PATTERN-SPEC-LIST is a comma-separated list of one or more PATTERN-SPEC attributes. (*pattern-spec* [, *pattern-spec*] . . .)

pattern-spec

are the characters that define the pattern that describe a set of objects. The PATTERN-SPEC for a diskfile pattern is a fully qualified diskfile name that contains at least one wildcard in either the subvolume or file name component that includes the following components:

- A volume name, which will include only valid volume characters; that is, wildcard characters are not part of the pattern, and if present, imply a one-dimensional search.
- A subvolume name, which might include wildcard characters and valid subvolume characters.
- A file name, which might include wildcard characters and valid file name characters.

WHERE *option-list*

specifies that only disk files in *filename-list* that have WARNING-MODE set are to be altered.

option-list has the form:

[( ] *option* [OR *option*] [ )]

option

can be:

WARNING-MODE

ALL

instructs Safeguard to use all the wildcard characters as a part of the search string, not as part of the pattern.

Consideration

- Deleting a diskfile-pattern authorization record places all files described by that pattern, that are not described by another pattern, under standard Guardian security.

  When you delete a diskfile-pattern authorization record, the disk files that are no longer described by diskfile patterns are subject to Safeguard authorization checks and auditing. All subsequent accesses to the diskfiles are subject to standard Guardian disk-file security checks.

When the Safeguard software deletes a diskfile pattern authorization record, the security-string for the files not protected by Safeguard are reset to the original value. (That is, the files receive the security it had before being added to Safeguard protection.)

Examples

1. To delete the diskfile pattern $ABC.*.*:

   ```
   DELETE DISKFILE-PATTERN $ABC.*.*
   ```

2. To delete all diskfile patterns that match the search pattern $ABC.*.*:

   ```
   DELETE DISKFILE-PATTERN $ABC.*.*, ALL
   ```

3. To delete all diskfile patterns that match the search pattern $AB*.D*.*F

   ```
   DELETE DISKFILE-PATTERN $AB*.D*.*F, ALL
   ```

FREEZE DISKFILE Command

FREEZE DISKFILE temporarily suspends the access authorities granted to users on a disk-file ACL. While a disk file is frozen, only the file owner, an owner on the ACL, the primary owner’s group manager, and the local super ID can access the file. Any other users get a security violation error (file error 48).

An owner of a disk file, the primary owner’s group manager, and the super ID can freeze a disk file.

Use THAW DISKFILE to restore all the access authorities that were in effect before the file was frozen.

```
FREEZE DISKFILE filename-list [ [ , ] WHERE option-list ]
```

`DISKFILE`

specifies DISKFILE as the object type of the FREEZE command. Omit it if DISKFILE is the assumed object type. (For more information on assumed object types, see the ASSUME Command on page 4-3.)

`filename-list`

specifies the disk file or files for which access is to be frozen. *`filename-list`* can be either:

```
  disk-file-name

( disk-file-name [ , disk-file-name ] ... )
```

disk-file-name

can be any disk-file name. The name can contain wild-card characters.

WHERE *option-list*

specifies that only disk files in *filename-list* that have LICENSE, PROGID, WARNING-MODE, TRUST ME, or TRUST SHARED set are to be frozen.

option-list has the form:

[( ] *option* [OR *option*] [ )]

option

can be one of:

```
PROGID
LICENSE
WARNING-MODE
TRUSTME (H-series only)
TRUSTSHARED (H-series only)
```

Considerations

Freezing a disk file that is currently open has no effect on any processes that have the file open. However, after those processes close the file, any attempt to reopen the file returns a security violation (file error 48). Before freezing a disk file, use the FUP LISTOPENS command to determine whether the file is currently open.

Examples

User PRS.HARRY (user ID 86,2) uses INFO DISKFILE to display the current status of a file:

```
=ASSUME DISKFILE
=INFO $data.harry.sales
```

A brief report shows:

```
                 LAST-MODIFIED     OWNER      STATUS     WARNING-MODE
$DATA.HARRY
 SALES           8JUN87, 14:32       86,2     THAWED         OFF

    086,001      R
    086,002      R,W,E,P,  O
    086,255      R,W,E,P,  O
```

To freeze the file:

```
=FREEZE $data.harry.sales
```

To see the new status:

```
=INFO $data.harry.sales
```

A brief report shows:

```
                LAST-MODIFIED     OWNER      STATUS     WARNING-MODE
$DATA.HARRY
 SALES          9JUN87, 10:18       86,2     FROZEN         OFF

    086,001     R
    086,002     R,W,E,P,  O
    086,255     R,W,E,P,  O
```

FREEZE DISKFILE-PATTERN Command

FREEZE DISKFILE-PATTERN temporarily suspends the access authorities granted to users on a diskfile pattern ACL. While a diskfile pattern is frozen, only the pattern owner, an owner on the ACL, the primary owner’s group manager, and the local super ID can access the pattern. Any other users get a security violation error (file error 48).

An owner of a pattern, the primary owner’s group manager, and the super ID can freeze a diskfile pattern.

Use THAW DISKFILE-PATTERN to restore all the access authorities that were in effect before the pattern was frozen.

```
FREEZE DISKFILE-PATTERN pattern-spec-list
[ , ] [ ALL ] [WHERE option-list ]
```

pattern-spec-list

is the same as the corresponding non-pattern object types. That is, a PATTERN-SPEC-LIST is a comma-separated list of one or more PATTERN-SPEC attributes. (*pattern-spec* [, *pattern-spec*] . . .)

pattern-spec

are the characters that define the pattern that describe a set of objects. The PATTERN-SPEC for a diskfile pattern is a fully qualified diskfile name that contains at least one wildcard in either the subvolume or file name component that includes the following components:

- A volume name, which will include only valid volume characters; that is, wildcard characters are not part of the pattern, and if present, imply a one-dimensional search.
- A subvolume name, which might include wildcard characters and valid subvolume characters.
- A file name, which might include wildcard characters and valid file name characters.

WHERE *option-list*

specifies that only disk files in *filename-list* that have WARNING-MODE set are to be altered.

option-list has the form:

```
[ ( ] option [ OR option ] [ ) ]
```

option

can be:

```
WARNING-MODE
```

`ALL`

instructs Safeguard to use all the wildcard characters as a part of the search string, not as part of the pattern.

Consideration

- The FREEZE command enforces special access rules on the object when the protection record is frozen. In general, these rules specify that only the owner of the protection record, that owners group manager, and the local SUPER.SUPER users are permitted (R,W,E,P,C,O) access. The volume and subvolume object types restrict only create access to the above users.

Examples

1. To freeze all diskfile patterns that specify a subvolume name beginning with the characters TEST:

```
FREEZE DISKFILE-PATTERN $*.TEST*.*, ALL
```

INFO DISKFILE Command

INFO DISKFILE displays the attribute values currently stored in a disk-file authorization record. INFO DISKFILE produces two types of reports: brief and detailed. The formats for the two report types follow the syntax.

Any user can produce an INFO report for any disk file.

```
INFO [ / OUT listfile / ] DISKFILE filename-list [ , ]

   [ display-option ] [ , display-option ] ...
```

`OUT` *listfile*

directs the INFO DISKFILE report to *listfile*. After executing the INFO command, SAFECOM redirects its output to the current OUT file.

For *listfile*, specify any file name. SAFECOM opens *listfile* and appends the INFO report to the file. If *listfile* does not exist, SAFECOM creates an EDIT file by that name and writes the INFO report to that file.

`DISKFILE`

specifies DISKFILE as the object type for the INFO command. Omit it if DISKFILE is the assumed object type. (For more information on assumed object types, see the ASSUME Command on page 4-3.)

`filename-list`

specifies the disk file or files for which INFO DISKFILE reports are produced. *`filename-list`* can be either:

`disk-file-name`

`(` *`disk-file-name`* `[ ,` *`disk-file-name`* `] ... )`

`disk-file-name`

can be any disk-file name. A name can contain wild-card characters.

`display-option`

can be one of:

```
DETAIL [ ON | OFF ]
WARNINGS [ ON | OFF ]
WHERE option-list
```

`DETAIL [ ON | OFF ]`

allows certain additional information to be included in this INFO command.

`DETAIL [ ON ]`

adds the *`audit-spec`* variables defined for the file and the current values of the LICENSE, PROGID, CLEARONPURGE, and PERSISTENT attributes to the INFO report. For a full description of the four *`audit-spec`* variables, see the SET DISKFILE Command on page 8-57.

`DETAIL OFF`

inhibits the display of additional information for this command. If you omit the DETAIL option, DETAIL OFF is the default.

`WARNINGS [ ON | OFF ]`

allows the display of warning messages for this command to be inhibited.

`WARNINGS [ ON ]`

causes the display of warning messages for this command. If you omit the WARNINGS option, WARNINGS ON is the default.

`WARNINGS OFF`

inhibits the display of warning messages for this command.

WHERE *option-list*

causes an INFO report to be displayed for each disk file in *filename-list* that has LICENSE, PROGID, WARNING-MODE, TRUST ME, or TRUST SHARED set.

option-list has the form:

[( ] *option* [OR *option*] [ )]

option

can be one of:

```
PROGID
LICENSE
WARNING-MODE
TRUSTME (H-series only)
TRUSTSHARED (H-series only)
```

INFO DISKFILE Brief Report

The brief INFO DISKFILE report gives you information about the disk file or files you specify. Figure 8-1 on page 8-45 shows the format of the brief INFO DISKFILE report.

Figure 8-1. INFO DISKFILE Brief Report Format

```
                 LAST-MODIFIED OWNER    STATUS   WARNING-MODE
$volume.subvol
 filename         date, time  owner-id  status     {ON|OFF}

    user-spec [DENY] authority-list
    user-spec [DENY] authority-list
       .
       .
       .
 [ NO ACCESS CONTROL LIST DEFINED! ]
```

Figure 8-1 contains these attribute values and status fields:

$*volume*.*subvol*
filename

is the name of the disk file whose existing attribute values are displayed.

LAST MODIFIED TIME
date, *time*

reports the date and time of the last change made to this disk-file authorization record. *date* and *time* are in local civil time.

OWNER
owner-id

is the user ID of the person who owns this disk file.

`STATUS`
`status`

is the current status of this disk file. *`status`* is either FROZEN or THAWED.

`WARNING-MODE`
`{ON|OFF}`

is the current warning-mode state of this disk file. ON indicates that the protection record is in warning mode. The initial value is OFF, which indicates that warning mode is disabled for this disk file.

`user-spec` `[DENY]` *`authority-list`*

is an entry in the ACL for this disk file. *`user-spec`* identifies a single user or user group. *`authority-list`* is a list of single-character codes that represent the access authorities granted to the user or group identified by *`user-spec`*. DENY indicates that the access authorities specified with *`authority-list`* are specifically denied to the user or group identified by *`user-spec`*.

`user-spec` can be any of:

```
group-num , member-num
group-num , *
*,*
\node-spec.group-num , member-num
\node-spec.group-num , *
\node-spec.*,*
```

`node-spec`

has this form:

`* | node-name | node-number`

`node-name`

specifies the system name.

`node-number`

specifies the Expand node number.

`group-num, member-num`

identifies a single local user.

`group-num,*`

identifies all the local users in the group that has *`group-num`*.

`*,*`

identifies all the local users at the node where this disk file resides.

`\`*`node-spec`*`.`*`group-num`*`, `*`member-num`*

identifies both the local user with user ID *`group-num`*, *`member-num`* and a network user who has the same user name and user ID as that local user.

`\`*`node-spec`*`.`*`group-num`*`,*`

identifies all the local users in the group identified by *`group-num`* and all network users whose *`group-num`* and *`group-name`* match those of the local group.

`\`*`node-spec`*`.*,*`

identifies all the local users on *`node-spec`* where this disk file resides and all network users who have access to this node.

`authority-list` can contain any of these codes:

R	READ authority
W	WRITE authority
E	EXECUTE authority
P	PURGE authority
C	CREATE authority
O	OWNER authority

`[ NO ACCESS CONTROL LIST DEFINED! ]`

indicates that no ACL entries are defined for this file. Use ALTER DISKFILE . . . ACCESS to define ACL entries for an existing file-authorization record. Only the local super ID can access a disk file for which no ACL is defined.

INFO DISKFILE Detailed Report

The detailed INFO DISKFILE report includes the auditing specifications currently defined for the protected file. Figure 8-2 shows the format of the detailed INFO DISKFILE report.

Figure 8-2. INFO DISKFILE Detailed Report Format

```
                 LAST-MODIFIED OWNER    STATUS   WARNING-MODE
$volume.subvol
 filename         date, time  owner-id  status     {ON|OFF}

    user-spec [DENY] authority-list
    user-spec [DENY] authority-list
       .
       .
       .
 [ NO ACCESS CONTROL LIST DEFINED! ]

   OBJECT-TEXT-DESCRIPTION =

  AUDIT-ACCESS-PASS = a-spec  AUDIT-MANAGE-PASS = a-spec
  AUDIT-ACCESS-FAIL = a-spec  AUDIT-MANAGE-FAIL = a-spec

  AUDIT-PRIV-LOGON { ON | OFF }

  LICENSE={ON|OFF} PROGID={ON|OFF} CLEARONPURGE={ON|OFF} PERSISTENT={ON|OFF}
  TRUST={ME|SHARED|OFF}            PRIV-LOGON { ON | OFF }
```

In addition to the disk-file attribute values displayed in the brief INFO DISKFILE report, the detailed INFO DISKFILE report displays these disk-file attribute values:

AUDIT-ACCESS-PASS = *a-spec* AUDIT-MANAGE-PASS = *a-spec*
AUDIT-ACCESS-FAIL = *a-spec* AUDIT-MANAGE-FAIL = *a-spec*

These values indicate the conditions under which the Safeguard software audits attempts to access this disk file or to change or read this file-authorization record. *a-spec* can be:

{ ALL | LOCAL | REMOTE | NONE }

For a complete description of each *a-spec*, see the appropriate *audit-spec* under the SET DISKFILE command.

LICENSE = {ON|OFF}

for program object files containing privileged code, indicates whether the object file is licensed for use by users other than the local super ID.

ON

This program object file is licensed for use by any user.

OFF

This program object file can be executed only by the local super ID.

PROGID = {ON|OFF}

for program object files, indicates whether the process accessor ID (PAID) of a process that is run from this object file is set to the user ID of the user who runs the process or to the user ID of the disk file’s primary owner.

`ON`

The PAID of a process that is run from this program object file is set to the user ID of the owner of this disk file.

`OFF`

The PAID of a process that is run from this program object file is set to the user ID of the user that runs the process.

`CLEARONPURGE = {ON|OFF}`

indicates whether the data pages for this disk file are physically cleared when the file is purged.

`ON`

When this disk file is purged, all its data pages on disk are physically cleared.

`OFF`

When this disk file is purged, its data pages on disk are returned to the pool of free pages for later allocation to other disk files, but they are not physically cleared.

`PERSISTENT = {ON|OFF}`

indicates whether the PERSISTENT attribute is set for this disk file. The PERSISTENT attribute specifies whether the authorization record for a disk file is retained if the disk file is purged.

`PERSISTENT ON`

indicates the authorization record for the disk file is retained if the file is purged. If you purge a file with PERSISTENT ON and later create a file with the same name, that file assumes the authorization record associated with the old file.

`PERSISTENT OFF`

indicates that the authorization record for the disk file is deleted if the file is purged.

`TRUST = {ME|SHARED|OFF}`

is the current setting of the TRUST attribute for a program object file. This attribute appears only on systems running H-series RVUs.

`ME`

The program can be trusted to not access the I/O buffers private to the process before I/O completion.

SHARED

The program can be trusted to not access the buffers that are private to the process or are shared with another process that also has TRUST SHARED set, before I/O completion.

OFF

The program is not to be trusted.

Considerations

When you specify WHERE LICENSE or WHERE PROGID, the INFO report includes all files with LICENSE or PROGID set, not just files protected by the Safeguard software. For files not protected by the Safeguard software, the following message is issued (unless the WARNINGS OFF option is used):

`* WARNING * RECORD FOR DISKFILE` *`filename`* `: NOT FOUND`

Examples

- Any user on the system can display Safeguard status information about the file $DATA.BENNY.PROFIT:

  ```
  =INFO DISKFILE $data.benny.profit, DETAIL
  ```

```
                LAST-MODIFIED     OWNER    STATUS      WARNING-MODE
$DATA.BENNY
 PROFIT         18SEP87, 9:33      86,4    THAWED          ON

         086,002     R,W
         086,004     R,W,E,P,  O
         086,255     R,W,E,P,  O
          086,*       R

          PROCESS-ACCESS = 255,255 R,W
                           200,10  C

  OBJECT-TEXT-DESCRIPTION =

  AUDIT-ACCESS-PASS = ALL       AUDIT-MANAGE-PASS = ALL
  AUDIT-ACCESS-FAIL = ALL       AUDIT-MANAGE-FAIL = NONE

  AUDIT-PRIV-LOGON = OFF

  LICENSE = OFF  PROGID = OFF  CLEARONPURGE = OFF  PERSISTENT = OFF
                                 PRIV-LOGON = OFF
```

This detailed INFO report shows that the file PROFIT belongs to the user who has user ID 86,4 (PRS.BENNY). The file-authorization record was last modified on September 18, 1987, at 9:33 a.m. The ACL for PROFIT indicates that all members of the PRS group can read the file, the user who has user ID 86,2 can read and write to the file, and both PRS.BENNY and his group manager have all four access authorities. The Safeguard software audits all successful or unsuccessful attempts to access the file and all successful attempts to change or read the disk file authorization record. The protection record is in warning mode.

- The following command produces a detailed INFO report for all disk files on the volume $DEV that have the LICENSE or PROGID attribute set ON:

  ```
  =INFO DISK $DEV.*.*, DETAIL, WHERE (LICENSE OR PROGID)
  ```

INFO DISKFILE-PATTERN Command

INFO DISKFILE-PATTERN displays the attribute values currently stored in a diskfile-pattern authorization record. INFO DISKFILE-PATTERN produces two types of reports: brief and detailed.

Any user can produce an INFO report for any disk file.

```
INFO [ / OUT listfile / ] DISKFILE-PATTERN pattern-spec-list
[ , ] [ ALL ] [ WHERE option-list ]
[ , ] [ display-option ] ...
```

OUT *listfile*

directs the INFO DISKFILE report to *listfile*. After executing the INFO command, SAFECOM redirects its output to the current OUT file.

For *listfile*, specify any file name. SAFECOM opens *listfile* and appends the INFO report to the file. If *listfile* does not exist, SAFECOM creates an EDIT file by that name and writes the INFO report to that file.

DISKFILE-PATTERN

specifies DISKFILE-PATTERN as the object type for the INFO command. Omit it if DISKFILE-PATTERN is the assumed object type. (For more information on assumed object types, see the ASSUME Command on page 4-3.)

pattern-spec-list

is the same as the corresponding non-pattern object types. That is, a PATTERN-SPEC-LIST is a comma-separated list of one or more PATTERN-SPEC attributes. (*pattern-spec* [, *pattern-spec*] . . .)

ALL

instructs Safeguard to use all the wildcard characters as a part of the search string, not as part of the pattern.

pattern-spec

are the characters that define the pattern that describe a set of objects. The PATTERN-SPEC for a diskfile pattern is a fully qualified diskfile name that contains at least one wildcard in either the subvolume or file name component that includes the following components:

- A volume name, which will include only valid volume characters; that is, wildcard characters are not part of the pattern, and if present, imply a one-dimensional search.
- A subvolume name, which might include wildcard characters and valid subvolume characters.
- A file name, which might include wildcard characters and valid file name characters.

display-option

can be one of:

```
DETAIL [ ON | OFF ]
WARNINGS [ ON | OFF ]
WHERE option-list
```

`DETAIL [ ON | OFF ]`

allows certain additional information to be included in this INFO command.

`DETAIL [ ON ]`

adds the *audit-spec* variables defined for the file to the INFO report. For a full description of the four *audit-spec* variables, see the SET DISKFILE Command on page 8-57.

`DETAIL OFF`

inhibits the display of additional information for this command. If you omit the DETAIL option, DETAIL OFF is the default.

`WARNINGS [ ON | OFF ]`

allows the display of warning messages for this command to be inhibited.

`WARNINGS [ ON ]`

causes the display of warning messages for this command. If you omit the WARNINGS option, WARNINGS ON is the default.

`WARNINGS OFF`

inhibits the display of warning messages for this command.

WHERE *option-list*

causes an INFO report to be displayed for each diskfile pattern in *filename-list* that has WARNING-MODE set.

option-list has the form:

```
[ ( ] option [ OR option ] [ ) ]
```

`option`

can be:

`WARNING-MODE`

Examples

1. To display the diskfile pattern $DATA.*TEST.* (that is, display a single diskfile pattern) using display user as name:

```
=DISPLAY USER AS NAME
```

```
=INFO DISKFILE-PATTERN $DATA.*TEST.*
```

This information appears:

```
              LAST-MODIFIED   OWNER    STATUS WARNING-MODE
$DATA.*TEST
              * 28SEP04, 5:44 MLH1.MGR THAWED      OFF

\KONA.PROD.CARLY R
\KONA.TEST.JIMMY R,W
GROUP TEST       R,W,E,P,C
GROUP \KONA.TEST R
\*.*.*           R
```

2. To display the diskfile pattern $DATA.*TEST.*, DETAIL (that is, display a single diskfile pattern) using display user as name:

```
=INFO DISKFILE-PATTERN $DATA.*TEST.*,DETAIL
```

This information appears:

```
              LAST-MODIFIED   OWNER    STATUS WARNING-MODE
$DATA.*TEST
              * 28SEP04, 5:44 MLH1.MGR THAWED      OFF

\KONA.PROD.CARLY R
\KONA.TEST.JIMMY R,W
GROUP TEST       R,W,E,P,C
GROUP \KONA.TEST R
\*.*.*           R

AUDIT-ACCESS-PASS = NONE       AUDIT-MANAGE-PASS = NONE
AUDIT-ACCESS-FAIL = NONE       AUDIT-MANAGE-FAIL = NONE

      CREATION                LAST-MODIFIED

USER NAME SUPER.SUPER           testman
USER TYPE USER (ID 255,255)     ALIAS (ID 164,255)
USER NODE LOCAL                 LOCAL
TIMESTAMP 28SEP2004, 05:28:48.870  28SEP2004, 05:44:22.588
```

3. To display the diskfile pattern $DATA.*TEST.* (that is, display a single diskfile pattern) using display user as number:

   ```
   =DISPLAY USER AS NUMBER
   ```

   ```
   =INFO DISKFILE-PATTERN $DATA.*TEST.*
   ```

   This information appears:

   ```
                  LAST-MODIFIED   OWNER   STATUS WARNING-MODE
   $DATA.*TEST
                  * 28SEP04, 5:44 164,255 THAWED      OFF

   \205.011,100     R
   \205.200,002     R,W
   GROUP 00200      R,W,E,P,C
   GROUP \205.00200 R
   \*.*,*             R
   ```

4. To display the diskfile patterns for all volumes starting with “$DATA” with subvolumes starting with “PROD” :

   ```
   INFO DISKFILE-PATTERN $DATA*.PROD*.*, ALL
   ```

5. To list all diskfile patterns that start with the letters $A.B.C:

   ```
   INFO DISKFILE-PATTERN $A.B.C*, ALL
   ```

6. To display multiple diskfile patterns that have warning-mode enabled:

   ```
   INFO DISKFILE-PATTERN $*.*.*, ALL, WHERE WARNING-MODE
   ```

RESET DISKFILE Command

RESET DISKFILE resets the current default disk file attribute values to their predefined values.

When you add an authorization record for a disk file, the current default disk file attribute values are used for any attributes you do not specify in the ADD DISKFILE command. (To set the default disk file attributes to specific values, use the SET DISKFILE command.)

```
RESET DISKFILE [ [ , ] disk-file-attribute-keyword ]

     [ , disk-file-attribute-keyword ] ...
```

`DISKFILE`

specifies DISKFILE as the object type of the RESET command. Omit it if DISKFILE is the assumed object type. (For more information on assumed object types, see the ASSUME Command on page 4-3.)

disk-file-attribute-keyword

sets the current default value of the *disk-file-attribute* indicated by *disk-file-attribute-keyword* to predefined values, as follows:

```
OWNER                   - User ID of the current SAFECOM user
ACCESS                  - Null (no access control list)
LICENSE                 - OFF
PROGID                  - OFF
CLEARONPURGE            - OFF
PERSISTENT              - OFF
OBJECT-TEXT-DESCRIPTION - Null (no descriptive text or blank)
AUDIT-ACCESS-PASS       - NONE (no auditing)
AUDIT-ACCESS-FAIL       - NONE (no auditing)
AUDIT-MANAGE-PASS       - NONE (no auditing)
AUDIT-MANAGE-FAIL       - NONE (no auditing)
WARNING-MODE            - OFF  (warning mode disabled)
TRUST                  - OFF  (H-series only)
```

For a complete description of *disk-file-attribute*, see the SET DISKFILE Command on page 8-57.

Considerations

- Specifying an attribute name without a value in an ADD or ALTER command causes the attribute to be assigned the predefined default value (as defined for the RESET DISKFILE Command on page 8-54).
- Executing RESET DISKFILE without specifying a *disk-file-attribute-keyword*

  If you enter RESET DISKFILE (or RESET when the assumed object type is DISKFILE) and you do not include any *disk-file-attribute-keyword*, all the disk file attributes are returned to their predefined values. The predefined values are listed in the preceding syntax description.

Examples

1. In this example, to display the values of the default disk-file attributes:

```
=SHOW DISKFILE
```

A brief report shows:

```
TYPE          OWNER    WARNING-MODE
 DISKFILE     33,6         OFF

  OBJECT-TEXT-DESCRIPTION =

  AUDIT-ACCESS-PASS = LOCAL       AUDIT-MANAGE-PASS = ALL
  AUDIT-ACCESS-FAIL = NONE        AUDIT-MANAGE-FAIL = LOCAL

  AUDIT-PRIV-LOGON = OFF
  LICENSE = OFF  PROGID = OFF  CLEARONPURGE = OFF  PERSISTENT = OFF
                               PRIV-LOGON = OFF

         033,001     R,  E
         033,002     R,W
         033,005     R,W
         033,006     R,W,E,P,  O
```

Then, to reset the disk-file attributes to their predefined values:

```
=RESET DISKFILE
```

Display the default values:

```
=SHOW DISKFILE
```

A brief report shows:

```
TYPE          OWNER    WARNING-MODE
 DISKFILE     33,6         OFF

  OBJECT-TEXT-DESCRIPTION =

  AUDIT-ACCESS-PASS = NONE       AUDIT-MANAGE-PASS = NONE
  AUDIT-ACCESS-FAIL = NONE       AUDIT-MANAGE-FAIL = NONE

  AUDIT-PRIV-LOGON = OFF

  LICENSE = OFF  PROGID = OFF  CLEARONPURGE = OFF  PERSISTENT = OFF
                               PRIV-LOGON = OFF

  NO ACCESS CONTROL LIST DEFINED!
```

2. This command resets the default OWNER attribute to the user ID of the current SAFECOM user and resets the AUDIT-ACCESS-PASS specification to NONE:

```
=RESET DISKFILE OWNER, AUDIT-ACCESS-PASS
```

RESET DISKFILE-PATTERN Command

RESET DISKFILE-PATTERN returns the named attribute to its predefined value from a default value that optionally had been specified via the SET command.

```
RESET DISKFILE-PATTERN [ [ , ] pattern-attribute-keyword ]
[ , pattern-attribute-keyword ] ...
```

pattern-attribute-keyword

sets the current default value of the *pattern-attribute* indicated by *pattern-attribute-keyword* to predefined values, as follows:

```
OWNER                - User ID of the current SAFECOM user
ACCESS               - Null (no access control list)
AUDIT-ACCESS-PASS    - NONE (no auditing)
AUDIT-ACCESS-FAIL    - NONE (no auditing)
AUDIT-MANAGE-PASS    - NONE (no auditing)
AUDIT-MANAGE-FAIL    - NONE (no auditing)
WARNING-MODE         - OFF (warning mode disabled)
```

For a complete description of *pattern-attribute*, see the SET DISKFILE-PATTERN command.

Considerations

- When you add a diskfile-pattern authorization record, the current default diskfile-pattern attribute values are used for any attributes you do not specify in the ADD DISKFILE-PATTERN command.
- Specifying an attribute name without a value in an ADD or ALTER command causes the attribute to be assigned the predefined default value (as defined for the RESET DISKFILE-PATTERN Command on page 8-56).
- Executing RESET DISKFILE-PATTERN without specifying a *pattern-attribute-keyword*

  If you enter RESET DISKFILE-PATTERN (or RESET when the assumed object type is DISKFILE-PATTERN) and you do not include any *pattern-attribute-keyword*, all the diskfile-pattern attributes are returned to their predefined values. The predefined values are listed in the preceding syntax description.

Example

1. To reset WARNING-MODE to its predefined value (OFF) for diskfile patterns:

   ```
   RESET DISKFILE-PATTERN WARNING-MODE
   ```

SET DISKFILE Command

SET DISKFILE establishes default values for one or more disk-file attributes. Later, when you add an authorization record for a disk file, the current default disk-file attribute values are used for any attributes you do not specify in your ADD DISKFILE command.

To display the current default disk-file attribute values, use the SHOW DISKFILE command.

```
SET DISKFILE [ , ]

   { LIKE disk-file-name | disk-file-attribute }

   [ , disk-file-attribute ] ...
```

`DISKFILE`

specifies DISKFILE as the object type of the SET command. Omit it if DISKFILE is the assumed object type. (For more information on assumed object types, see the ASSUME Command on page 4-3.)

`OBJECT-TEXT-DESCRIPTION`

allows you to store printable characters, which are associated with the objects, as comments. These comments can be used to manage the object authorization record.

The text description field can accommodate 255 bytes of text data.

Note. The text specified in the text description field overwrites existing data, if any. Also, when LIKE clause is used with SET DISKFILE command, the object text description field is not copied with other object authorization record attributes.

The OBJECT-TEXT-DESCRIPTION attribute is supported only on systems running H06.16 and later H-series RVUs.

`LIKE` *`disk-file-name`*

sets the current default *`disk-file-attribute`* values to the same as those currently defined for *`disk-file-name`*.

`disk-file-name`

identifies the disk file whose existing attribute values are to be the current default *`disk-file-attribute`* values. *`disk-file-name`* can be any disk-file name.

`disk-file-attribute`

defines a current default value for the specified disk-file attribute. The *`disk-file-attribute`*s are:

```
OWNER [owner-id]
ACCESS access-spec [ ; access-spec ] ...
LICENSE {ON|OFF}
PROGID {ON|OFF}
CLEARONPURGE {ON|OFF}
PERSISTENT {ON|OFF}
OBJECT-TEXT-DESCRIPTION "[any-text]"
```

```
AUDIT-ACCESS-PASS [audit-spec]
AUDIT-ACCESS-FAIL [audit-spec]
AUDIT-MANAGE-PASS [audit-spec]
AUDIT-MANAGE-FAIL [audit-spec]
WARNING-MODE {ON|OFF}
TRUST {ME|SHARED|OFF} (H-series only)
AUDIT-PRIV-LOGON { ON | OFF }
PRIV-LOGON { ON | OFF}
```

Note. The attributes, AUDIT-PRIV-LOGON and PRIV-LOGON, are supported only on systems running H06.11 and later H-series RVUs and G06.32 and later G-series RVUs.

`OWNER [`*`owner-id`*`]`

specifies the owner of a disk file. (A disk-file owner also owns the disk-file authorization record.) *owner-id* can be either of:

```
[\node-spec.]group-name.member-name
[\node-spec.]group-num , member-num
```

If you omit *owner-id*, *owner-id* is set to your user ID (that is, the process accessor ID of the current SAFECOM process).

`ACCESS` *`access-spec`* `[ ;` *`access-spec`* `] ...`

changes the ACL for *filename-list* by adding or deleting ACL entries or by changing the authority list of a current ACL entry.

An ACL contains as many as 50 entries that grant or deny access authorities to users and user groups.

access-spec has the form:

```
user-list  [-] [DENY] authority-list
```

```
group-list [-] [DENY] authority-list
```

user-list

specifies users who are granted (or denied) the access authorities specified with the following *authority-list*. *user-list* can be either of:

```
  net-user-spec
```

```
( net-user-spec [ , net-user-spec ] ... )
```

net-user-spec can be any of:

```
[\node-spec.]adm-group-name.user-name
[\node-spec.]adm-group-num , user-num
[\node-spec.]adm-group-name.*
[\node-spec.]adm-group-num , *
[\node-spec.]*.*
[\node-spec.]*,*
```

-

(minus-sign) operates on existing ACL entries. The minus-sign form of *access-spec* modifies the current default ACL. The *authority* entries are removed from the default ACL entries for the users specified with *user-list*.

group-list

can take either of the following forms:

net-group-spec

(*net-group-spec* [, *net-user-spec*] ...)

net-group-spec can take any of these forms:

GROUP [NAME][\node-spec.] *group-nam*e

GROUP NUMBER [\node-spec.]

node-spec

takes this form:

* | *node-name* | *node-number*

node-name

specifies the system name.

node-number

specifies the Expand node number.

adm-group-name

specifies the name of the administrative group.

admin-group-num

specifies the group number of an administrative group.

group-name

specifies the name of any group.

group-num

specifies the group number of any group.

-

(minus-sign) operates on the existing ACL entries. The minus-sign form of *access-spec* modifies the current default ACL. The *authority* entries

are removed from the default ACL entries for the users specified with *`user-list`*.

`DENY`

denies the users or user groups specified with the preceding *`user-list`* the access authorities specified in these *`authority-list`*.

`authority-list`

specifies the access authorities to be granted (or denied).

`authority-list` can be any of:

```
authority

( authority [ , authority ] ... )

  *
```

`authority`

is any one of the access authorities:

```
R[EAD]
W[RITE]
E[XECUTE]
P[URGE]
C[REATE]
O[WNER]
```

`*`

indicates all the disk-file access authorities except CREATE authority. (These access authorities include R, W, E, P, and O.)

`LICENSE { ON | OFF }`

has meaning only for disk files containing object code for privileged programs. (A privileged program contains callable or privileged procedures.) Normally only the super ID (255,255) can run privileged programs. Through the LICENSE attribute, the super ID can license a privileged program’s object file for use by other users. Only the super ID can license a file, but any owner of the file can revoke a license.

`LICENSE ON`

licenses all program object files in the `filename-list` of subsequent ADD DISKFILE commands.

`LICENSE OFF`

revokes the license of all program object files in the *`filename-list`* of subsequent ADD DISKFILE commands.

`PROGID {ON|OFF}`

has meaning only for program object disk files. The PROGID attribute affects the way a process's process accessor ID (PAID) is set when the process is run. Normally, a PAID is set to the user ID of the user who runs the process. But if PROGID for a program object file is set to ON when the program is run, the PAID of the resulting process is set to the user ID of the object file's primary owner.

`PROGID ON`

means that the PROGID attribute is set to ON for all program object files in the *filename-list* of subsequent ADD DISKFILE commands.

`PROGID OFF`

means that the PROGID attribute is set to OFF for all program object files specified in the *filename-list* of subsequent ADD DISKFILE commands.

`CLEARONPURGE {ON|OFF}`

specifies whether the data pages for a disk file are physically cleared when the disk file is purged.

Normally, a disk process purges a disk file by removing the file entry from the volume directory. This action frees the disk space occupied by the file for later allocation to other disk files but does not physically alter the data pages allocated on disk for the file. Those data pages are not physically altered until they are allocated to another disk file and are subsequently overwritten with new data.

When a disk file that has the CLEARONPURGE attribute set ON is purged, the disk process not only removes the file entry from the volume directory, it also writes null characters to all the data pages allocated to the purged file.

`CLEARONPURGE ON`

The CLEARONPURGE attribute is set to ON for all disk files in *filename-list* of subsequent ADD DISKFILE commands.

`CLEARONPURGE OFF`

the CLEARONPURGE attribute is set to OFF for all disk files in *filename-list* of subsequent ADD DISKFILE commands.

`PERSISTENT {ON|OFF}`

specifies whether the authorization record for a disk file is to be retained if the disk file is purged.

When PERSISTENT is ON, the authorization record for the disk file is retained if the file is purged. If you purge a file with PERSISTENT ON and later create a file with the same name, that file assumes the authorization record associated with the old file.

When PERSISTENT is OFF, the authorization record for the disk file is deleted if the file is purged.

`PERSISTENT ON`

indicates that the PERSISTENT attribute is set to ON for all disk files in *`filename-list`* for subsequent ADD DISKFILE commands.

`PERSISTENT OFF`

indicates that the PERSISTENT attribute is set to OFF for all disk files in *`filename-list`* for subsequent ADD DISKFILE commands.

`OBJECT-TEXT-DESCRIPTION "[any-text]"`

allows you to store printable characters as comments. These comments are associated with the objects and are used to manage the object authorization record.

The text description field can accommodate 255 bytes of text data.

Note. The text specified in the text description field overwrites existing data, if any. Also, when LIKE clause is used with SET DISKFILE command, the OBJECT-TEXT-DESCRIPTION field is not copied with other object authorization record attributes.

The OBJECT-TEXT-DESCRIPTION attribute is supported only on systems running J06.05 and later J-series RVUs, H06.16 and later H-series RVUs, and G06.32 and later G-series RVUs.

`AUDIT-ACCESS-PASS [`*`audit-spec`*`]`

establishes an *`audit-spec`* for successful attempts to access a disk file. This *`audit-spec`* specifies the conditions under which an audit record is written to the audit file when a disk file is successfully accessed. (A disk file is accessed when it is opened, renamed, purged, or, for a program object file, when it is run.)

The form of *`audit-spec`* is:

```
{ ALL | LOCAL | REMOTE | NONE }
```

`ALL`

All successful access attempts are audited.

`LOCAL`

Only successful access attempts made by local users are audited.

`REMOTE`

Only successful access attempts made by remote users are audited.

`NONE`

No successful access attempts are audited.

Omitting *`audit-spec`* specifies NONE.

`AUDIT-ACCESS-FAIL [`*`audit-spec`*`]`

establishes an *`audit-spec`* for unsuccessful attempts to access a disk file. This *`audit-spec`* specifies the conditions under which an audit record is written to the audit file when an attempt to access a disk file fails. (A disk file is accessed when it is opened, renamed, purged, or, for a program object file, when it is run.)

The form of *`audit-spec`* is:

```
{ ALL | LOCAL | REMOTE | NONE }
```

`ALL`

All unsuccessful access attempts are audited.

`LOCAL`

Only unsuccessful access attempts made by local users are audited.

`REMOTE`

Only unsuccessful access attempts made by remote users are audited.

`NONE`

No unsuccessful access attempts are audited.

Omitting *`audit-spec`* specifies NONE.

`AUDIT-MANAGE-PASS [`*`audit-spec`*`]`

establishes an *`audit-spec`* for successful attempts to manage a disk-file-authorization record. This *`audit-spec`* specifies the conditions under which an audit record is written to the audit file when the disk file authorization record is managed.

The form of *`audit-spec`* is:

```
{ ALL | LOCAL | REMOTE | NONE }
```

`ALL`

All successful management attempts are audited.

`LOCAL`

Only successful management attempts made by local users are audited.

`REMOTE`

Only successful management attempts made by remote users are audited.

`NONE`

No successful management attempts are audited.

Omitting *`audit-spec`* specifies NONE.

`AUDIT-MANAGE-FAIL [`*`audit-spec`*`]`

establishes an *`audit-spec`* for unsuccessful attempts to manage a disk file authorization record. This *`audit-spec`* specifies the conditions under which an audit record is written to the audit file when an attempt to manage the disk file authorization record fails.

The form of *`audit-spec`* is:

```
{ ALL | LOCAL | REMOTE | NONE }
```

`ALL`

All unsuccessful management attempts are audited.

`LOCAL`

Only unsuccessful management attempts made by local users are audited.

`REMOTE`

Only unsuccessful management attempts made by remote users are audited.

`NONE`

No unsuccessful management attempts are audited.

Omitting *`audit-spec`* specifies NONE.

`WARNING-MODE { ON | OFF }`

defines whether warning mode is enabled for the specified disk file. The value is required. For more information on warning mode, see the *Safeguard Administrator's Manual*.

ON enables warning mode for the specified disk file. The initial value is OFF, which disables warning mode for the specified disk file.

`TRUST {ME|SHARED|OFF}`

establishes the current default setting of the TRUST attribute for a program object file. This attribute is valid only on systems running H-series RVUs. Only the super ID can set this attribute.

`ME`

specifies that the program can be trusted to not access the buffers private to the process before I/O completion.

`SHARED`

specifies that the program can be trusted to not access the buffers that are private to the process or are shared with another process that also has TRUST SHARED set, before I/O completion.

`OFF`

specifies that the program is not to be trusted.

`PRIV-LOGON { ON | OFF }`

establishes whether the program file (object disk file) can request additional logon related sensitive features. The conditions can be ON or OFF. The default is OFF.

`ON`

a process created from this program file can request a logon without specifying a password.

A process originated from a program file calling USER_AUTHENTICATE_ with a 2 and 15 bit set to ON, the requesting user for authentication need not give a password. Even with wrong password the user will be able to logon successfully as bit 2 and 15 in the options field. In case of only bit 2 set to 1 and bit 15 as 0; no fail delay will take place. That is, no failure delay will be imposed even after three attempts with wrong password. The authentication will not be successful but there will be no delay imposed.

Also establishes whether the program file (object disk file) can request a delay to be imposed for failed logon attempts. The conditions can be ON or OFF. The default is OFF.

`ON`

a process created from this program file is not subjected to logon failure delays.

PRIV-LOGON may also be used in the WHERE expression of a command to restrict scope of that command to files with PRIV-LOGON ON.

Examples

1. In this example, user ID 33,6 sets the default disk file attributes:

```
=ASSUME DISKFILE
=SET ACCESS 33,1 (r,e); 33,2 (r,w); 33,5 (r,w); &
=33,6 (r,w,e,p,o)
=SET AUDIT-ACCESS-PASS local , AUDIT-MANAGE-PASS all
=SET AUDIT-MANAGE-FAIL local
```

User ID 33,6 requests a SHOW report:

```
=SHOW
```

A SHOW report displays:

```
TYPE          OWNER    WARNING-MODE
 DISKFILE     33,6        OFF

  OBJECT-TEXT-DESCRIPTION =

  AUDIT-ACCESS-PASS = LOCAL       AUDIT-MANAGE-PASS = ALL
  AUDIT-ACCESS-FAIL = NONE        AUDIT-MANAGE-FAIL = LOCAL

  AUDIT-PRIV-LOGON = OFF

  LICENSE = OFF  PROGID = OFF  CLEARONPURGE = OFF  PERSISTENT = OFF
                               PRIV-LOGON = OFF

         033,001     R,  E
         033,002     R,W
         033,005     R,W
         033,006     R,W,E,P,  O
```

Note. The attributes, AUDIT-PRIV-LOGON and PRIV-LOGON, are supported only on systems running H06.11 and later H-series RVUs and G06.32 and later G-series RVUs.

User 33,6 adds a record for MYFILE:

```
=ADD myfile
```

2. Later, user ID 33,6 (who wants to secure a new disk file named COPY) enters a SET LIKE command, which sets the default disk file attributes to match the record for MYFILE. User ID 33,6 then specifies other attribute changes when the disk file is added:

```
=SET DISKFILE LIKE myfile
=SHOW DISKFILE
=ADD DISKFILE copy, OWNER 33,5, ACCESS 33,5 (e,p); &
=33,6 - (w,p,o)
```

The OWNER attribute in the ADD DISKFILE command gives the file COPY a different owner from that of MYFILE. Also, the ACCESS specifications give the owner of COPY full access to the file and remove write and purge access from user 33,6. The INFO DISKFILE command verifies this:

```
=INFO DISKFILE copy, DETAIL
```

```
                LAST-MODIFIED     OWNER    STATUS    WARNING-MODE
$DATA.JOAN
 COPY           11OCT94, 7:32      33,5    THAWED       OFF

         033,001     R,  E
         033,002     R,W
         033,005     R,W,E,P,
         033,006     R,  E

  OBJECT-TEXT-DESCRIPTION =

  AUDIT-ACCESS-PASS = LOCAL     AUDIT-MANAGE-PASS = ALL
  AUDIT-ACCESS-FAIL = NONE      AUDIT-MANAGE-FAIL = NONE

  AUDIT-PRIV-LOGON = OFF

  LICENSE = OFF  PROGID = OFF  CLEARONPURGE = OFF  PERSISTENT = OFF
                                  PRIV-LOGON = OFF
```

Note. The attributes, AUDIT-PRIV-LOGON and PRIV-LOGON, are supported only on systems running H06.11 and later H-series RVUs and G06.32 and later G-series RVUs.

SET DISKFILE-PATTERN Command

SET DISKFILE-PATTERN establishes default values for attributes. These values are used whenever the ADD command does not explicitly state the value of the attribute.

To display the current default diskfile-pattern attribute values, use the SHOW DISKFILE-PATTERN command.

```
SET DISKFILE-PATTERN [ , ]
[ LIKE pattern-spec | pattern-attribute ]
[ , pattern-attribute ] ...
```

`LIKE` *`pattern-spec`*

adopts the existing attribute values of *`pattern-spec`* as the *`pattern-attribute`* values to be used for the authorization record or records being added.

`pattern-spec`

are the characters that define the pattern that describe a set of objects. The PATTERN-SPEC for a diskfile pattern is a fully qualified diskfile name that contains at least one wildcard in either the subvolume or file name component that includes the following components:

- A volume name, which will include only valid volume characters; that is, wildcard characters are not part of the pattern, and if present, imply a one-dimensional search. No wildcards are allowed in the volume when used for a LIKE operation.

- A subvolume name, which might include wildcard characters and valid subvolume characters.
- A file name, which might include wildcard characters and valid file name characters.

pattern-attribute

defines a pattern attribute value for the diskfile-pattern authorization record or records being added. The pattern attributes are:

```
OWNER [owner-id]
ACCESS access-spec [ ; access-spec ] ...
AUDIT-ACCESS-PASS [audit-spec]
AUDIT-ACCESS-FAIL [audit-spec]
AUDIT-MANAGE-PASS [audit-spec]
AUDIT-MANAGE-FAIL [audit-spec]
WARNING-MODE {ON|OFF}
```

OWNER [*owner-id*]

specifies the new owner of the diskfile pattern. *owner-id* can be either of:

```
[\*.]group-name.member-name
[\*.]group-num , member-num
```

If you omit *owner-id*, *owner-id* is set to your user ID.

ACCESS *access-spec* [; *access-spec*] ...

changes the ACL for *filename-list* by adding or deleting ACL entries or by changing the authority list of a current ACL entry.

An ACL contains as many as 50 entries that grant or deny access authorities to users and user groups.

access-spec has the form:

```
user-list  [-] [DENY] authority-list

group-list [-] [DENY] authority-list
```

user-list

specifies users who are granted (or denied) the access authorities specified with these *authority-list*. *user-list* can be either of:

```
  net-user-spec

( net-user-spec [ , net-user-spec ] ... )
```

net-user-spec can be any of:

```
[\node-spec.]adm-group-name.user-name
[\node-spec.]adm-group-num , user-num
[\node-spec.]adm-group-name.*
[\node-spec.]adm-group-num , *
```

`[\`*`node-spec`*`.]*.*`
`[\`*`node-spec`*`.]*,*`

-

(minus-sign) operates on the existing ACL entries. The minus-sign form of *`access-spec`* modifies the current default ACL. The *`authority`* entries are removed from the default ACL entries for the users specified with *`user-list`*.

`group-list`

can take either of these forms:

`net-group-spec`

`(` *`net-group-spec`* `[ ,` *`net-user-spec`* `] ... )`

`net-group-spec`

can take any of these forms:

`GROUP [NAME][\`*`node-spec`*`.]` *`group-nam`*`e`

`GROUP NUMBER [\`*`node-spec`*`.]`

`node-spec`

has the form:

`*` `|` *`node-name`* `|` *`node-number`*

`node-name`

specifies the system name.

`node-number`

specifies the Expand node number.

`adm-group-name`

specifies the name of the administrative group.

`admin-group-name`

specifies the group number of an administrative group.

`group-name`

specifies the name of any group.

`group-num`

specifies the group number of any group.

`-`

(minus-sign) operates on the existing ACL entries. The minus-sign form of *`access-spec`* modifies the current default ACL. The *`authority`* entries are removed from the default ACL entries for the users specified with *`user-list`*.

Note. Specifying ACCESS *`access-spec`* through the ADD command does not override the current default ACL (established through the SET command). Instead, any ACL entries specified with the ADD command are added to the current default ACL, and the entire ACL is defined for the disk file whose authorization record is being added.

`DENY`

denies the users or groups specified by *`user-list`* the access authorities specified by *`authority-list`*.

`authority-list`

specifies the access authorities to be granted (or denied) to *`user-list`*. *`authority-list`* can be any one of:

```
  authority
( authority [ , authority ] ... )
  *
```

`authority`

is any one of:

```
R[EAD]
W[RITE]
E[XECUTE]
P[URGE]
C[REATE]
O[WNER]
```

`*`

(asterisk) specifies all the disk-file access authorities (R, W, E, P, C, and O).

`AUDIT-ACCESS-PASS [`*`audit-spec`*`]`

changes the *`audit-spec`* for successful attempts to access the diskfile pattern. The form of *`audit-spec`* is:

```
{ ALL | LOCAL | REMOTE | NONE }
```

For a description of *`audit-spec`*, see the SET DISKFILE Command on page 8-57. Omitting *`audit-spec`* specifies NONE.

`AUDIT-ACCESS-FAIL [`*`audit-spec`*`]`

changes the *audit-spec* for unsuccessful attempts to access the diskfile pattern. The form of *audit-spec* is:

```
{ ALL | LOCAL | REMOTE | NONE }
```

For a description of *audit-spec*, see the SET DISKFILE Command on page 8-57. Omitting *audit-spec* specifies NONE.

`AUDIT-MANAGE-PASS [`*`audit-spec`*`]`

changes the *audit-spec* for successful attempts to manage (change or read) a diskfile-pattern authorization record. The form of *audit-spec* is:

```
{ ALL | LOCAL | REMOTE | NONE }
```

For a description of *audit-spec*, see the SET DISKFILE Command on page 8-57. Omitting *audit-spec* specifies NONE.

`AUDIT-MANAGE-FAIL [`*`audit-spec`*`]`

changes the *audit-spec* for unsuccessful attempts to manage (change or read) a diskfile-pattern authorization record. The form of *audit-spec* is:

```
{ ALL | LOCAL | REMOTE | NONE }
```

For a description of *audit-spec*, see the SET DISKFILE Command on page 8-57. Omitting *audit-spec* specifies NONE.

`WARNING-MODE { ON | OFF }`

defines whether warning mode is enabled for the specified diskfile pattern. The value is required. For more information on warning mode, see the *Safeguard Administrator's Manual*.

ON enables warning mode for the specified diskfile pattern. The initial value is OFF, which disables warning mode for the specified diskfile pattern.

Example

1. To set the default owner to be PROD.DBA:

```
SET DISKFILE-PATTERN OWNER PROD.DBA
```

SHOW DISKFILE Command

SHOW DISKFILE displays the current default values for the disk-file attribute.

When you add a disk file authorization record, the current default disk file attributes are used for any attributes you do not specify in the ADD DISKFILE command. (To set the default disk file attributes to specific values, use the SET DISKFILE command.)

```
SHOW [ / OUT listfile / ] DISKFILE
```

OUT *listfile*

directs the SHOW DISKFILE report to *listfile.* After it executes the SHOW command, SAFECOM redirects its output to the current OUT file.

For *listfile*, specify any file name. SAFECOM opens *listfile* and appends the SHOW DISKFILE report to the file. If *listfile* does not exist, SAFECOM creates an EDIT-format file and then writes the SHOW DISKFILE report to that file.

DISKFILE

specifies DISKFILE as the object type of the SHOW command. Omit it if DISKFILE is the assumed object type. (For more information on assumed object types, see the ASSUME Command on page 4-3.)

SHOW DISKFILE Report Format

The SHOW DISKFILE command displays the disk-file attributes and their current default values in the format shown in Figure 8-3.

Figure 8-3. SHOW DISKFILE Report Format

```
TYPE            OWNER     WARNING-MODE
 DISKFILE       gn,un       {ON|OFF}

  OBJECT-TEXT-DESCRIPTION =

  AUDIT-ACCESS-PASS = a-spec       AUDIT-MANAGE-PASS = a-spec
  AUDIT-ACCESS-FAIL = a-spec       AUDIT-MANAGE-FAIL = a-spec

  AUDIT-PRIV-LOGON ( ON | OFF }

  LICENSE={ON|OFF} PROGID={ON|OFF} CLEARONPURGE={ON|OFF} PERSISTENT={ON|OFF}
  TRUST  ={ME|SHARED|OFF}          PRIV-LOGON { ON | OFF }

     user-spec [DENY] authority-list
     user-spec [DENY] authority-list
          .        .         .
          .        .         .
  [ NO ACCESS CONTROL LIST DEFINED! ]
```

The SHOW DISKFILE report displays these disk file attributes and values:

OWNER *gn, un*

is the user ID (group number and member number) of the user who will own this disk-file authorization record if a file with these attribute values is added to Safeguard protection.

WARNING-MODE
{ON|OFF}

is the current warning-mode state of this disk file. ON indicates that the protection record is in warning mode. The initial value is OFF, which indicates that warning mode is disabled for this disk file.

```
AUDIT-ACCESS-PASS = a-spec    AUDIT-MANAGE-PASS = a-spec
AUDIT-ACCESS-FAIL = a-spec    AUDIT-MANAGE-FAIL = a-spec
```

indicate the conditions under which the Safeguard software audits attempts to access this file or to change or read its authorization record. These four fields are described under the SET DISKFILE Command on page 8-57.

`LICENSE = { ON|OFF }`

indicates whether the LICENSE attribute is set on. For more information, see the INFO DISKFILE Command on page 8-43.

`PROGID = { ON|OFF }`

indicates whether the PROGID attribute is set on. For more information, see the INFO DISKFILE Command on page 8-43.

`CLEARONPURGE = { ON|OFF }`

indicates whether the CLEARONPURGE attribute is set on. For more information, see the INFO DISKFILE Command on page 8-43.

`PERSISTENT = { ON|OFF }`

indicates whether the PERSISTENT attribute is set on. For more information, see the INFO DISKFILE Command on page 8-43.

`TRUST = { ME|SHARED|OFF }`

is the current default setting of the TRUST attribute. For more information, see the INFO DISKFILE Command on page 8-43. This attribute appears only on systems running H-series RVUs.

`user-spec` `[DENY]` *`authority-list`*

is a current default ACL entry for disk files. For more information, see the INFO DISKFILE Command on page 8-43.

`[ NO ACCESS CONTROL LIST DEFINED! ]`

indicates that no default ACL is defined. Use SET DISKFILE...ACCESS to define default ACL entries, or use ADD DISKFILE... ACCESS to define ACL entries when you create an authorization record for a disk file.

△ **Caution.** If you do not specify an ACL for a disk file, only the local super ID can access the file. However, the owner of the file can alter the authorization record to authorize access.

Examples

User 33,3 owns the disk file $DATA.MONEY.BUSNS. This user enters commands to set up default attribute values before adding a Safeguard authorization record for the file:

```
=ASSUME DISKFILE
=SET ACCESS 33,3 (r,w,e,p) ; 33,255 (r,w,e,p)
=SET ACCESS 33,13 DENY (r,w,e,p) ; 33,16 DENY (r,w)
=SET AUDIT-ACCESS-PASS all, AUDIT-MANAGE-PASS local
=SHOW
```

```
TYPE              OWNER     WARNING-MODE
 DISKFILE         33,3          OFF

  OBJECT-TEXT-DESCRIPTION =

  AUDIT-ACCESS-PASS = ALL         AUDIT-MANAGE-PASS = LOCAL
  AUDIT-ACCESS-FAIL = NONE        AUDIT-MANAGE-FAIL = NONE

  AUDIT-PRIV-LOGON = OFF

  LICENSE = OFF  PROGID = OFF  CLEARONPURGE = OFF  PERSISTENT = OFF
  TRUST  = OFF                 PRIV-LOGON = OFF

         033,003      R,W,E,P
         033,013 DENY R,W,E,P
         033,016 DENY R,W
         033,255      R,W,E,P
```

Note. The attributes, AUDIT-PRIV-LOGON and PRIV-LOGON, are supported only on systems running H06.11 and later H-series RVUs and G06.32 and later G-series RVUs.

Now user 33,3 adds an authorization record for this file:

```
=ADD $data.money.busns
```

SHOW DISKFILE-PATTERN Command

The SHOW DISKFILE-PATTERN command displays the current default values for the attributes associated with the object type.

```
SHOW [ / OUT listfile / ] DISKFILE-PATTERN
```

OUT *listfile*

directs the SHOW DISKFILE-PATTERN report to *listfile*. After executing the SHOW command, SAFECOM redirects its output to the current OUT file.

For *listfile*, specify any file name. SAFECOM opens *listfile* and appends the SHOW report to the file. If *listfile* does not exist, SAFECOM creates an EDIT file by that name and writes the SHOW report to that file.

`DISKFILE-PATTERN`

specifies DISKFILE-PATTERN as the object type for the SHOW command. Omit it if DISKFILE-PATTERN is the assumed object type. (For more information on assumed object types, see the ASSUME Command on page 4-3.)

Example

1. To show the current default values for the diskfile pattern:

```
SHOW DISKFILE-PATTERN
```

Output appears:

```
TYPE                      OWNER                 WARNING-MODE
 DISKFILE-PATTERN         20,33                      OFF

  AUDIT-ACCESS-PASS = NONE       AUDIT-MANAGE-PASS = NONE
  AUDIT-ACCESS-FAIL = NONE       AUDIT-MANAGE-FAIL = NONE

             \*.*,*           R,W,E,P,C,O
```

THAW DISKFILE Command

THAW DISKFILE restores the ACL for a frozen disk file. After a frozen disk file is thawed, any users who were granted access authority by the file ACL can once again access the file.

An owner of a disk file, the primary owner's group manager, and the super ID can thaw a frozen disk file.

THAW DISKFILE has no effect on a disk file that is not frozen.

```
THAW DISKFILE filename-list [ [ , ] WHERE option-list ]
```

`DISKFILE`

specifies DISKFILE as the object type of the THAW command. Omit it if DISKFILE is the assumed object type. (For more information on assumed object types, see the ASSUME Command on page 4-3.)

`filename-list`

specifies one or more disk files for which access is to be thawed. *`filename-list`* can be either:

```
  disk-file-name
( disk-file-name [ , disk-file-name ] ... )
```

`disk-file-name`

can be any disk file name. A name can contain wild-card characters.

`WHERE` *`option-list`*

specifies that only disk files in *`filename-list`* that have LICENSE, PROGID, WARNING-MODE, TRUST ME, or TRUST SHARED set are to be thawed.

`option-list` `has the form:`

`[ ( ]` *`option`* `[ OR` *`option`* `] [ ) ]`

`option`

can be one of:

```
PROGID
LICENSE
WARNING-MODE
TRUSTME (H-series only)
TRUSTSHARED (H-series only)
```

Examples

The file $DATA.MONEY.BUSNS is frozen. The file owner can enter THAW DISKFILE to restore the file’s ACL:

```
=THAW DISKFILE $data.money.busns
```

THAW DISKFILE-PATTERN Command

THAW DISKFILE-PATTERN restores the ACL for a frozen diskfile pattern. After a frozen diskfile pattern is thawed, any users who were granted access authority by the file ACL can once again access the pattern.

An owner of a diskfile pattern, the primary owner’s group manager, and the super ID can thaw a frozen diskfile pattern.

THAW DISKFILE-Pattern has no effect on a disk file that is not frozen.

```
THAW DISKFILE-PATTERN pattern-spec-list
[ , ] [ ALL ][ WHERE option-list ]
```

`pattern-spec-list`

is the same as the corresponding non-pattern object types. That is, a PATTERN-SPEC-LIST is a comma-separated list of one or more PATTERN-SPEC attributes. (*`pattern-spec`* [, *`pattern-spec`*] . . .)

`pattern-spec`

are the characters that define the pattern that describe a set of objects. The PATTERN-SPEC for a diskfile pattern is a fully qualified diskfile name that contains at least one wildcard in either the subvolume or file name component that includes the following components:

- A volume name, which will include only valid volume characters; that is, wildcard characters are not part of the pattern, and if present, imply a one-dimensional search.
- A subvolume name, which might include wildcard characters and valid subvolume characters.
- A file name, which might include wildcard characters and valid file name characters.

`WHERE` *`option-list`*

specifies that only disk files in *`filename-list`* that have WARNING-MODE set are to be altered.

`option-list` has the form:

`[ ( ] ` *`option`* ` [ OR ` *`option`* ` ] [ ) ]`

`option`

can be:

`WARNING-MODE`

`ALL`

instructs Safeguard to use all the wildcard characters as a part of the search string, not as part of the pattern.

Example

1. To thaw all diskfile patterns that have a volume name ending in the letter P:

```
THAW DISKFILE-PATTERN $*P.*.*, ALL
```

SAFECOM Saved Diskfile Pattern Commands

The SAFECOM saved-diskfile-pattern commands enable you to create and manage diskfile-pattern protection records with volume-level wild card.

This section includes the individual syntax descriptions for the SAFECOM saved disk-file security commands.

ADD SAVED-DISKFILE-PATTERN Command

ADD SAVED-DISKFILE-PATTERN creates a SAVED-DISKFILE-PATTERN record.

You can specify values for the disk-file attributes in the ADD SAVED-DISKFILE-PATTERN command. The default values are used for any attributes not specified in the ADD SAVED-DISKFILE-PATTERN command.

```
ADD SAVED-DISKFILE-PATTERN pattern-spec-list [ , ]
[ LIKE pattern-spec | pattern-attribute ]
[ , pattern-attribute ] ...
```

pattern-spec-list

is the same as the corresponding non-pattern object types. That is, a PATTERN-SPEC-LIST is a comma-separated list of one or more PATTERN-SPEC attributes. (*pattern-spec [, pattern-spec]...*).

LIKE *pattern-spec*

adopts the existing attribute values of *pattern-spec* as the *pattern-attribute* values to be used for the records being added.

pattern-spec

are the characters that define the pattern that describe a set of objects. The PATTERN-SPEC for a saved-diskfile-pattern is a fully qualified diskfile name that contains at least one wildcard in either the subvolume or file name component that includes the following components:

- A volume name, which might include wildcard characters and valid volume characters.
- A subvolume name, which might include wildcard characters and valid subvolume characters.
- A file name, which might include wildcard characters and valid file name characters.

pattern-attribute

defines a pattern attribute value for the diskfile-pattern authorization record or records being added. The pattern attributes are:

```
OWNER [owner-id]
ACCESS access-spec [ ; access-spec ] ...
AUDIT-ACCESS-PASS [audit-spec]
AUDIT-ACCESS-FAIL [audit-spec]
AUDIT-MANAGE-PASS [audit-spec]
AUDIT-MANAGE-FAIL [audit-spec]
WARNING-MODE {ON|OFF}
```

OWNER [*owner-id*]

specifies the new owner of the diskfile pattern. *owner-id* can be either of:

```
[\*.]group-name.member-name
[\*.]group-num , member-num
```

If you omit *owner-id*, *owner-id* is set to your user ID.

ACCESS *access-spec* [; *access-spec*] ...

changes the ACL for *filename-list* by adding or deleting ACL entries or by changing the authority list of a current ACL entry.

An ACL contains as many as 50 entries that grant or deny access authorities to users and user groups.

access-spec has the form:

```
user-list  [-] [DENY] authority-list

group-list [-] [DENY] authority-list
```

user-list

specifies users who are granted (or denied) the access authorities specified with the following *authority-list*. *user-list* can be either of the following:

```
  net-user-spec

( net-user-spec [ , net-user-spec ] ... )
```

net-user-spec can be any of:

```
[\node-spec.]adm-group-name.user-name
[\node-spec.]adm-group-num , user-num
[\node-spec.]adm-group-name.*
[\node-spec.]adm-group-num , *
[\node-spec.]*.*
[\node-spec.]*,*
```

-

(minus-sign) operates on the existing ACL entries. The minus-sign form of *access-spec* modifies the current default ACL. The *authority* entries are removed from the default ACL entries for the users specified with *user-list*.

group-list

can take either of these forms:

```
  net-group-spec

( net-group-spec [ , net-user-spec ] ... )
```

net-group-spec

can take either of the following forms:

GROUP [NAME][\node-spec.] *group-name*e

GROUP NUMBER [\node-spec.]

node-spec

has the form:

* | *node-name* | *node-number*

node-name

specifies the system name.

node-number

specifies the Expand node number.

adm-group-name

specifies the name of the administrative group.

admin-group-name

specifies the group number of an administrative group.

group-name

specifies the name of any group.

group-num

specifies the group number of any group.

-

(minus-sign) operates on the existing ACL entries. The minus-sign form of *access-spec* modifies the current default ACL. The *authority* entries are removed from the default ACL entries for the users specified with *user-list*.

Note. Specifying ACCESS *access-spec* through the ADD command does not override the current default ACL (established through the SET command). Instead, any ACL entries specified with the ADD command are added to the current default ACL, and the entire ACL is defined for the disk file whose authorization record is being added.

DENY

denies the users or groups specified by *user-list* the access authorities specified by *authority-list*.

authority-list

specifies the access authorities to be granted (or denied) to *user-list*. *authority-list* can be any one of:

```
  authority

( authority [ , authority ] ... )

  *
```

authority

is any one of:

```
R[EAD]
W[RITE]
E[XECUTE]
P[URGE]
C[REATE]
O[WNER]
```

*

(asterisk) specifies all the disk-file access authorities (R, W, E, P, C, and O).

AUDIT-ACCESS-PASS [*audit-spec*]

changes the *audit-spec* for successful attempts to access the diskfile pattern. The form of *audit-spec* is:

```
{ ALL | LOCAL | REMOTE | NONE }
```

For a description of *audit-spec*, see the SET DISKFILE Command on page 8-57. Omitting *audit-spec* specifies NONE.

AUDIT-ACCESS-FAIL [*audit-spec*]

changes the *audit-spec* for unsuccessful attempts to access the diskfile pattern. The form of *audit-spec* is:

```
{ ALL | LOCAL | REMOTE | NONE }
```

For a description of *audit-spec*, see the SET DISKFILE Command on page 8-57. Omitting *audit-spec* specifies NONE.

AUDIT-MANAGE-PASS [*audit-spec*]

changes the *audit-spec* for successful attempts to manage (change or read) a diskfile-pattern authorization record. The form of *audit-spec* is:

```
{ ALL | LOCAL | REMOTE | NONE }
```

For a description of *audit-spec*, see the SET DISKFILE Command on page 8-57. Omitting *audit-spec* specifies NONE.

`AUDIT-MANAGE-FAIL [`*`audit-spec`*`]`

changes the *`audit-spec`* for unsuccessful attempts to manage (change or read) a diskfile-pattern authorization record. The form of *`audit-spec`* is:

```
{ ALL | LOCAL | REMOTE | NONE }
```

For a description of *`audit-spec`*, see the SET DISKFILE Command on page 8-57. Omitting *`audit-spec`* specifies NONE.

`WARNING-MODE { ON | OFF }`

defines whether the warning mode is enabled for the specified diskfile pattern. The value is required. For information about the warning mode, see the *Safeguard Administrator's Manual*.

ON enables the warning mode for the specified diskfile pattern. The initial value is OFF, which disables the warning mode for the specified diskfile pattern.

Examples

1. To add a saved-diskfile-pattern record that describes all files that reside on $DATA with subvolume names that begin with PROD:

```
ADD SAVED-DISKFILE-PATTERN $DATA.PROD*.*, &
ACCESS PROD.* (R,W)
```

2. To add a saved-diskfile-pattern record for all files in subvolume $A.B:

```
ADD SAVED-DISKFILE-PATTERN $A.B.*, ACCESS *.* (R,W)
```

3. To add a saved-diskfile-pattern record for every disk file named FILE followed by one alphanumeric character that is in a subvolume REPORT on every volume beginning with $DATA:

```
ADD SAVED-DISKFILE-PATTERN $DATA*.REPORT.FILE?
```

ALTER SAVED-DISKFILE-PATTERN Command

ALTER SAVED-DISKFILE-PATTERN changes one or more saved-diskfile-pattern attribute values. The primary owner of a pattern, the primary owner's group manager, the local super ID, and any user with OWNER authority on the ACL can change a pattern authorization record.

```
ALTER SAVED-DISKFILE-PATTERN pattern-spec-list [ , ][ALL]
[ WHERE option-list ] [ , ]
[ LIKE pattern-spec | pattern-attribute ]
[ , pattern-attribute ]...
```

`pattern-spec-list`

is the same as the corresponding non-pattern object types. That is, a PATTERN-SPEC-LIST is a comma-separated list of one or more PATTERN-SPEC attributes. (`pattern-spec [, pattern-spec]...`).

`ALL`

instructs Safeguard to use all the wildcard characters as part of the search string, and not as part of the pattern.

`LIKE pattern-spec`

adopts the existing attribute values of *pattern-spec* as the *pattern-attribute* values to be used for the authorization record or records being added.

pattern-spec

are the characters that define the pattern that describe a set of objects. The PATTERN-SPEC for a saved-diskfile-pattern is a fully qualified diskfile name that contains at least one wildcard in either the subvolume or file name component that includes the following components:

- A volume name, which might include wildcard characters and valid volume characters.
- A subvolume name, which might include wildcard characters and valid subvolume characters.
- A file name, which might include wildcard characters and valid file name characters.

`WHERE option-list`

specifies that only disk files in *filename-list* that have WARNING-MODE set must be altered.

option-list has the form:

```
[ ( ] option [ OR option ] [ ) ]
```

option

can be:

`WARNING-MODE`

pattern-attribute

defines a pattern attribute value for the saved-diskfile-pattern authorization record or records being added. The pattern attributes are:

```
OWNER [owner-id]
ACCESS access-spec [ ; access-spec ] ...
AUDIT-ACCESS-PASS [audit-spec]
```

```
AUDIT-ACCESS-FAIL [audit-spec]
AUDIT-MANAGE-PASS [audit-spec]
AUDIT-MANAGE-FAIL [audit-spec]
WARNING-MODE {ON|OFF}
```

```
OWNER [owner-id]
```

specifies the new owner of the disk file or files. *owner-id* can be either of:

```
[\*.]group-name.member-name
[\*.]group-num , member-num
```

If you omit *owner-id*, *owner-id* is set to your user ID.

```
ACCESS access-spec [ ; access-spec ] ...
```

changes the ACL for *filename-list* by adding or deleting ACL entries or by changing the authority list of a current ACL entry.

An ACL contains as many as 50 entries that grant or deny access authorities to users and user groups.

access-spec has the form:

```
user-list  [-] [DENY] authority-list

group-list [-] [DENY] authority-list
```

user-list

specifies users who are granted (or denied) the access authorities specified with the following *authority-list*. *user-list* can be either of:

```
  net-user-spec

( net-user-spec [ , net-user-spec ] ... )
```

net-user-spec can be any of:

```
[\node-spec.]adm-group-name.user-name
[\node-spec.]adm-group-num , user-num
[\node-spec.]adm-group-name.*
[\node-spec.]adm-group-num , *
[\node-spec.]*.*
[\node-spec.]*,*
```

-

(minus-sign) operates on existing ACL entries. The minus-sign form of *access-spec* modifies the current default ACL. The *authority* entries are removed from the default ACL entries for the users specified with *user-list*.

group-list

can take either of these forms:

net-group-spec

(*net-group-spec* [, *net-user-spec*] ...)

net-group-spec

can take any of these forms:

GROUP [NAME][\node-spec.] *group-nam*e

GROUP NUMBER [\node-spec.]

node-spec

has the form:

* | *node-name* | *node-number*

node-name

specifies the system name.

node-number

specifies the Expand node number.

adm-group-name

specifies the name of the administrative group.

admin-group-name

specifies the group number of an administrative group.

group-name

specifies the name of any group.

group-num

specifies the group number of any group.

-

(minus-sign) operates on existing ACL entries. The minus-sign form of *access-spec* modifies the current default ACL. The *authority* entries

are removed from the default ACL entries for the users specified with *user-list*.

Note. Specifying ACCESS *access-spec* through the ADD command does not override the current default ACL (established through the SET command). Instead, any ACL entries specified with the ADD command are added to the current default ACL, and the entire ACL is defined for the disk file whose authorization record is being added.

DENY

denies the users or groups specified by *user-list* the access authorities specified by *authority-list*.

authority-list

specifies the access authorities to be granted (or denied) to *user-list*. *authority-list* can be any one of:

```
  authority
( authority [ , authority ] ... )
  *
```

authority

is any one of:

```
R[EAD]
W[RITE]
E[XECUTE]
P[URGE]
C[REATE]
O[WNER]
```

*

(asterisk) specifies all the disk-file access authorities (R, W, E, P, C, and O).

AUDIT-ACCESS-PASS [*audit-spec*]

changes the *audit-spec* for successful attempts to access the diskfile pattern. The form of *audit-spec* is:

```
{ ALL | LOCAL | REMOTE | NONE }
```

For a description of *audit-spec*, see the SET DISKFILE Command on page 8-57. Omitting *audit-spec* specifies NONE.

`AUDIT-ACCESS-FAIL [`*`audit-spec`*`]`

changes the *audit-spec* for unsuccessful attempts to access the diskfile pattern. The form of *audit-spec* is:

```
{ ALL | LOCAL | REMOTE | NONE }
```

For a description of *audit-spec*, see the SET DISKFILE Command on page 8-57. Omitting *audit-spec* specifies NONE.

`AUDIT-MANAGE-PASS [`*`audit-spec`*`]`

changes the *audit-spec* for successful attempts to manage (change or read) a diskfile-pattern authorization record. The form of *audit-spec* is:

```
{ ALL | LOCAL | REMOTE | NONE }
```

For a description of *audit-spec*, see the SET DISKFILE Command on page 8-57. Omitting *audit-spec* specifies NONE.

`AUDIT-MANAGE-FAIL [`*`audit-spec`*`]`

changes the *audit-spec* for unsuccessful attempts to manage (change or read) a diskfile-pattern authorization record. The form of *audit-spec* is:

```
{ ALL | LOCAL | REMOTE | NONE }
```

For a description of *audit-spec*, see the SET DISKFILE Command on page 8-57. Omitting *audit-spec* specifies NONE.

`WARNING-MODE { ON | OFF }`

defines whether the warning mode is enabled for the specified diskfile pattern. The value is required. For more information on warning mode, see the *Safeguard Administrator's Manual*.

ON enables warning mode for the specified diskfile pattern. The initial value is OFF, which disables warning mode for the specified diskfile pattern.

Examples

1. To alter a saved-diskfile-pattern record to provide super ID read and write access:

```
ALTER SAVED-DISKFILE-PATTERN $DATA.APLOGS.LOG*, &
ACCESS SUPER.SUPER (R,W)
```

2. To alter all saved-diskfile-pattern records that match $DATA*.APLOGS.LOG*:

```
ALTER SAVED-DISKFILE-PATTERN $DATA*.APLOGS.LOG*, ALL, &
ACCESS SUPER.SUPER (R,W)
```

This command alters the saved-diskfile-pattern records for all matching patterns. For example, if the following patterns exist, they are altered as:

- $DATA01.APLOGS.LOGAPR*
- $DATA1.APLOGS.LOG*
- $DATA123.APLOGS.LOG?
- $DATA.APLOGS.LOG????
- $DATABLE.APLOGS.LOGON?1A

DELETE SAVED-DISKFILE-PATTERN Command

DELETE SAVED-DISKFILE-PATTERN deletes the pattern record for a saved-diskfile-pattern.

The owner of a saved-diskfile-pattern, the primary owner's group manager, the local super ID, and any user with OWNER authority on the ACL can delete a saved-diskfile-pattern authorization record.

```
DELETE SAVED-DISKFILE-PATTERN pattern-spec-list [ , ] [ ALL ]
[ WHERE option-list ] ...
```

pattern-spec-list

is the same as the corresponding non-pattern object types. That is, a PATTERN-SPEC-LIST is a comma-separated list of one or more PATTERN-SPEC attributes. (*pattern-spec* [, *pattern-spec*] . . .).

pattern-spec

are the characters that define the pattern that describe a set of objects. The PATTERN-SPEC for a saved-diskfile-pattern is a fully qualified diskfile name that contains at least one wildcard in either the subvolume or file name component that includes the following components:

- A volume name, which might include wildcard characters and valid volume characters.
- A subvolume name, which might include wildcard characters and valid subvolume characters.
- A file name, which might include wildcard characters and valid file name characters.

WHERE *option-list*

specifies that only disk files in `filename-list` that have WARNING-MODE set must be altered.

option-list has the following form:

[( ] option [OR option] [ )]

option

can be:

WARNING-MODE

`ALL`

instructs Safeguard to use all the wildcard characters as part of the search string, not as part of the pattern.

Examples

1. To delete the saved-diskfile-pattern $ABC.*.*:

```
DELETE SAVED-DISKFILE-PATTERN $ABC.*.*
```

2. To delete all saved-diskfile-pattern protection records that match the search pattern $ABC.*.*:

```
DELETE SAVED-DISKFILE-PATTERN $ABC.*.*, ALL
```

FREEZE SAVED-DISKFILE-PATTERN Command

FREEZE SAVED-DISKFILE-PATTERN temporarily suspends saved-diskfile-pattern protection records being considered by Safeguard to create the corresponding diskfile-pattern protection records during SYNC. While a saved-diskfile-pattern protection record is frozen, only the pattern owner, an owner on the ACL, the primary owner's group manager, and the local super ID can access the pattern. Any other users receive a security violation error (file error 48).

An owner of a pattern, the primary owner's group manager, and the super ID can freeze a saved-diskfile-pattern.

Use the THAW SAVED-DISKFILE-PATTERN command to restore all the access authorities that were in effect before the pattern was frozen.

```
FREEZE SAVED-DISKFILE-PATTERN pattern-spec-list
[ , ] [ ALL ] [WHERE option-list ] ...
```

`pattern-spec-list`

is the same as the corresponding non-pattern object types. That is, a PATTERN-SPEC-LIST is a comma-separated list of one or more PATTERN-SPEC attributes. (*`pattern-spec`* [, *`pattern-spec`*] . . .).

`pattern-spec`

are the characters that define the pattern that describe a set of objects. The PATTERN-SPEC for a saved-diskfile-pattern is a fully qualified diskfile name that contains at least one wildcard in either the subvolume or file name component that includes the following components:

- A volume name, which might include wildcard characters and valid volume characters.
- A subvolume name, which might include wildcard characters and valid subvolume characters.
- A file name, which might include wildcard characters and valid file name characters.

`WHERE` *`option-list`*

specifies that only disk files in `filename-list` that have WARNING-MODE set are to be altered.

`option-list` has the following form:

`[ ( ] option [ OR option ] [ ) ]`

`option`

can be:

WARNING-MODE

`ALL`

instructs Safeguard to use all the wildcard characters as part of the search string, not as part of the pattern.

Examples

To freeze all the saved-diskfile-pattern protection records that specify a subvolume name beginning with the characters TEST, use the following command:

```
FREEZE SAVED-DISKFILE-PATTERN $*.TEST*.*, ALL
```

INFO SAVED-DISKFILE-PATTERN Command

INFO SAVED-DISKFILE-PATTERN displays the current attribute values of the specified saved-diskfile-pattern. The INFO SAVED-DISKFILE-PATTERN command produces the following types of reports: brief and detailed.

Any user can produce an INFO report for any saved-diskfile-pattern protection record.

```
INFO [ / OUT listfile / ] SAVED-DISKFILE-PATTERN pattern-
spec-list
[ , ] [ ALL ] [ WHERE option-list ]
[ , ] [ display-option ] ...
```

`OUT` *`listfile`*

directs the INFO DISKFILE report to *`listfile`*. After executing the INFO command, SAFECOM redirects its output to the current OUT file.

For *listfile*, specify any file name. SAFECOM opens *listfile* and appends the INFO report to the file. If *listfile* does not exist, SAFECOM creates an EDIT file by that name and writes the INFO report to that file.

`SAVED-DISKFILE-PATTERN`

specifies SAVED-DISKFILE-PATTERN as the object type for the INFO command. Omit it if SAVED-DISKFILE-PATTERN is the assumed object type. (For more information on assumed object types, see the ASSUME Command on page 4-3.)

pattern-spec-list

is the same as the corresponding non-pattern object types. That is, a PATTERN-SPEC-LIST is a comma-separated list of one or more PATTERN-SPEC attributes. (*pattern-spec* [, *pattern-spec*] . . .).

`ALL`

instructs Safeguard to use all the wildcard characters as part of the search string, not as part of the pattern.

pattern-spec

are the characters that define the pattern that describe a set of objects. The PATTERN-SPEC for a saved-diskfile-pattern is a fully qualified diskfile name that contains at least one wildcard in either the subvolume or file name component that includes the following components:

- A volume name, which might include wildcard characters and valid volume characters.
- A subvolume name, which might include wildcard characters and valid subvolume characters.
- A file name, which might include wildcard characters and valid file name characters.

display-option

can be one of:

```
DETAIL [ ON | OFF ]
WARNINGS [ ON | OFF ]
WHERE option-list
```

`DETAIL [ ON | OFF ]`

allows certain additional information to be included in this INFO command.

`DETAIL [ ON ]`

adds the *audit-spec* variables defined for the file to the INFO report.

DETAIL [OFF]

inhibits the display of additional information for this command. The default value is DETAIL OFF.

WARNINGS [ON | OFF]

allows the display of warning messages for this command to be inhibited.

WARNINGS [ON]

causes the display of warning messages for this command. The default value is WARNINGS ON.

WARNINGS [OFF]

inhibits the display of warning messages for this command.

WHERE *option-list*

specifies that only disk files in `filename-list` that have WARNING-MODE set must be altered.

option-list has the following form:

[( ] option [OR option] [ )]

option

can be:

WARNING-MODE

Examples

1. To display the saved-diskfile-pattern $DATA.*TEST.* :

```
=INFO SAVED-DISKFILE-PATTERN $DATA.*TEST.*
```

The display appears as:

```
LAST-MODIFIED  OWNER  STATUS  WARNING-MODE
$DATA.*TEST
*              255,255        28SEP04, 5:44 THAWED OFF

\KONA.PROD.CARLY R
\KONA.TEST.JIMMY R,W
GROUP TEST       R,W,E,P,C
GROUP \KONA.TEST R
\*.*.*           R
```

2. To display the saved-diskfile-pattern $DATA.*TEST.*, DETAIL :

```
=INFO SAVED-DISKFILE-PATTERN $DATA.*TEST.*,DETAIL
```

The display appears as:

```
              LAST-MODIFIED   OWNER    STATUS WARNING-MODE
$DATA.*TEST
              * 28SEP04, 5:44 255,255 THAWED      OFF

\KONA.PROD.CARLY R
\KONA.TEST.JIMMY R,W
GROUP TEST       R,W,E,P,C
GROUP \KONA.TEST R
\*.*.*           R

AUDIT-ACCESS-PASS = NONE       AUDIT-MANAGE-PASS = NONE
AUDIT-ACCESS-FAIL = NONE       AUDIT-MANAGE-FAIL = NONE

      CREATION                LAST-MODIFIED

USER NAME SUPER.SUPER           testman
USER TYPE USER (ID 255,255)     ALIAS (ID 164,255)
USER NODE LOCAL                 LOCAL
TIMESTAMP 28SEP2004, 05:28:48.870  28SEP2004, 05:44:22.588
```

3. To display the saved-diskfile-pattern protection records for all volumes starting with “$DATA” with subvolumes starting with “PROD”:

```
INFO SAVED-DISKFILE-PATTERN $DATA*.PROD*.*, ALL
```

4. To display multiple saved-diskfile-pattern protection records that have warning-mode enabled:

```
INFO SAVED-DISKFILE-PATTERN $*.*.*, ALL, WHERE WARNING-MODE
```

RESET SAVED-DISKFILE-PATTERN Command

RESET SAVED-DISKFILE-PATTERN resets the current saved-diskfile -pattern attribute values to its predefined values.

```
RESET SAVED-DISKFILE-PATTERN [ [ , ] pattern-attribute-
keyword ]
[ , pattern-attribute-keyword ] ...
```

pattern-attribute-keyword

sets the current default value of the *pattern-attribute* indicated by *pattern-attribute-keyword* to predefined values, as follows:

```
OWNER              - User ID of the current SAFECOM user
ACCESS             - Null (no access control list)
AUDIT-ACCESS-PASS  - NONE (no auditing)
AUDIT-ACCESS-FAIL  - NONE (no auditing)
AUDIT-MANAGE-PASS  - NONE (no auditing)
AUDIT-MANAGE-FAIL  - NONE (no auditing)
WARNING-MODE       - OFF (warning mode disabled)
```

For a complete description of *pattern-attribute*, see the SET DISKFILE-PATTERN Command on page 8-68.

SET SAVED-DISKFILE-PATTERN Command

SET SAVED-DISKFILE-PATTERN establishes default values for attributes. These values are used whenever the ADD command does not explicitly state the value of the attribute.

To display the current default saved-diskfile-pattern attribute values, use the SHOW SAVED-DISKFILE-PATTERN command.

```
SET SAVED-DISKFILE-PATTERN [ , ]
[ LIKE pattern-spec | pattern-attribute ]
[ , pattern-attribute ] ...
```

LIKE *pattern-spec*

adopts the existing attribute values of *pattern-spec* as the *pattern-attribute* values to be used for the authorization record or records being added.

pattern-spec

are the characters that define the pattern that describe a set of objects. The PATTERN-SPEC for a saved-diskfile-pattern is a fully qualified diskfile name that contains at least one wildcard in either the subvolume or file name component that includes the following components:

- A volume name, which might include wildcard characters and valid volume characters.
- A subvolume name, which might include wildcard characters and valid subvolume characters.
- A file name, which might include wildcard characters and valid file name characters.

pattern-attribute

defines a pattern attribute value for the diskfile-pattern authorization record or records being added. The pattern attributes are:

```
OWNER [owner-id]
ACCESS access-spec [ ; access-spec ] ...
AUDIT-ACCESS-PASS [audit-spec]
AUDIT-ACCESS-FAIL [audit-spec]
AUDIT-MANAGE-PASS [audit-spec]
AUDIT-MANAGE-FAIL [audit-spec]
WARNING-MODE {ON|OFF}
```

OWNER [*owner-id*]

specifies the new owner of the diskfile pattern. *owner-id* can be either of:

```
[\*.]group-name.member-name
[\*.]group-num , member-num
```

If you omit *owner-id*, *owner-id* is set to your user ID.

ACCESS *access-spec* [; *access-spec*] ...

changes the ACL for *filename-list* by adding or deleting ACL entries or by changing the authority list of a current ACL entry.

An ACL contains as many as 50 entries that grant or deny access authorities to users and user groups.

access-spec has the form:

```
user-list  [-] [DENY] authority-list

group-list [-] [DENY] authority-list
```

user-list

specifies users who are granted (or denied) the access authorities specified with these *authority-list*. *user-list* can be either of:

```
  net-user-spec

( net-user-spec [ , net-user-spec ] ... )
```

net-user-spec can be any of:

```
[\node-spec.]adm-group-name.user-name
[\node-spec.]adm-group-num , user-num
[\node-spec.]adm-group-name.*
[\node-spec.]adm-group-num , *
[\node-spec.]*.*
[\node-spec.]*,*
```

-

(minus-sign) operates on existing ACL entries. The minus-sign form of `access-spec` modifies the current default ACL. The `authority` entries are removed from the default ACL entries for the users specified with `user-list`.

group-list

can take either of these forms:

```
net-group-spec
```

```
( net-group-spec [ , net-user-spec ] ... )
```

net-group-spec

can take any of these forms:

```
GROUP [NAME][\node-spec.] group-name
```

```
GROUP NUMBER [\node-spec.]
```

node-spec

has the form:

```
* | node-name | node-number
```

node-name

specifies the system name.

node-number

specifies the Expand node number.

adm-group-name

specifies the name of the administrative group.

admin-group-name

specifies the group number of an administrative group.

group-name

specifies the name of any group.

group-num

specifies the group number of any group.

`-`

(minus-sign) operates on existing ACL entries. The minus-sign form of *`access-spec`* modifies the current default ACL. The *`authority`* entries are removed from the default ACL entries for the users specified with *`user-list`*.

Note. Specifying ACCESS *`access-spec`* through the ADD command does not override the current default ACL (established through the SET command). Instead, any ACL entries specified with the ADD command are added to the current default ACL, and the entire ACL is defined for the disk file whose authorization record is being added.

`DENY`

denies the users or groups specified by *`user-list`* the access authorities specified by *`authority-list`*.

`authority-list`

specifies the access authorities to be granted (or denied) to *`user-list`*. *`authority-list`* can be any one of:

```
  authority
( authority [ , authority ] ... )
  *
```

`authority`

is any one of:

```
R[EAD]
W[RITE]
E[XECUTE]
P[URGE]
C[REATE]
O[WNER]
```

`*`

(asterisk) specifies all the disk-file access authorities (R, W, E, P, C, and O).

`AUDIT-ACCESS-PASS [`*`audit-spec`*`]`

changes the *`audit-spec`* for successful attempts to access the diskfile pattern. The form of *`audit-spec`* is:

```
{ ALL | LOCAL | REMOTE | NONE }
```

For a description of *`audit-spec`*, see the SET DISKFILE Command on page 8-57. Omitting *`audit-spec`* specifies NONE.

`AUDIT-ACCESS-FAIL [`*`audit-spec`*`]`

changes the *audit-spec* for unsuccessful attempts to access the diskfile pattern. The form of *audit-spec* is:

```
{ ALL | LOCAL | REMOTE | NONE }
```

For a description of *audit-spec*, see the SET DISKFILE Command on page 8-57. Omitting *audit-spec* specifies NONE.

`AUDIT-MANAGE-PASS [`*`audit-spec`*`]`

changes the *audit-spec* for successful attempts to manage (change or read) a diskfile-pattern authorization record. The form of *audit-spec* is:

```
{ ALL | LOCAL | REMOTE | NONE }
```

For a description of *audit-spec*, see the SET DISKFILE Command on page 8-57. Omitting *audit-spec* specifies NONE.

`AUDIT-MANAGE-FAIL [`*`audit-spec`*`]`

changes the *audit-spec* for unsuccessful attempts to manage (change or read) a diskfile-pattern authorization record. The form of *audit-spec* is:

```
{ ALL | LOCAL | REMOTE | NONE }
```

For a description of *audit-spec*, see the SET DISKFILE Command on page 8-57. Omitting *audit-spec* specifies NONE.

`WARNING-MODE { ON | OFF }`

defines whether the warning mode is enabled for the specified diskfile pattern. The value is required. For more information on warning mode, see the *Safeguard Administrator's Manual*.

ON enables warning mode for the specified diskfile pattern. The initial value is OFF, which disables warning mode for the specified diskfile pattern.

Example

To set the default owner to be PROD.DBA:

```
=SET SAVED-DISKFILE-PATTERN OWNER PROD.DBA
```

SHOW SAVED-DISKFILE-PATTERN Command

SHOW SAVED-DISKFILE-PATTERN displays the current default values for the attributes associated with the object type.

```
SHOW [ / OUT listfile / ] SAVED-DISKFILE-PATTERN
```

`OUT` *`listfile`*

directs the SHOW SAVED-DISKFILE-PATTERN report to *`listfile`*. After executing the SHOW command, SAFECOM redirects its output to the current OUT file.

For *`listfile`*, specify any file name. SAFECOM opens *`listfile`* and appends the SHOW report to the file. If *`listfile`* does not exist, SAFECOM creates an EDIT file by that name and writes the SHOW report to that file.

`SAVED-DISKFILE-PATTERN`

specifies SAVED-DISKFILE-PATTERN as the object type of the SHOW command. Omit it if SAVED-DISKFILE-PATTERN is the assumed object type. (For more information on assumed object types, see the ASSUME Command on page 4-3.)

Example

To show the current default values for the diskfile pattern:

```
=SHOW SAVED-DISKFILE-PATTERN
```

THAW SAVED-DISKFILE-PATTERN Command

THAW SAVED-DISKFILE-PATTERN restores a frozen saved-diskfile-pattern protection record. After a frozen saved-diskfile-pattern is thawed, Safeguard considers that protection record during the SYNC operation to create the corresponding diskfile-pattern protection record.

An owner of a diskfile-pattern, the primary owner’s group manager, and the super ID can thaw a frozen saved-diskfile-pattern protection record.

```
THAW SAVED-DISKFILE-PATTERN pattern-spec-list
[ , ] [ ALL ][ WHERE option-list ] ...
```

`pattern-spec-list`

is the same as the corresponding non-pattern object types. That is, a PATTERN-SPEC-LIST is a comma-separated list of one or more PATTERN-SPEC attributes. (*`pattern-spec`* [, *`pattern-spec`*] . . .).

`pattern-spec`

are the characters that define the pattern that describe a set of objects. The PATTERN-SPEC for a saved-diskfile-pattern record is a fully qualified diskfile name that contains at least one wildcard in either the subvolume or file name component that includes the following components:

- A volume name, which might include wildcard characters and valid volume characters.

- A subvolume name, which might include wildcard characters and valid subvolume characters.
- A file name, which might include wildcard characters and valid file name characters.

`WHERE` *`option-list`*

specifies that only disk files in `filename-list` that have WARNING-MODE set must be altered.

`option-list` has the following form:

`[ ( ] option [ OR option ] [ ) ]`

`option`

can be:

WARNING-MODE

`ALL`

instructs Safeguard to use all the wildcard characters as part of the search string, not as part of the pattern.

Example

To thaw all saved-diskfile-patterns that have a volume name ending in the letter P, use the following command :

```
THAW SAVED-DISKFILE-PATTERN $*P.*.*, ALL
```

9

Disk Volume and Subvolume Security Commands

SAFECOM volume and subvolume security commands control who can create and access disk files. The disk volume and subvolume commands also specify when the Safeguard software should audit attempts to create or read volume or subvolume authorization records.

By default, only a local super-group user can add a volume authorization record to the Safeguard object database, but any user can add a subvolume authorization record. However, through the use of an access control list (ACL) for the OBJECTTYPEs VOLUME and SUBVOLUME, this behavior can be modified. For more information, see Section 12, OBJECTTYPE Security Commands. After a record is added, all attempts to create or own files on that volume or subvolume are subject to a Safeguard authorization check and, optionally, to Safeguard auditing.

This section describes volume and subvolume ownership and how the Safeguard software authorizes attempts to create or access disk files on protected volumes and subvolumes. It also summarizes the volume and subvolume security commands. Following the command summary, each command is described in detail.

Volume Authorization Record Ownership

A disk volume has no owner until a local super group user places the volume under Safeguard control. By default, only a local super group user can add a disk volume authorization record. (For more information about controlling this class of objects, see VOLUME on page 12-2.) Every Safeguard object access authorization record contains an OWNER attribute. The OWNER attribute contains the user ID of the user who can manage the Safeguard access controls for the disk volume.

However, the user who adds the record can set the OWNER attribute to the user ID of any user (by including an OWNER specification in a SET VOLUME or ADD VOLUME command). Thus the owner of a disk volume might not be a local super group user. The owner of a protected disk volume authorization record, the owner’s group manager, and the super ID can transfer ownership to another user by changing the OWNER attribute through the ALTER VOLUME command.

In addition, the initial owner can add owners to an ACL. Additional ownership is defined by the OWNER authority code for ACL entries and is an independent extension of the initial owner. Additional owners can do anything that the initial owner is permitted to do. They are equal, in every way, to the initial owner. For example, they can modify the Safeguard authorization records for any volume they own, and they can access any volume they own when that volume has been frozen.

The OWNER authority can be used to deny explicitly a local super ID any of the authorities implicitly granted to the super ID, including OWNER. The OWNER authority can always be specified for all volumes protected by the Safeguard software. With an ACL in effect, the OWNER authority is always included whenever the * (asterisk) authority code is used. It can also be abbreviated as O.

With the Safeguard software, the owner of a volume can also be defined as a network user. A network user who owns a protected volume can use the Safeguard software from a remote node to control access to that volume (provided the user has remote passwords set up between the two systems).

For more information about OWNER, see VOLUME on page 12-2.

Subvolume Authorization Record Ownership

The rules governing subvolume authorization record ownership are nearly identical to those governing volume ownership. The only exception is that although only a local super group user can add an authorization record for a volume (by default), any local user can add an authorization record for a subvolume. For more information about controlling this class of objects, see OBJECTTYPE SUBVOLUME in Section 12.

Volume and Subvolume Access Authorities

ACLs for disk volumes and subvolumes can grant the following six access authorities:

CREATE	Create a disk file on a volume or subvolume
OWNER	Change the authorization records
READ	Read disk files within the volume or subvolume
WRITE	Write disk files within the volume or subvolume
EXECUTE	Execute disk files within the volume or subvolume
PURGE	Purge disk files within the volume or subvolume

Note. READ, WRITE, EXECUTE, and PURGE access for a volume or subvolume are consulted only when the Safeguard global configuration attribute CHECK-VOLUME or CHECK-SUBVOLUME is turned on. If the option is turned off, only CREATE and OWNER are applicable. For more information about the global configuration attributes, see Section 16, Safeguard Subsystem Commands.

Volume and Subvolume Access Authorization

When a user attempts to create or access a disk file, the Safeguard software performs a series of authorization checks to determine whether to allow the user to create or access the file on the specified volume or subvolume. The following paragraphs describe the authorization checking performed by the Safeguard software.

For any attempt to create or access a disk file, the Safeguard software checks the requester’s authority as follows:

1. It determines whether an authorization record exists for the volume on which the file is to be created.
2. If a volume authorization record exists, it checks the ACL to determine whether the user has the authority to create or access a file on that volume.
3. If the volume ACL does not grant the user the authority, the user’s request is rejected with a security violation (file error 48).

This behavior can be modified depending on the settings of the following Safeguard configuration options: CHECK-VOLUME, CHECK-SUBVOLUME, CHECK-FILENAME, CHECK-DISKFILE-PATTERN, DIRECTION-DISKFILE, and COMBINATION-DISKFILE.

You can use patterns to specify protection records. For more information on diskfile patterns, see the *Safeguard User’s Guide*.

Section 16, Safeguard Subsystem Commands, and Appendix B, Disk-File Access Rules, describe these configuration options and their effects on creating a disk file.

If the volume ACL grants the user the authority to create a disk file, the Safeguard software then determines whether an authorization record exists for the subvolume on which the disk file is to be created. When an authorization record exists for the subvolume, the Safeguard software checks whether the subvolume ACL grants the user the authority to create a disk file. If the subvolume ACL grants the user the authority to create a disk file, the user’s file-creation request succeeds. However, when the user lacks the authority to create a disk file on the subvolume, the file-creation request is rejected with a security violation (file error 48).

If no authorization record exists for the volume, a user’s file-creation request is rejected only if both an authorization record for the subvolume exists and the subvolume ACL does not grant the user CREATE authority. If no authorization record exists for either the volume or subvolume, any user can create a disk file on the subvolume.

The Safeguard software does not restrict the creation of temporary files, such as swap files. Volume and subvolume authorization records are not checked when a temporary file is created.

Volume and Subvolume Security Command Summary

Table 9-1 on page 9-4 lists the disk volume and subvolume security commands and gives a brief description of each.

Table 9-1. Disk Volume and Subvolume Security Command Summary

Command	Description
ADD [SUB]VOLUME*	Adds a volume or subvolume authorization record with the specified attribute values. The current default volume or subvolume attribute values are used for any attributes not specified in the ADD VOLUME or ADD SUBVOLUME command. Only a local super group user can add a record for a disk volume unless the default action is overridden with an ACL for OBJECTTYPE VOLUME or SUBVOLUME.
ALTER [SUB]VOLUME*	Changes one or more attribute values in a volume or subvolume authorization record. For all attributes except ACCESS, ALTER replaces the current value with the specified value. For ACCESS, ALTER changes the existing ACL to incorporate *access-spec*.
DELETE [SUB]VOLUME*	Deletes an authorization record for a volume or subvolume. After a volume authorization record is deleted, only attempts to create disk files on protected subvolumes on the volume are subject to Safeguard security checks or auditing. After a subvolume authorization record is deleted, attempts to create disk files on the subvolume are subject to Safeguard security checks and auditing only if the volume on which the subvolume resides is protected. In addition, in either case, disk files with persistent protection are subject to Safeguard security checks and auditing.
FREEZE [SUB]VOLUME*	Temporarily suspends the file-creation authority granted to users with a volume or subvolume ACL. On any frozen volume or subvolume, file creation and access authority is granted only to an owner, the primary owner’s group manager, and the super ID.
INFO [SUB]VOLUME*	Displays the existing attribute values in a disk volume or subvolume authorization record
RESET [SUB]VOLUME	Sets one or more default volume or subvolume attribute values to predefined values
SET [SUB]VOLUME	Sets one or more default volume or subvolume attribute values to specified values. When a volume or subvolume authorization record is added, the current default attribute values are used for any attributes not specified in the ADD command.
SHOW [SUB]VOLUME	Displays the current default values of the volume or subvolume attribute
THAW [SUB]VOLUME*	For frozen volumes or subvolumes, restores the file-creation authority granted to users on that volume or subvolume ACL

* The ADD, ALTER, DELETE, FREEZE,THAW, and INFO commands used with VOLUME or SUBVOLUME, when there is no existing VOLUME or SUBVOLUME matching the given pattern, will display the "Record not found" error.

Syntax of Disk Volume and Subvolume Security Commands

The rest of this section contains individual syntax descriptions for the SAFECOM disk volume and subvolume security commands. Commands are presented in alphabetical order, and most of the command descriptions contain these elements:

- A summary of the function performed by the command, including restrictions on who can use the command
- The syntax of the command, including descriptions of the command parameters and variables
- The format for the command listing or report (for commands that produce listings or reports)
- Considerations for using the command
- Examples of command usage

ADD VOLUME and SUBVOLUME Commands

ADD VOLUME creates one or more volume authorization records. ADD SUBVOLUME creates one or more subvolume authorization records. After an authorization record is created for a volume or subvolume, all attempts to create a disk file on the volume or subvolume are subject to a Safeguard authorization check and, optionally, to Safeguard auditing.

By default, only a local super group user can add a volume authorization record, but any local user can add a subvolume authorization record. For more information, see SUBVOLUME on page 12-2 and VOLUME on page 12-2.

You can use SET VOLUME or SET SUBVOLUME to establish a set of default attribute values. Then use either ADD VOLUME or ADD SUBVOLUME to name the volume or subvolume to which the default attribute values are to be applied. You can also specify values for volume or subvolume attributes in your ADD command. The current default values are used for any attributes not specified in the ADD command.

```
ADD VOLUME volume-list [ , ]

   [ LIKE [\system.]$volume | vol-subvol-attribute ]

   [ , vol-subvol-attribute ] ...


ADD SUBVOLUME subvol-list [ , ]

   [ LIKE [\system.][$volume.]subvol | vol-subvol-attribute ]

   [ , vol-subvol-attribute ] ...
```

`volume-list`

specifies one or more disk volumes for which authorization records are to be added. *`volume-list`* can be either of:

```
  $volume

( $volume [ , $volume ] ... )
```

`$volume`

can be any volume name. The name can contain wild-card characters.

`OBJECT-TEXT-DESCRIPTION`

allows you to store printable characters, which are associated with the objects, as comments. These comments can be used to manage the object authorization record.

The text description field can accommodate 255 bytes of text data.

Note. The text specified in the text description field overwrites existing data, if any. Also, when LIKE clause is used with ADD VOLUME or ADD SUBVOLUME command, the object text description field is not copied with other object authorization record attributes.

The OBJECT-TEXT-DESCRIPTION attribute is supported only on systems running H06.16 and later H-series RVUs.

`subvol-list`

specifies one or more subvolumes for which authorization records are to be added. *`subvol-list`* can be either:

```
  subvol-name

( subvol-name [ , subvol-name ... ] )
```

`subvol-name`

can be any subvolume name. The name can contain wild-card characters.

```
LIKE [\system.]$volume
LIKE [\system.][$volume.]subvol
```

adopts the existing attribute values of the specified volume or subvolume as the attribute values to be used for the authorization record or records being added. If you omit *`\system`*, your current default system name is used. Similarly, if you omit *`$volume`* in a subvolume specification, your current default volume name is used.

vol-subvol-attribute

defines an attribute value for the volume or subvolume for which an authorization record is being added. The *vol-subvol-attribute*s are:

```
OWNER [owner-id]
ACCESS access-spec [ ; access-spec ] ...
OBJECT-TEXT-DESCRIPTION "[any-text]"
AUDIT-ACCESS-PASS [audit-spec]
AUDIT-ACCESS-FAIL [audit-spec]
AUDIT-MANAGE-PASS [audit-spec]
AUDIT-MANAGE-FAIL [audit-spec]
WARNING-MODE {ON|OFF}
```

OWNER [*owner-id*]

specifies the new owner of the volume or subvolume. *owner-id* can be either:

```
[\node-spec.]group-name.member-name
[\node-spec.]group-num , member-num
```

If you omit *owner-id*, *owner-id* is set to your user ID.

ACCESS *access-spec* [; *access-spec*] ...

changes the ACL for *filename-list* by adding or deleting ACL entries or by changing the authority list of a current ACL entry.

An ACL contains as many as 50 entries that grant or deny access authorities to users and user groups.

access-spec has the form:

```
user-list  [-] [DENY] authority-list
```

```
group-list [-] [DENY] authority-list
```

user-list

specifies users who are granted (or denied) the access authorities specified with the following *authority-list*. *user-list* can be either of:

```
  net-user-spec
```

```
( net-user-spec [ , net-user-spec ] ... )
```

net-user-spec can be any of these forms:

```
[\node-spec.]adm-group-name.user-name
[\node-spec.]adm-group-num , user-num
[\node-spec.]adm-group-name.*
[\node-spec.]adm-group-num , *
[\node-spec.]*.*
[\node-spec.]*,*
```

-

(minus-sign) operates on existing ACL entries. The minus-sign form of *access-spec* modifies the current default ACL. The *authority* entries are removed from the default ACL entries for the users specified with *user-list*.

group-list

can take either of the following forms:

net-group-spec

(*net-group-spec* [, *net-user-spec*] ...)

net-group-spec can take any of the following forms:

GROUP [NAME][\node-spec.] *group-name*e

GROUP NUMBER [\node-spec.]

node-spec

takes this form:

* | *node-name* | *node-number*

node-name

specifies the system name.

node-number

specifies the Expand node number.

adm-group-name

specifies the name of the administrative group.

adm-group-num

specifies the group number of an administrative group.

group-name

specifies the name of any group.

group-num

specifies the group number of any group.

-

(minus-sign) operates on existing ACL entries. The minus-sign form of *access-spec* modifies the current default ACL. The *authority* entries

are removed from the default ACL entries for the users specified with *user-list*.

Note. Specifying ACCESS *access-spec* through the ADD command does not override the current default ACL (established through the SET command). Instead, any ACL entries specified with the ADD command are added to the current default ACL, and the entire ACL is defined for the volume or subvolume whose authorization record is being added.

DENY

denies the user or users specified in the *user-list* the access authorities specified by *authority-list*.

authority-list

specifies the access authorities granted (or denied) to *user-list*.

authority-list can be any one of:

```
  authority
( authority [ , authority ] ... )
  *
```

authority

is the authority to create a disk file on the volume or subvolume being altered.

authority can be one of:

```
C[REATE]
O[WNER]
R[EAD]
W[RITE]
E[XECUTE]
P[URGE]
```

*

(asterisk) specifies all six authorities in any volume or subvolume *access-spec*s.

OBJECT-TEXT-DESCRIPTION “[any-text]”

allows you to store printable characters as comments. These comments are associated with the objects and are used to manage the object authorization record.

The text description field can accommodate 255 bytes of text data.

Note. The text specified in the text description field overwrites existing data, if any. Also, when LIKE clause is used with ADD VOLUME and SUBVOLUME command, the OBJECT-TEXT-DESCRIPTION field is not copied with other object authorization record attributes.

The OBJECT-TEXT-DESCRIPTION attribute is supported only on systems running J06.05 and later J-series RVUs and H06.16 and later H-series RVUs.

`AUDIT-ACCESS-PASS [`*`audit-spec`*`]`

changes the *`audit-spec`* for successful attempts to create a disk file on the volume or subvolume. The form of *`audit-spec`* is:

```
{ ALL | LOCAL | REMOTE | NONE }
```

For a description of each *`audit-spec`*, see the SET VOLUME and SUBVOLUME Commands on page 9-28. Omitting *`audit-spec`* specifies NONE.

`AUDIT-ACCESS-FAIL [`*`audit-spec`*`]`

changes the *`audit-spec`* for unsuccessful attempts to create a disk file on the volume or subvolume. The form of *`audit-spec`* is:

```
{ ALL | LOCAL | REMOTE | NONE }
```

For a description of each *`audit-spec`*, see the SET VOLUME and SUBVOLUME Commands on page 9-28. Omitting *`audit-spec`* specifies NONE.

`AUDIT-MANAGE-PASS [`*`audit-spec`*`]`

changes the *`audit-spec`* for successful attempts to manage a volume or subvolume authorization record. The form of *`audit-spec`* is:

```
{ ALL | LOCAL | REMOTE | NONE }
```

For a description of each *`audit-spec`*, see the SET VOLUME and SUBVOLUME Commands on page 9-28. Omitting *`audit-spec`* specifies NONE.

`AUDIT-MANAGE-FAIL [`*`audit-spec`*`]`

changes the *`audit-spec`* for unsuccessful attempts to manage a volume or subvolume authorization record. The form of *`audit-spec`* is:

```
{ ALL | LOCAL | REMOTE | NONE }
```

For a description of each *`audit-spec`*, see the SET VOLUME and SUBVOLUME Commands on page 9-28. Omitting *`audit-spec`* specifies NONE.

```
WARNING-MODE { ON | OFF }
```

defines whether warning mode is enabled for the specified volume or subvolume. The value is required. For more information on warning mode, see the *Safeguard Administrator's Manual*.

ON enables warning mode for the specified volume or subvolume. The initial value is OFF, which disables warning mode for the specified volume or subvolume.

Considerations

- Volume and subvolume security can be managed from remote nodes.

  By default, only a local super group user can add an authorization record for a volume, and only a local user can add an authorization record for a subvolume. However, when a user adds an authorization record for a volume or subvolume, the user can specify the owner (through the OWNER attribute) as a network user ID. A network user who owns a volume or subvolume authorization record can manage the Safeguard access controls from a remote node.

- Attributes in an ADD command affect only the record added.

  Any attribute specifications in an ADD VOLUME or ADD SUBVOLUME command affect only the authorization record being created and do not change the current default attribute values. This condition is also true for a LIKE clause in an ADD command.

Examples

1. Using a LIKE clause is a convenient way to add a new volume or subvolume authorization record that has the same attribute values as those defined for another volume or subvolume. For example, this command adds an authorization record for the volume $DATA1 with the same attribute values as those currently defined for $DATA2 and allows you to enter text description for the record:

   ```
   =ADD VOLUME $data1, OBJECT-TEXT-DESCRIPTION “Record Created”&
   ,LIKE $data2
   ```

2. Modify the attribute values specified in a LIKE clause by specifying the values for the attributes you want to change in the ADD command. For example, this command adds an authorization record for $DATA1 that has the same attribute values as $DATA2 except for the OWNER attribute and allows you to enter text description for the created record:

   ```
   =ADD VOLUME $data1, OBJECT-TEXT-DESCRIPTION “Record Created”&
   ,LIKE $data2, OWNER sales.mike
   ```

ALTER VOLUME and SUBVOLUME Commands

ALTER VOLUME changes one or more attribute values in a volume authorization record. ALTER SUBVOLUME changes one or more attribute values in a subvolume authorization record.

An owner of a volume, the primary owner’s group manager, and the super ID can change a volume authorization record. Similarly, an owner of a subvolume authorization record, the primary owner’s group manager, and the super ID can change a subvolume authorization record.

Except for the ACCESS attribute, specifying a new attribute value with either ALTER VOLUME or ALTER SUBVOLUME replaces the current attribute value with the specified value. Using ALTER to specify a new ACCESS *access-spec* adds the new *access-spec* to the existing ACL for the specified volume or subvolume. To remove existing ACL entries, use the minus-sign (-) form of *access-spec*.

```
ALTER VOLUME volume-list [ , ]

   { LIKE [\system.]$volume | vol-subvol-attribute }

   [ , vol-subvol-attribute ] ...


ALTER SUBVOLUME subvol-list [ , ]

   { LIKE [\system.][$volume.]subvol | vol-subvol-attribute }

   [ , vol-subvol-attribute ] ...
```

volume-list

specifies one or more disk volumes for which authorization records are to be changed. *volume-list* can be either of:

```
  $volume

( $volume [ , $volume ] ... )
```

$volume

can be any volume name. The name can contain wild-card characters.

subvol-list

specifies one or more subvolumes for which authorization records are to be changed. *subvol-list* can be either:

```
  subvol-name

( subvol-name [ , subvol-name ... ] )
```

subvol-name

can be any subvolume name. The name can contain wild-card characters.

```
LIKE [\system.]$volume
LIKE [\system.][$volume.]subvol
```

changes the attribute values of *volume-list* or *subvol-list* to be the same as those currently defined for the volume or subvolume specified in the LIKE attribute. If you omit *system*, your current default system name is used. Similarly, if you omit $*volume*, your current default volume name is used. For the ACCESS attribute, LIKE adds ACL entries or authorities only to existing entries. It does not replace or delete ACL entries or authorities.

vol-subvol-attribute

changes the value of the specified *vol-subvol-attribute* for the volume or subvolume whose authorization record is being altered. The *vol-subvol-attribute*s are:

```
OWNER [owner-id]
ACCESS access-spec [ ; access-spec ] ...
OBJECT-TEXT-DESCRIPTION “[any-text]”
RESET-OBJECT-TEXT-DESCRIPTION
AUDIT-ACCESS-PASS [audit-spec]
AUDIT-ACCESS-FAIL [audit-spec]
AUDIT-MANAGE-PASS [audit-spec]
AUDIT-MANAGE-FAIL [audit-spec]
WHERE option-list
WARNING-MODE {ON|OFF}
```

OWNER [*owner-id*]

specifies the new owner of the volume or subvolume. *owner-id* can be either:

```
[\node-spec.]group-name.member-name
[\node-spec.]group-num , member-num
```

If you omit *owner-id*, *owner-id* is set to your user ID.

ACCESS *access-spec* [; *access-spec*] ...

changes the ACL for *filename-list* by adding or deleting ACL entries or by changing the authority list of a current access-control-list entry.

An ACL contains as many as 50 entries that grant or deny access authorities to users and user groups.

access-spec has the form:

```
user-list  [-] [DENY] authority-list

group-list [-] [DENY] authority-list
```

user-list

specifies users who are granted (or denied) the access authorities specified with the following *authority-list*. *user-list* can be either of:

net-user-spec

(*net-user-spec* [, *net-user-spec*] ...)

net-user-spec can be any of these forms:

```
[\node-spec.]adm-group-name.user-name
[\node-spec.]adm-group-num , user-num
[\node-spec.]adm-group-name.*
[\node-spec.]adm-group-num , *
[\node-spec.]*.*
[\node-spec.]*,*
```

-

(minus-sign) operates on existing ACL entries. The minus-sign form of *access-spec* modifies the current default ACL. The *authority* entries are removed from the default ACL entries for the users specified with *user-list*.

group-list

can take either of the following forms:

net-group-spec

(*net-group-spec* [, *net-user-spec*] ...)

net-group-spec can take any of the following forms:

GROUP [NAME][\node-spec.] *group-nam*e

GROUP NUMBER [\node-spec.]

node-spec

takes this form:

* | *node-name* | *node-number*

node-name

specifies the system name.

node-number

specifies the Expand node number.

adm-group-name

specifies the name of the administrative group.

adm-group-num

specifies the group number of an administrative group.

group-name

specifies the name of any group.

group-num

specifies the group number of any group.

-

(minus-sign) operates on existing ACL entries. The minus-sign form of *access-spec* modifies the current default ACL. The *authority* entries are removed from the default ACL entries for the users specified with *user-list*.

DENY

denies the user or users specified in the *user-list* the access authorities specified by *authority-list*.

authority-list

specifies the access authorities granted (or denied) to *user-list*.

authority-list can be one of:

```
  authority

( authority [ , authority ] ... )

  *
```

authority

is the authority to create a disk file on the volume or subvolume being altered.

authority can be one of:

```
C[REATE]
O[WNER]
R[EAD]
W[RITE]
E[XECUTE]
P[URGE]
```

`*`

(asterisk) specifies all six authorities in any volume or subvolume *`access-spec`*s.

`OBJECT-TEXT-DESCRIPTION “[any-text]”`

allows you to store printable characters as comments. These comments are associated with the objects and are used to manage the object authorization record.

The text description field can accommodate 255 bytes of text data.

Note. The text specified in the text description field overwrites existing data, if any. Also, when LIKE clause is used with ALTER VOLUME and SUBVOLUME command, the OBJECT-TEXT-DESCRIPTION field is not copied with other object authorization record attributes. Also, if you specify OBJECT-TEXT-DESCRIPTION without any text in the quotation marks, the object text description for this record is removed.

The OBJECT-TEXT-DESCRIPTION attribute is supported only on systems running J06.05 and later J-series RVUs, H06.16 and later H-series RVUs, and G06.32 and later G-series RVUs.

`RESET-OBJECT-TEXT-DESCRIPTION`

Resets the object description to Null.

Note. The RESET-OBJECT-TEXT-DESCRIPTION attribute is supported only on systems running J06.05 and later J-series RVUs and H06.16 and later H-series RVUs and G06.32 and later G-series RVUs.

`AUDIT-ACCESS-PASS [`*`audit-spec`*`]`

changes the *`audit-spec`* for successful attempts to create or access a disk file on the volume or subvolume. The form of *`audit-spec`* is:

```
{ ALL | LOCAL | REMOTE | NONE }
```

For a description of each *`audit-spec`*, see the SET VOLUME and SUBVOLUME Commands on page 9-28. Omitting *`audit-spec`* specifies NONE.

`AUDIT-ACCESS-FAIL [`*`audit-spec`*`]`

changes the *`audit-spec`* for unsuccessful attempts to create or access a disk file on the volume or subvolume. The form of *`audit-spec`* is:

```
{ ALL | LOCAL | REMOTE | NONE }
```

For a description of each *`audit-spec`*, see the SET VOLUME and SUBVOLUME Commands on page 9-28. Omitting *`audit-spec`* specifies NONE.

`AUDIT-MANAGE-PASS [`*`audit-spec`*`]`

changes the *`audit-spec`* for successful attempts to manage a volume or subvolume authorization record. The form of *`audit-spec`* is:

```
{ ALL | LOCAL | REMOTE | NONE }
```

For a description of each *`audit-spec`*, see the SET VOLUME and SUBVOLUME Commands on page 9-28. Omitting *`audit-spec`* specifies NONE.

`AUDIT-MANAGE-FAIL [`*`audit-spec`*`]`

changes the *`audit-spec`* for unsuccessful attempts to manage a volume or subvolume authorization record. The form of *`audit-spec`* is:

```
{ ALL | LOCAL | REMOTE | NONE }
```

For a description of each *`audit-spec`*, see the SET VOLUME and SUBVOLUME Commands on page 9-28. Omitting *`audit-spec`* specifies NONE.

`WHERE` *`option-list`*

specifies that only volumes or subvolumes in *`filename-list`* that have LICENSE, PROGID, or WARNING-MODE set are to be altered.

`option-list` `has the form:`

```
( ] option [ OR option ] [ ) ]
```

`option`

can be either:

```
PROGID
LICENSE
WARNING-MODE
```

`WARNING-MODE { ON | OFF }`

defines whether warning mode is enabled for the specified volume or subvolume. The value is required. For more information on warning mode, see the *Safeguard Administrator's Manual*.

ON enables warning mode for the specified volume or subvolume. The initial value is OFF, which disables warning mode for the specified volume or subvolume.

Examples

This command transfers ownership of the RECORDS subvolume to the user with user ID 86,13 and allows all users who are members of group number 86 to create files on the subvolume and add object text description:

```
=ALTER SUBVOLUME records, OBJECT-TEXT-DESCRIPTION “Record &
altered”,OWNER 86,13, ACCESS 86,* c
```

DELETE VOLUME and SUBVOLUME Commands

DELETE VOLUME deletes a disk volume authorization record. DELETE SUBVOLUME deletes a subvolume authorization record. After a volume or subvolume authorization record is deleted, the volume or subvolume is no longer subject to Safeguard authorization checks or auditing.

An owner of a volume, the primary owner’s group manager, and the super ID can delete a volume authorization record. Similarly, an owner of a subvolume authorization record, the primary owner’s group manager, and the super ID can delete a subvolume authorization record.

```
DELETE VOLUME volume-list [ [ , ] WHERE option-list ]


DELETE SUBVOLUME subvol-list [ [ , ] WHERE option-list ]
```

volume-list

specifies one or more disk volumes for which authorization records are to be deleted. *volume-list* can be either of:

`$volume`

`( $volume [ , $volume ] ... )`

$volume

can be any volume name. The name can contain wild-card characters.

subvol-list

specifies one or more subvolumes for which authorization records are to be deleted. *subvol-list* can be either:

`subvol-name`

`( subvol-name [ , subvol-name ... ] )`

subvol-name

can be any subvolume name. The name can contain wild-card characters.

`WHERE` *`option-list`*

specifies that only volumes or subvolumes in *`filename-list`* that have LICENSE, PROGID, or WARNING-MODE set are to be deleted.

`option-list` `has the form:`

```
[ ( ] option [ OR option ] [ ) ]
```

`option`

can be either:

```
PROGID
LICENSE
WARNING-MODE
```

Examples

The user who owns the authorization records for the three subvolumes MAIL, PERSNL, and REPORTS can delete the Safeguard authorization records by entering this command:

```
=DELETE SUBVOLUME (mail, persnl, reports)
```

Now these three subvolumes are no longer subject to Safeguard access control or auditing. However, the Safeguard volume access controls remain in effect for any of the subvolumes that reside on a disk volume protected by the Safeguard software.

FREEZE VOLUME and SUBVOLUME Commands

FREEZE VOLUME temporarily suspends the access authorities granted to users through a volume ACL. While a volume is frozen, only the owners of the volume's authorization records, the primary owner's group manager, and the super ID can create or access disk files on the volume.

Similarly, FREEZE SUBVOLUME temporarily suspends the access authorities granted to users through a subvolume ACL. While a subvolume is frozen, only the owners of the subvolume authorization record, the primary owner's group manager, and the super ID can create or access disk files on the subvolume.

An owner of a volume, the primary owner's group manager, and the super ID can freeze a volume authorization record. Similarly, an owner of a subvolume authorization record, the primary owner's group manager, and the super ID can freeze a subvolume authorization record.

To restore a frozen volume ACL, use the THAW VOLUME command. To restore a frozen subvolume ACL, use the THAW SUBVOLUME command.

```
FREEZE VOLUME volume-list [ [ , ] WHERE option-list ]


FREEZE SUBVOLUME subvol-list [ [ , ] WHERE option-list ]
```

volume-list

specifies one or more disk volumes that are to be frozen. *volume-list* can be either:

```
  $volume

( $volume [ , $volume ] ... )
```

$volume

can be any volume name. The name can contain wild-card characters.

subvol-list

specifies one or more subvolumes that are to be frozen. *subvol-list* can be either:

```
  subvol-name

( subvol-name [ , subvol-name ... ] )
```

subvol-name

can be any subvolume name. The name can contain wild-card characters.

WHERE *option-list*

specifies that only volumes or subvolumes in *filename-list* that have LICENSE, PROGID, or WARNING-MODE set are to be frozen.

option-list has the form:

```
[ ( ] option [ OR option ] [ ) ]
```

option

can be either:

```
PROGID
LICENSE
WARNING-MODE
```

Examples

User PRS.HARRY is about to leave on vacation. To protect his files from tampering or loss, he suspends access to important subvolumes:

```
=FREEZE SUBVOLUME ($data.harry, $data.hgmail,&
=$data.hgsales)
```

Now no one (except Harry and his group manager) can create files on these three subvolumes until Harry or his manager enters a THAW SUBVOLUME command.

INFO VOLUME and SUBVOLUME Commands

INFO VOLUME displays reports on the attribute values currently stored in a volume authorization record. INFO SUBVOLUME displays reports on the attribute values currently stored in a subvolume authorization record. Each INFO command produces two types of reports: brief and detailed. The formats for the two report types are illustrated following the command syntax.

Any user can produce an INFO report on any volume or subvolume.

```
INFO [ /OUT listfile/ ] VOLUME  volume-list [ [ , ] DETAIL ]

INFO [ /OUT listfile/ ] SUBVOLUME

   subvol-list [ [ , ] DETAIL ]
```

OUT *listfile*

directs the INFO report to *listfile*. After executing the INFO command, SAFECOM redirects its output to the current OUT file.

For *listfile*, specify any file name. SAFECOM opens *listfile* and appends the INFO report to the file. If *listfile* does not exist, SAFECOM creates an EDIT file and writes the INFO report to that file.

volume-list

specifies one or more disk volumes for which INFO reports are to be produced. *volume-list* can be either:

$*volume*

($*volume* [, $*volume*] ...)

$*volume*

can be any volume name. The name can contain wild-card characters.

subvol-list

specifies one or more subvolumes for which INFO reports are to be produced. *subvol-list* can be either:

subvol-name

(*subvol-name* [, *subvol-name* ...])

subvol-name

can be any subvolume name. The name can contain wild-card characters.

DETAIL

adds the current *audit-spec*s for the volumes or subvolumes being reported. For a full description of the four *audit-spec*s, see the SET VOLUME and SUBVOLUME Commands on page 9-28.

INFO VOLUME and SUBVOLUME Brief Report

The brief INFO VOLUME or INFO SUBVOLUME report gives you information about the volumes or subvolumes you specify. Figure 9-1 illustrates the format of the brief INFO report for both volumes and subvolumes.

Figure 9-1. INFO VOLUME and SUBVOLUME Brief Report Format

```
                 LAST-MODIFIED OWNER    STATUS    WARNING-MODE
$volume[.subvol]
                  date, time   owner-id status      {ON|OFF}

    user-spec [DENY] authority
    user-spec [DENY] authority
       .
       .
       .
 [ NO ACCESS CONTROL LIST DEFINED! ]
```

Figure 9-1 contains the following attribute values and status fields:

$volume

for INFO VOLUME reports, is the name of the disk volume whose existing attribute values are being displayed.

$volume.subvol

for INFO SUBVOLUME reports, is the name of the subvolume whose existing attribute values are being displayed.

`LAST MODIFIED`
`date, time`

indicates the date and time of the last change made to this volume or subvolume authorization record. *`date`* and *`time`* are in local civil time.

`OWNER`
`owner-id`

is the user ID of the user who owns this volume or subvolume authorization record.

`STATUS`
`status`

indicates the current status of this volume or subvolume. *`status`* is FROZEN or THAWED.

`WARNING-MODE`
`{ON|OFF}`

is the current warning-mode state of this volume or subvolume. ON indicates that the protection record is in warning mode. The initial value is OFF, which indicates that warning mode is disabled for this volume or subvolume.

`user-spec` `[DENY]` *`authority`*

is an entry in the ACL for this volume or subvolume. *`user-spec`* identifies a single user or user group. *`user-spec`* has these forms:

```
group-num , member-num
group-num, *
*,*
\node-spec.group-num , member-num
\node-spec.group-num, *
\node-spec.*,*
```

`node-spec`

takes this form:

`* | node-name | node-number`

`node-name`

specifies the system name.

`node-number`

specifies the Expand node number.

group-num , member-num

identifies a single local user.

group-num,*

identifies all the local users in the group that has *group-num*.

,

identifies all the local users at the node where this volume or subvolume resides.

\node-spec.*group-num , member-num*

identifies the local user who has the user ID *group-num*, *member-num* and a network user who has both the same user name and user ID as that local user.

\node-spec.*group-num*,*

identifies all the local users in the group that has *group-num* and all network users in that group who have the same *group-num*.

\node-spec.*,*

identifies all the local users on this volume or subvolume node and all network users who have access to this node.

DENY

indicates that an access authority is specifically denied to *user-spec*.

authority

can contain one of R, W, E, P, C, or O.

R	READ authority for the volume or subvolume
W	WRITE authority for the volume or subvolume
E	EXECUTE authority for the volume or subvolume
P	PURGE authority for the volume or subvolume
C	CREATE authority for the volume or subvolume
O	OWNER authority for the volume or subvolume

[NO ACCESS CONTROL LIST DEFINED!]

indicates this volume or subvolume has no default ACL. Use ALTER...ACCESS to define ACL entries. Only the local super ID can access a volume or subvolume that has no ACL.

INFO VOLUME and SUBVOLUME Detailed Report

The detailed INFO VOLUME and SUBVOLUME report includes the auditing specifications for the protected volume or subvolume. Figure 9-2 shows the format of the detailed INFO VOLUME and SUBVOLUME report.

Figure 9-2. INFO VOLUME and SUBVOLUME Detailed Report Format

```
                 LAST-MODIFIED OWNER    STATUS    WARNING-MODE
$volume[.subvol]
                  date, time   owner-id status      {ON|OFF}

    user-spec [DENY] authority
    user-spec [DENY] authority
       .
       .
       .
 [ NO ACCESS CONTROL LIST DEFINED! ]

  OBJECT-TEXT-DESCRIPTION =

  AUDIT-ACCESS-PASS = a-spec  AUDIT-MANAGE-PASS = a-spec
  AUDIT-ACCESS-FAIL = a-spec  AUDIT-MANAGE-FAIL = a-spec
```

In addition to the attributes displayed in the brief INFO report, the detailed INFO report displays these attribute values:

```
AUDIT-ACCESS-PASS = a-spec  AUDIT-MANAGE-PASS = a-spec
AUDIT-ACCESS-FAIL = a-spec  AUDIT-MANAGE-FAIL = a-spec
```

These values indicate the conditions under which the Safeguard software audits attempts to create a disk file on this volume or subvolume and attempts to manage this authorization record.

a-spec can be:

```
   { ALL | LOCAL | REMOTE | NONE }
```

For a full description of each *a-spec*, see the appropriate *audit-spec* under the SET VOLUME and SUBVOLUME Commands on page 9-28.

Examples

Using the ASSUME SUBVOLUME and the INFO commands, a user displays a report for the subvolume RAGS on the disk volume $SILK:

```
=ASSUME SUBVOLUME
=INFO $silk.rags
```

The display shows:

```
                   LAST-MODIFIED       OWNER      STATUS     WARNING-MODE
$SILK.RAGS
                 15AUG86, 12:22    \*.86,2      THAWED          OFF

      \*.086,002            C
         086,010            C
         086,255            C,O
```

RESET VOLUME and SUBVOLUME Commands

RESET VOLUME resets the current default values of the volume attribute values to their predefined values. RESET SUBVOLUME resets the current default subvolume attributes to their predefined values.

When you add an authorization record for a volume or a subvolume, the current default attribute values are used for any attributes you do not specify in the ADD command. (To set the default attribute values to specific values, use the SET VOLUME or SET SUBVOLUME command.)

```
RESET VOLUME [ [ , ] vol-subvol-attribute-keyword ]

   [ , vol-subvol-attribute-keyword ] ...


RESET SUBVOLUME [ [ , ] vol-subvol-attribute-keyword ]

   [ , vol-subvol-attribute-keyword ] ...
```

`VOLUME | SUBVOLUME`

specifies either VOLUME or SUBVOLUME as the object type of the RESET command. Omit it if the correct VOLUME or SUBVOLUME is the assumed object type. (For more information, see the ASSUME Command on page 4-3.)

`vol-subvol-attribute-keyword`

sets the current default value of the specified attribute to a predefined value. The *`vol-subvol-attribute-keyword`*s and predefined values are:

OWNER	The user ID of the current user
ACCESS	Null (no ACL)
OBJECT-TEXT-DESCRIPTION	Null (no descriptive text or blank)
AUDIT-ACCESS-PASS	NONE (no auditing)
AUDIT-ACCESS-FAIL	NONE (no auditing)
AUDIT-MANAGE-PASS	NONE (no auditing)
AUDIT-MANAGE-FAIL	NONE (no auditing)
WARNING-MODE	OFF (warning mode disabled)

For a complete description of each *`vol-subvol-attribute`*, see the SET VOLUME and SUBVOLUME Commands on page 9-28.

Consideration

- Specifying an attribute name without a value in an ADD or ALTER command causes the attribute to be assigned the predefined default value (as defined for the RESET command).
- If you enter RESET VOLUME or RESET SUBVOLUME (or RESET when VOLUME or SUBVOLUME is the assumed object type) but you do not include any *vol-subvol-attribute-keyword*, all the current default values for the volume or subvolume attributes are returned to their predefined values. The predefined values are listed before the syntax for SET VOLUME and SUBVOLUME Commands on page 9-28.

Examples

In this example, a RESET VOLUME command restores all the volume attributes to their predefined values.

First the SHOW VOLUME command displays the volume status:

```
=SHOW VOLUME
```

```
TYPE           OWNER     WARNING-MODE
 VOLUME         33,13        OFF

  OBJECT-TEXT-DESCRIPTION =

  AUDIT-ACCESS-PASS = ALL       AUDIT-MANAGE-PASS = ALL
  AUDIT-ACCESS-FAIL = NONE      AUDIT-MANAGE-FAIL = NONE

         018,*                C
         033,*                C
         086,*                C
```

Then the RESET VOLUME command is entered:

```
=RESET VOLUME
```

Another SHOW VOLUME command is entered:

```
=SHOW VOLUME
```

The display shows:

```
TYPE           OWNER      WARNING-MODE
 VOLUME         86,2          OFF

 OBJECT-TEXT-DESCRIPTION =

  AUDIT-ACCESS-PASS = NONE      AUDIT-MANAGE-PASS = NONE
  AUDIT-ACCESS-FAIL = NONE      AUDIT-MANAGE-FAIL = NONE

  NO ACCESS CONTROL LIST DEFINED!
```

SET VOLUME and SUBVOLUME Commands

SET VOLUME establishes default values for one or more volume attributes. SET SUBVOLUME establishes default values for one or more subvolume attributes. When you add an authorization record for a volume or subvolume, the current default values for the volume or subvolume attributes are used for any attributes you do not specify in your ADD command.

To display the current default attribute values, use the SHOW VOLUME or SHOW SUBVOLUME command.

```
SET VOLUME volume-list [ , ]

   { LIKE [\system.]$volume | vol-subvol-attribute }

   [ , vol-subvol-attribute ] ...


SET SUBVOLUME subvol-list [ , ]

   { LIKE [\system.][$volume.]subvol | vol-subvol-attribute }

   [ , vol-subvol-attribute ] ...
```

```
LIKE [\system.]$volume
LIKE [\system.][$volume.]subvol
```

sets the current default volume or subvolume attributes to the same as those currently defined for *volume* or *subvolume*. If you omit *\system*, your current default system name is used. Similarly, if you omit *$volume*, your current default volume name is used.

vol-subvol-attribute

defines a default value for the specified volume or subvolume attribute. The *vol-subvol-attributes* are:

```
OWNER [owner-id]
ACCESS access-spec [ ; access-spec ] ...
OBJECT-TEXT-DESCRIPTION "[any-text]"
AUDIT-ACCESS-PASS [audit-spec]
AUDIT-ACCESS-FAIL [audit-spec]
AUDIT-MANAGE-PASS [audit-spec]
AUDIT-MANAGE-FAIL [audit-spec]
WARNING-MODE {ON|OFF}
```

```
OWNER [owner-id]
```

specifies the owner of a volume or subvolume. *owner-id* can be either:

```
[\node-spec.]group-name.member-name
[\node-spec.]group-num , member-num
```

If you omit *owner-id*, *owner-id* is set to your user ID (that is, the user ID of the current user).

```
ACCESS access-spec [ ; access-spec ] ...
```

changes the ACL for *filename-list* by adding or deleting ACL entries or by changing the authority list of a current ACL entry.

An ACL contains as many as 50 entries that grant or deny access authorities to users and user groups.

access-spec has the form:

```
user-list  [-] [DENY] authority-list
group-list [-] [DENY] authority-list
```

user-list

specifies users who are granted (or denied) the access authorities specified with the following *authority-list*. *user-list* can be either:

```
  net-user-spec

( net-user-spec [ , net-user-spec ] ... )
```

net-user-spec can be any of:

```
[\node-spec.]adm-group-name.user-name
[\node-spec.]adm-group-num , user-num
[\node-spec.]adm-group-name.*
[\node-spec.]adm-group-num , *
[\node-spec.]*.*
[\node-spec.]*,*
```

-

(minus-sign) operates on existing ACL entries. The minus-sign form of *access-spec* modifies the current default ACL. The *authority* entries are removed from the default ACL entries for the users specified with *user-list*.

group-list

can be either of:

```
  net-group-spec

( net-group-spec [ , net-user-spec ] ... )
```

net-group-spec can be any of:

```
GROUP [NAME][\node-spec.] group-name

GROUP NUMBER [\node-spec.]
```

node-spec

takes this form:

`* | `*`node-name`*` | `*`node-number`*

node-name

specifies the system name.

node-number

specifies the Expand node number.

adm-group-name

specifies the name of the administrative group.

admin-group- name

specifies the group number of an administrative group.

group-name

specifies the name of any group.

group-num

specifies the group number of any group.

-

(minus-sign) operates on existing ACL entries. The minus-sign form of *access-spec* modifies the current default ACL. The *authority* entries are removed from the default ACL entries for the users specified with *user-list*.

DENY

specifically denies *user-list* the access authorities specified by *authority-list*:

authority-list

specifies the access authorities to be granted (or denied) to the user or users specified with *user-list*.

authority-list can be:

authority

authority [, *authority*] ...)

*

`authority`

is the authority to create and access a disk file on a volume or subvolume.

`authority` can be any of:

```
R[EAD]
W[RITE]
E[XECUTE]
P[URGE]
C[REATE]
O[WNER]
```

`*`

(asterisk) all authorities in any volume or subvolume *`access-spec`*.

`OBJECT-TEXT-DESCRIPTION "[any-text]"`

allows you to store printable characters as comments. These comments are associated with the objects and are used to manage the object authorization record.

The text description field can accommodate 255 bytes of text data.

Note. The text specified in the text description field overwrites existing data, if any. Also, when LIKE clause is used with SET VOLUME and SUBVOLUME command, the OBJECT-TEXT-DESCRIPTION field is not copied with other object authorization record attributes

The OBJECT-TEXT-DESCRIPTION attribute is supported only on systems running J06.05 and later J-series RVUs and H06.16 and later H-series RVUs and G06.32 and later G-series RVUs.

`AUDIT-ACCESS-PASS [`*`audit-spec`*`]`

establishes an *`audit-spec`* for successful attempts to create or access a disk file on a volume or a subvolume. This *`audit-spec`* specifies the conditions under which an audit record is written to the audit file when a disk file is created or accessed.

The form of *`audit-spec`* is:

```
{ ALL | LOCAL | REMOTE | NONE }
```

`ALL`

All successful attempts to create or access a disk file are audited.

`LOCAL`

Only successful attempts to create or access a disk file by local users are audited.

`REMOTE`

Only successful attempts to create or access a disk file by remote users are audited.

`NONE`

No successful attempts to create or access a disk file are audited.

Omitting *`audit-spec`* specifies NONE.

`AUDIT-ACCESS-FAIL [`*`audit-spec`*`]`

establishes an *`audit-spec`* for unsuccessful attempts to create or access a disk file on a volume or subvolume. This *`audit-spec`* specifies the conditions under which an audit record is written to the audit file when an attempt to create or access a disk file fails.

The form of *`audit-spec`* is:

```
{ ALL | LOCAL | REMOTE | NONE }
```

`ALL`

All unsuccessful attempts to create or access a disk file are audited.

`LOCAL`

Only unsuccessful attempts to create or access a disk file by local users are audited.

`REMOTE`

Only unsuccessful attempts to create or access a disk file by remote users are audited.

`NONE`

No unsuccessful attempts to create or access a disk file are audited.

Omitting *`audit-spec`* specifies NONE.

`AUDIT-MANAGE-PASS [`*`audit-spec`*`]`

establishes an *`audit-spec`* for successful attempts to manage a volume or subvolume authorization record. This *`audit-spec`* specifies the conditions under which an audit record is written to the audit file when a volume or subvolume authorization record is successfully accessed.

The form of *`audit-spec`* is:

```
{ ALL | LOCAL | REMOTE | NONE }
```

`ALL`

All successful management attempts are audited.

`LOCAL`

Only successful management attempts by local users are audited.

`REMOTE`

Only successful management attempts by remote users are audited.

`NONE`

No successful management attempts are audited.

Omitting *`audit-spec`* specifies NONE.

`AUDIT-MANAGE-FAIL [`*`audit-spec`*`]`

establishes an *`audit-spec`* for unsuccessful attempts to manage a volume or subvolume authorization record. This *`audit-spec`* specifies the conditions under which an audit record is written to the audit file when an attempt to access a volume or subvolume authorization record fails.

The form of *`audit-spec`* is:

```
{ ALL | LOCAL | REMOTE | NONE }
```

`ALL`

All unsuccessful management attempts are audited.

`LOCAL`

Only unsuccessful management attempts by local users are audited.

`REMOTE`

Only unsuccessful management attempts by remote users are audited.

`NONE`

No unsuccessful management attempts are audited.

Omitting *`audit-spec`* specifies NONE.

`WARNING-MODE { ON | OFF }`

defines whether warning mode is enabled for the specified volume or subvolume. The value is required. For more information on warning mode, see the *Safeguard Administrator's Manual*.

ON enables warning mode for the specified volume or subvolume. The initial value is OFF, which disables warning mode for the specified volume or subvolume.

Examples

These commands allow all members of group 86 (except user 86,8) to create files on a subvolume for which an authorization record is added with the specified default values. Also, the Safeguard software audits successful attempts to manage the subvolume authorization record:

```
=ASSUME SUBVOLUME
=SET ACCESS 86,* c; 86,8 DENY c
=SET AUDIT-MANAGE-PASS all
=SHOW
```

The display shows:

```
TYPE            OWNER     WARNING-MODE
 SUBVOLUME       86,2         OFF

  OBJECT-TEXT-DESCRIPTION =

  AUDIT-ACCESS-PASS = NONE      AUDIT-MANAGE-PASS = ALL
  AUDIT-ACCESS-FAIL = NONE      AUDIT-MANAGE-FAIL = NONE

         086,008 DENY         C
         086,*                C
```

SHOW VOLUME and SUBVOLUME Commands

SHOW VOLUME displays the current default values for the volume attributes. SHOW SUBVOLUME displays the current default values for the subvolume attribute.

When you add a volume authorization record, the default volume attribute values are used for any attributes you do not specify in the ADD VOLUME command. Similarly, when you add a subvolume authorization record, the default subvolume attribute values are used for any attributes you do not specify in the ADD SUBVOLUME command.

```
SHOW [ / OUT listfile / ] { VOLUME | SUBVOLUME }
```

`OUT` *`listfile`*

directs the SHOW VOLUME or SHOW SUBVOLUME report to *`listfile`*. After you execute the SHOW command, SAFECOM redirects output to the current OUT file.

For *`listfile`*, specify any file name. SAFECOM opens *`listfile`* and appends the SHOW report to the file. If *`listfile`* does not exist, SAFECOM creates an EDIT-format file and writes the SHOW report to that file.

SHOW VOLUME and SUBVOLUME Report Format

Figure 9-3 on page 9-35 illustrates the format for the SHOW VOLUME and SHOW SUBVOLUME command display.

Figure 9-3. SHOW VOLUME and SUBVOLUME Report Format

```
TYPE             OWNER     WARNING-MODE
 {vol|svol}      gn,un        {ON|OFF}

  OBJECT-TEXT-DESCRIPTION =

  AUDIT-ACCESS-PASS = a-spec  AUDIT-MANAGE-PASS = a-spec
  AUDIT-ACCESS-FAIL = a-spec  AUDIT-MANAGE-FAIL = a-spec

    user-spec [DENY] authority
    user-spec [DENY] authority
         .        .          .
         .        .          .
  [ NO ACCESS CONTROL LIST DEFINED! ]
```

The SHOW report displays the following volume and subvolume attributes and values:

TYPE
{vol|svol}

indicates the object type for this SHOW command. {vol|svol} is either VOLUME or SUBVOLUME.

OWNER *gn,un*

indicates the user ID (group number and member number) of the user who will own this volume or subvolume authorization record if a volume or subvolume having these attribute values is added to Safeguard protection.

WARNING-MODE
{ON|OFF}

is the current warning-mode state of this volume or subvolume. ON indicates that the protection record is in warning mode. The initial value is OFF, which indicates that warning mode is disabled for this volume or subvolume.

AUDIT-ACCESS-PASS = *a-spec*   AUDIT-MANAGE-PASS = *a-spec*
AUDIT-ACCESS-FAIL = *a-spec*   AUDIT-MANAGE-FAIL = *a-spec*

indicates the conditions under which the Safeguard software audits attempts to create files on this volume or subvolume and attempts to manage this authorization record. These fields are described in detail for *audit-spec* under the SET {VOLUME|SUBVOLUME} command.

user-spec [DENY] *authority*

provides an ACL entry for this volume or subvolume. For more information, see INFO VOLUME and SUBVOLUME Brief Report on page 9-22.

`[ NO ACCESS CONTROL LIST DEFINED! ]`

indicates no default ACL entries are defined. Use SET...ACCESS to define default ACL entries. You can also use ADD...ACCESS to define ACL entries when you create an authorization record.

△ **Caution.** If you do not specify an ACL for a volume or subvolume, only the local super ID can access the volume or subvolume.

Examples

The following SHOW VOLUME report displays the current default attribute values:

```
=SHOW VOLUME
```

```
TYPE           OWNER      WARNING-MODE
 VOLUME        86,2            OFF

  OBJECT-TEXT-DESCRIPTION =

  AUDIT-ACCESS-PASS = ALL       AUDIT-MANAGE-PASS = ALL
  AUDIT-ACCESS-FAIL = NONE      AUDIT-MANAGE-FAIL = NONE

         018,*               C
         033,*               C
         086,*               C
```

This SHOW VOLUME display indicates that:

- Any member of user groups 18, 33, and 86 can create files on this volume.
- Auditing is specified for all successful attempts to create or access files on this volume or to manage the volume’s authorization record.
- The user ID (86,2) owns this volume.

THAW VOLUME and SUBVOLUME Commands

THAW VOLUME restores the file-creation authority granted to users by a frozen volume ACL. After a frozen volume is thawed, any user granted file-creation authority by the volume ACL can once again create disk files on that volume.

Similarly, THAW SUBVOLUME thaws a frozen subvolume ACL.

An owner of a volume, the primary owner’s group manager, and the super ID can thaw a volume authorization record. Similarly, an owner of a subvolume authorization record, the primary owner’s group manager, and the super ID can thaw a subvolume authorization record.

THAW VOLUME and THAW SUBVOLUME have no effect on volumes and subvolumes that are not frozen.

```
THAW VOLUME volume-list [ [ , ] WHERE option-list ]

THAW SUBVOLUME subvol-list [ [ , ] WHERE option-list ]
```

volume-list

specifies one or more disk volumes to be thawed. *volume-list* can be either:

```
  $volume

( $volume [ , $volume ] ... )
```

$volume

can be any volume name. The name can contain wild-card characters.

subvol-list

specifies one or more subvolumes to be thawed. *subvol-list* can be either:

```
  subvol-name

( subvol-name [ , subvol-name ... ] )
```

subvol-name

can be any subvolume name. The name can contain wild-card characters.

WHERE *option-list*

specifies that only volumes or subvolumes in *filename-list* that have LICENSE, PROGID, or WARNING-MODE set are to be thawed.

option-list has the form:

```
[ ( ] option [ OR option ] [ ) ]
```

option

can be either:

```
PROGID
LICENSE
WARNING-MODE
```

Examples

The owner of subvolume $DATA.DEBITS uses the following commands to check the status of the subvolume and then restore the frozen subvolume ACL:

```
=INFO SUBVOLUME $data.debits
```

This display shows:

```
                  LAST-MODIFIED     OWNER     STATUS     WARNING-MODE
$DATA.DEBITS
                  9NOV86, 11:38      33,13    FROZEN          OFF

          033,013             C
```

These commands are entered:

```
=THAW SUBVOLUME $data.debits
=INFO SUBVOLUME $data.debits
```

This display shows:

```
                  LAST-MODIFIED     OWNER     STATUS      WARNING-MODE
$DATA.DEBITS
                 16DEC86,  9:13      33,13    THAWED          OFF

          033,013             C
```

10

Device and Subdevice Security Commands

With SAFECOM device and subdevice security commands, any user whose ID appears in the access control list (ACL) as owner of a protected device or subdevice can control access to that device or subdevice.

By default, only a local super-group user can add a device or subdevice authorization record to the Safeguard object data base. After an authorization record is added for a device or subdevice, all attempts to open the device or subdevice are subject to a Safeguard authorization check and, optionally, to Safeguard auditing. However, this behavior is configurable by creating or changing the ACL for OBJECTTYPE DEVICE or SUBDEVICE. For more information, see Section 12, OBJECTTYPE Security Commands.

The owner of a device or subdevice authorization record can control access to the device or subdevice by managing the ACL for that device. A device authorization record owner can also specify when the Safeguard software is to audit attempts to access the device or subdevice as well as attempts to manage the device or subdevice authorization record.

This section describes device and subdevice ownership and explains how the Safeguard software authorizes attempts to access protected devices and subdevices. It also summarizes the device and subdevice security commands. Following the command summary, the commands are described in detail.

Device and Subdevice Authorization Record Ownership

A device or subdevice has no authorization record until the device or subdevice is placed under the control of the Safeguard software facility by a super-group user. (For more information on adding authorization records, see DEVICE on page 12-2 or SUBDEVICE on page 12-2.) Every authorization record has an OWNER attribute that contains the user ID of the user who can manage the Safeguard access controls for the device or subdevice.

However, the user who adds the record can set the OWNER attribute to the user ID of any user (by including an OWNER specification in a SET DEVICE or SET SUBDEVICE or ADD DEVICE or ADD SUBDEVICE command). The owner of a protected authorization record can also transfer ownership to another user by changing the OWNER attribute with the ALTER DEVICE or ALTER SUBDEVICE command.

Because the primary owner can add owners to an ACL, additional ownership is defined by the OWNER authority code for ACL entries and is an independent extension of the primary owner. Additional owners can do anything that the primary owner is permitted to do. They are equal, in every way, to the primary owner. For example, they can modify the Safeguard authorization records for any device or subdevice they own, and they can access any device or subdevice for which they own the authorization record when that device or subdevice has been FROZEN.

An owner can deny explicitly a local super ID any of the authorities implicitly granted to the super ID (including OWNER) and have this denial actively enforced all of the time. When a device or subdevice is under Safeguard protection, all the security attributes are controlled by the Safeguard software for that device or subdevice.

When used with an ACL, the OWNER authority can always be specified for all devices or subdevices protected by the Safeguard software. The OWNER authority is always included when the * authority code is used. It can also be abbreviated as O for simplicity.

With the Safeguard software, the owner of an authorization record can also be defined as a network user. A network user who owns an authorization record can use the Safeguard software from a remote node to control access (provided the user has remote passwords set up between the two systems).

Device and Subdevice Access Authorities

The ACL for a device or subdevice can grant any combination of these access authorities to users and user groups:

READ	Open a device or subdevice for input operations
WRITE	Open a device or subdevice for output operations
OWNER	Manage the authorization records

Device and Subdevice Access Authorization

When a process attempts to open a protected device or subdevice, the Safeguard software checks the process group list and the ACL to determine whether READ or WRITE authority is granted to the user identified by the process’s process accessor ID (PAID). If that user has READ or WRITE authority, the open request is allowed to complete successfully. If the user has neither READ nor WRITE authority, the open request is rejected with a security violation error (file error 48).

The Safeguard software distinguishes between local and remote open requests. A remote open request is made by a process that was started by a network user logged on to a remote system. When a process is remote with respect to the device or subdevice that it is attempting to open, the network user must also be granted remote access. Otherwise, the Safeguard software rejects the open request with a security violation error (file error 48).

For example, suppose a remote process with a PAID of 4,5 attempts to open a device or subdevice. The device ACL must grant either READ or WRITE authority to *.4,5,

.4,, or *.*,*. Otherwise, the open request is rejected with a security violation error (file error 48).

An open request that has passed the Safeguard authorization check can nevertheless fail. For example, if a process attempts to open a device or subdevice already opened by another process that has exclusive access, the second open attempt fails with file error 12 (file in use). (For more information, see the *System Procedure Calls Reference Manual*.)

Device and Subdevice Security Command Summary

Table 10-1 lists the device and subdevice security commands and gives a brief description of each.

Table 10-1. Device and Subdevice Security Command Summary (page 1 of 2)

Command	**Description**
ADD [SUB]DEVICE	Adds a device or subdevice authorization record with the specified device or subdevice attribute values. The current default device or subdevice attribute values are used for any attributes not specified in the ADD DEVICE or ADD SUBDEVICE command. (Only a local super-group user can add an authorization record for a device or subdevice unless otherwise specified by the OBJECTTYPE DEVICE or OBJECTTYPE SUBDEVICE ACL.)
ALTER [SUB]DEVICE	Changes one or more attribute values in a device or subdevice authorization record. For all attributes except ACCESS, ALTER DEVICE or SUBDEVICE replaces the current value with the specified value. For the ACCESS attribute, ALTER DEVICE or SUBDEVICE changes the existing ACL to incorporate *access-spec*.
DELETE [SUB]DEVICE	Deletes a device or subdevice authorization record. Afterward, requests to open the device or subdevice are no longer subject to Safeguard authorization checks or auditing.
FREEZE [SUB]DEVICE	Temporarily suspends the access authorities granted to users in a device or subdevice ACL. (Only the owner of an authorization record for a device or subdevice, the owner’s group manager, and the local super ID can access a frozen device or subdevice.)
INFO [SUB]DEVICE	Displays the existing attribute values in a device or subdevice authorization record.
RESET [SUB]DEVICE	Sets one or more default device or subdevice attribute values to predefined values.

Table 10-1. Device and Subdevice Security Command Summary (page 2 of 2)

Command	Description
SET [SUB]DEVICE	Sets one or more default device or subdevice attribute values to specified values. When a device or subdevice authorization record is added, the current default device or subdevice attribute values are used for any attributes not specified in the ADD DEVICE or ADD SUBDEVICE command.
SHOW [SUB]DEVICE	Displays the current default values of the device or subdevice attributes.
THAW [SUB]DEVICE	For frozen devices and subdevices, restores the access authorities granted to users in the device or subdevice ACL.

Syntax of Device and Subdevice Security Commands

The rest of this section contains individual syntax descriptions for the SAFECOM device and subdevice security commands. Commands are presented in alphabetical order, and most of the command descriptions contain these elements:

- A summary of the function performed by the command, including the restrictions on who can use the command
- The syntax of the command, including descriptions of the command parameters and variables
- The format for the command listing or report (for commands that produce listings or reports)
- Considerations for the use of the command
- Examples of command usage

ADD DEVICE and SUBDEVICE Commands

ADD DEVICE or SUBDEVICE creates a Safeguard authorization record for one or more devices or subdevices. After an authorization record is created for a device or subdevice, all attempts to access the device or subdevice are subject to Safeguard authorization checks and, optionally, to Safeguard auditing.

Only a local super-group user can add an authorization record for a device or subdevice unless otherwise specified by the OBJECTTYPE DEVICE or SUBDEVICE ACL.

You can use SET DEVICE or SET SUBDEVICE to establish default attribute values and then use ADD DEVICE or ADD SUBDEVICE to name the devices or subdevices to which the default attribute values are to be applied. You can also specify values for the device attributes in your ADD DEVICE or ADD SUBDEVICE command. The current default values are used for any attributes not specified in your command.

```
ADD DEVICE device-list [ , ]

   [ LIKE device-name | device-attribute ]

   [ , device-attribute ] ...


ADD SUBDEVICE subdevice-list [ , ]

   [ LIKE subdevice-name | device-attribute ]

   [ , device-attribute ] ...
```

device-list

specifies one or more devices for which authorization records are to be added. *device-list* can be either:

device-name

(*device-name* [, *device-name*] ...)

device-name

can be any device name. The name cannot contain wild-card characters.

LIKE *device-name*

adopts the existing device attribute values of *device-name* as the attribute values to be used for the authorization record or records being added.

device-name

identifies the device whose current *device-attribute* values are to be assigned to the subdevice authorization record or records being added. *device-name* can be any device name.

subdevice-list

specifies one or more subdevices for which authorization records are to be added. *subdevice-list* can be either:

subdevice-name

(*subdevice-name* [, *subdevice-name* ...])

subdevice-name

can be any subdevice name. The name cannot contain wild-card characters.

`LIKE` *`subdevice-name`*

adopts the existing device attribute values of *`subdevice-name`* as the attribute values to be used for the authorization record or records being added.

`subdevice-name`

identifies the subdevice whose current *`device-attribute`* values are to be assigned to the subdevice authorization record or records being added. *`subdevice-name`* can be any subdevice name.

`device-attribute`

defines a device attribute value for the device or subdevice authorization record or records being added. The *`device-attribute`*s are:

```
OWNER [owner-id]
ACCESS access-spec [ ; access-spec ] ...
OBJECT-TEXT-DESCRIPTION "[any-text]"
AUDIT-ACCESS-PASS [audit-spec]
AUDIT-ACCESS-FAIL [audit-spec]
AUDIT-MANAGE-PASS [audit-spec]
AUDIT-MANAGE-FAIL [audit-spec]
WARNING-MODE {ON|OFF}
```

`OWNER [`*`owner-id`*`]`

specifies the new owner of the device or subdevice being altered. *`owner-id`* can be either:

```
[\node-spec.]group-name.member-name
[\node-spec.]group-num , member-num
```

If you omit *`owner-id`*, *`owner-id`* is set to your user ID.

`ACCESS` *`access-spec`* `[ ;` *`access-spec`* `] ...`

changes the ACL for *`filename-list`* by adding or deleting ACL entries or by changing the authority list of a current ACL entry.

An ACL contains as many as 50 entries that grant or deny access authorities to users and user groups.

`access-spec` has the form:

```
user-list  [-] [DENY] authority-list

group-list [-] [DENY] authority-list
```

user-list

specifies users who are granted (or denied) the access authorities specified with the following *authority-list*. *user-list* can be either:

net-user-spec

(*net-user-spec* [, *net-user-spec*] ...)

net-user-spec can be any of:

```
[\node-spec.]adm-group-name.user-name
[\node-spec.]adm-group-num , user-num
[\node-spec.]adm-group-name.*
[\node-spec.]adm-group-num , *
[\node-spec.]*.*
[\node-spec.]*,*
```

-

(minus-sign) operates on existing ACL entries. The minus-sign form of *access-spec* modifies the current default ACL. The *authority* entries are removed from the default ACL entries for the users specified with *user-list*.

group-list

can be either of:

net-group-spec

(*net-group-spec* [, *net-user-spec*] ...)

net-group-spec can be any of:

GROUP [NAME][*node-spec*.] *group-nam*e

GROUP NUMBER [*node-spec*.]

node-spec

takes this form:

* | *node-name* | *node-number*

node-name

specifies the system name.

node-number

specifies the Expand node number.

adm-group-name

specifies the name of the administrative group.

adm-group-num

specifies the group number of an administrative group.

group-name

specifies the name of any group.

group-num

specifies the group number of any group.

-

(minus-sign) operates on existing ACL entries. The minus-sign form of *access-spec* modifies the current default ACL. The *authority* entries are removed from the default ACL entries for the users specified with *user-list*.

Note. Specifying ACCESS *access-spec* with ADD DEVICE or SUBDEVICE does not override the current default ACL (established through SET DEVICE or SUBDEVICE). Instead, any ACL entries specified in ADD DEVICE or SUBDEVICE are used to modify the current ACL, and then the entire ACL is defined for the device or subdevice authorization record being added.

DENY

denies the users or user groups specified by *user-list* the access authorities specified by *authority-list*.

authority-list

specifies the access authorities to be granted (or denied) to *user-list*. *authority-list* can be one of:

```
  authority

( authority [ , authority ] ... )

  *
```

authority

is one of:

```
R[EAD]
W[RITE]
O[WNER]
```

*

(asterisk) specifies all authorities (READ, WRITE, and OWNER).

`OBJECT-TEXT-DESCRIPTION "[any-text]"`

allows you to store printable characters as comments. These comments are associated with the objects and are used to manage the object authorization record.

The text description field can accommodate 255 bytes of text data.

Note. The text specified in the text description field overwrites existing data, if any. Also, when LIKE clause is used with ADD DEVICE and SUBDEVICE command, the OBJECT-TEXT-DESCRIPTION field is not copied with other object authorization record attributes.

The OBJECT-TEXT-DESCRIPTION attribute is supported only on systems running J06.05 and later J-series RVUs and H06.16 and later H-series RVUs and G06.32 and later G-series RVUs.

`AUDIT-ACCESS-PASS [`*`audit-spec`*`]`

changes the *audit-spec* for successful attempts to access the device or subdevice. The form of *audit-spec* is:

`{ ALL | LOCAL | REMOTE | NONE }`

For a description of the *audit-spec*s, see the SET DEVICE and SUBDEVICE Commands on page 10-26. Omitting *audit-spec* specifies NONE.

`AUDIT-ACCESS-FAIL [`*`audit-spec`*`]`

changes the *audit-spec* for unsuccessful attempts to access the device or subdevice. The form of *audit-spec* is:

`{ ALL | LOCAL | REMOTE | NONE }`

For a description of the *audit-spec*s, see the SET DEVICE and SUBDEVICE Commands on page 10-26. Omitting *audit-spec* specifies NONE.

`AUDIT-MANAGE-PASS [`*`audit-spec`*`]`

changes the *audit-spec* for successful attempts to manage this device or subdevice authorization record. The form of *audit-spec* is:

`{ ALL | LOCAL | REMOTE | NONE }`

For a description of the *audit-spec*s, see the SET DEVICE and SUBDEVICE Commands on page 10-26. Omitting *audit-spec* specifies NONE.

`AUDIT-MANAGE-FAIL [`*`audit-spec`*`]`

changes the *`audit-spec`* for unsuccessful attempts to manage this device or subdevice authorization record. The form of *`audit-spec`* is:

```
{ ALL | LOCAL | REMOTE | NONE }
```

For a description of the *`audit-spec`*s, see the SET DEVICE and SUBDEVICE Commands on page 10-26. Omitting *`audit-spec`* specifies NONE.

`WARNING-MODE { ON | OFF }`

defines whether warning mode is enabled for the specified device or subdevice. The value is required. For more information on warning mode, see the *Safeguard Administrator's Manual*.

ON enables warning mode for the specified device or subdevice. The initial value is OFF, which disables warning mode for the specified device or subdevice.

Consideration

Any attribute specifications in an ADD DEVICE or SUBDEVICE command affect only the authorization record being created and do not change the current default values. This condition is also true for a LIKE clause in an ADD DEVICE or SUBDEVICE command.

Example

You can use a LIKE *`device-name`* clause to define all the attribute values for a device and then change any of those values by specifying one or more attribute values after the LIKE attribute. For example, this command adds an authorization record for $LP2 that has the same device attribute values (and ACL) as $LP1 except for the OWNER attribute and provide its description:

```
=ADD DEVICE $lp2, OBJECT-TEXT-DESCRIPTION “Record created”,&
LIKE $lp1, OWNER super.bob
```

ALTER DEVICE and SUBDEVICE Commands

ALTER DEVICE or SUBDEVICE changes one or more attribute values in an authorization record.

An owner of a device, the primary owner’s group manager, and the super ID can alter a device authorization record. Similarly, an owner of a subdevice’s authorization record, the primary owner’s group manager, and the super ID can alter a subdevice authorization record.

Except for the ACCESS attribute, new attribute values specified in an ALTER DEVICE or SUBDEVICE command replace the existing attribute values with the specified values. Using ALTER DEVICE or SUBDEVICE to specify a new ACCESS *`access-`*

spec adds the new *access-spec* to the existing ACL. To remove authorities previously granted to users, use the minus-sign (-) form of *access-spec*.

```
ALTER DEVICE device-list [ , ]

   { LIKE device-name | device-attribute }

   [ , device-attribute ] ...

ALTER SUBDEVICE subdevice-list [ , ]

   { LIKE subdevice-name | device-attribute }

   [ , device-attribute ] ...
```

device-list

specifies one or more devices for which authorization records are to be changed. All devices specified must already have Safeguard authorization records (created through the ADD DEVICE command). *device-list* can be either:

device-name

(*device-name* [, *device-name*] ...)

device-name

can be any device name. The name can contain wild-card characters.

LIKE *device-name*

adopts the existing device attribute values of *device-name* as the attribute values to be used for the authorization record or records being altered. For the ACCESS attribute, LIKE only adds ACL entries or adds authorities to existing entries. It does not replace or delete ACL entries or authorities.

device-name

identifies the device whose current *device-attribute* values are to be assigned to the device authorization record or records being altered. *device-name* can be any device name.

subdevice-list

specifies one or more subdevices for which authorization records are to be changed. *subdevice-list* can be either:

subdevice-name

(*subdevice-name* [, *subdevice-name* ...])

subdevice-name

can be any subdevice name. The name can contain wild-card characters.

LIKE *subdevice-name*

adopts the existing device attribute values of *subdevice-name* as the attribute values to be used for the authorization record or records being altered.

subdevice-name

identifies the subdevice whose current *subdevice-attribute* values are to be assigned to the subdevice authorization record or records being changed. *subdevice-name* can be any subdevice name.

device-attribute

changes the existing value of the specified device attribute for the devices or subdevices being altered. The *device-attribute* variables are:

```
OWNER [owner-id]
ACCESS access-spec [ ; access-spec ] ...
OBJECT-TEXT-DESCRIPTION “[any-text]”
RESET-OBJECT-TEXT-DESCRIPTION
AUDIT-ACCESS-PASS [audit-spec]
AUDIT-ACCESS-FAIL [audit-spec]
AUDIT-MANAGE-PASS [audit-spec]
AUDIT-MANAGE-FAIL [audit-spec]
WHERE WARNING-MODE
WARNING-MODE {ON|OFF}
```

OWNER [*owner-id*]

specifies the new owner of the devices or subdevices being altered. *owner-id* can be either:

```
[\node-spec.]group-name.member-name
[\node-spec.]group-num , member-num
```

If you omit *owner-id*, *owner-id* is set to your user ID.

ACCESS *access-spec* [; *access-spec*] ...

changes the ACL for *filename-list* by adding or deleting ACL entries or by changing the authority list of a current ACL entry.

An ACL contains as many as 50 entries that grant or deny access authorities to users and user groups.

access-spec has the form:

```
user-list  [-] [DENY] authority-list
group-list [-] [DENY] authority-list
```

user-list

specifies users who are granted (or denied) the access authorities specified with the following *authority-list*. *user-list* can be either:

```
  net-user-spec

( net-user-spec [ , net-user-spec ] ... )
```

net-user-spec can be any of:

```
[\node-spec.]adm-group-name.user-name
[\node-spec.]adm-group-num , user-num
[\node-spec.]adm-group-name.*
[\node-spec.]adm-group-num , *
[\node-spec.]*.*
[\node-spec.]*,*
```

-

(minus-sign) operates on existing ACL entries. The minus-sign form of *access-spec* modifies the current default ACL. The *authority* entries are removed from the default ACL entries for the users specified with *user-list*.

group-list

can be either of

```
  net-group-spec

( net-group-spec [ , net-user-spec ] ... )
```

net-group-spec can be any of:

```
GROUP [NAME][\node-spec.] group-name

GROUP NUMBER [\node-spec.]
```

node-spec

takes this form:

```
* | node-name | node-number
```

node-name

specifies the system name.

node-number

specifies the Expand node number.

adm-group-name

specifies the name of the administrative group.

adm-group-num

specifies the group number of an administrative group.

group-name

specifies the name of any group.

group-num

specifies the group number of any group.

-

(minus-sign) operates on existing ACL entries. The minus-sign form of *access-spec* modifies the current default ACL. The *authority* entries are removed from the default ACL entries for the users specified with *user-list*.

DENY

denies the users or user groups specified by *user-list* the access authorities specified by *authority-list*.

authority-list

specifies the access authorities to be granted (or denied) to *user-list*. *authority-list* can be one of:

```
  authority
( authority [ , authority ] ... )
      *
```

authority

is any of:

```
R[EAD]
W[RITE]
O[WNER]
```

*

(asterisk) specifies read, write, and owner.

OBJECT-TEXT-DESCRIPTION “[any-text]”

allows you to store printable characters as comments. These comments are associated with the objects and are used to manage the object authorization record.

The text description field can accommodate 255 bytes of text data.

Note. The text specified in the text description field overwrites existing data, if any. Also, when LIKE clause is used with ALTER DEVICE and SUBDEVICE command, the OBJECT-TEXT-DESCRIPTION field is not copied with other object authorization record attributes. Also, if you specify OBJECT-TEXT-DESCRIPTION without any text in the quotation marks, the object text description for this record is removed.

The OBJECT-TEXT-DESCRIPTION attribute is supported only on systems running J06.05 and later J-series RVUs, H06.16 and later H-series RVUs, and G06.32 and later G-series RVUs.

`RESET-OBJECT-TEXT-DESCRIPTION`

Resets the object description to Null.

Note. The RESET-OBJECT-TEXT-DESCRIPTION attribute is supported only on systems running J06.05 and later J-series RVUs and H06.16 and later H-series RVUs and G06.32 and later G-series RVUs.

`AUDIT-ACCESS-PASS [`*`audit-spec`*`]`

changes the *audit-spec* for successful attempts to access the device or subdevice. The form of *audit-spec* is:

```
{ ALL | LOCAL | REMOTE | NONE }
```

For a description of the *audit-spec*s, see the SET DEVICE and SUBDEVICE Commands on page 10-26. Omitting *audit-spec* specifies NONE.

`AUDIT-ACCESS-FAIL [`*`audit-spec`*`]`

changes the *audit-spec* for unsuccessful attempts to access the device or subdevice. The form of *audit-spec* is:

```
{ ALL | LOCAL | REMOTE | NONE }
```

For a description of the *audit-spec*s, see the SET DEVICE and SUBDEVICE Commands on page 10-26. Omitting *audit-spec* specifies NONE.

`AUDIT-MANAGE-PASS [`*`audit-spec`*`]`

changes the *audit-spec* for successful attempts to manage this authorization record. The form of *audit-spec* is:

```
{ ALL | LOCAL | REMOTE | NONE }
```

For a description of the *audit-spec*s, see the SET DEVICE and SUBDEVICE Commands on page 10-26. Omitting *audit-spec* specifies NONE.

`AUDIT-MANAGE-FAIL [`*`audit-spec`*`]`

changes the *`audit-spec`* for unsuccessful attempts to manage this authorization record. The form of *`audit-spec`* is:

```
{ ALL | LOCAL | REMOTE | NONE }
```

For a description of the *`audit-spec`*s, see the SET DEVICE and SUBDEVICE Commands on page 10-26. Omitting *`audit-spec`* specifies NONE.

`WHERE WARNING-MODE`

specifies that only devices or subdevices in *`filename-list`* that have WARNING-MODE set are to be deleted.

`WARNING-MODE { ON | OFF }`

defines whether warning mode is enabled for the specified device or subdevice. The value is required. For more information on warning mode, see the *Safeguard Administrator's Manual*.

ON enables warning mode for the specified device or subdevice. The initial value is OFF, which disables warning mode for the specified device or subdevice.

Consideration

Using ALTER DEVICE or SUBDEVICE to change one or more attributes for a device or subdevice has no effect on any processes that currently have the device or subdevice open.

For example, if you change an ACL to deny both READ and WRITE authority to a user who is running a process that is currently accessing the device or subdevice, the user’s process continues accessing the device or subdevice until it closes the device. Then any attempt to reopen the device or subdevice results in a security violation error (file error 48).

Example

The owner of the tape drive $TAPE enters an ALTER DEVICE command to change the ACL for $TAPE. Afterward, user PRS.HARRY (user ID 86,2) can no longer read from or write to $TAPE, and the network user ID 33,13 can both read from and write to $TAPE.

To verify the current device status:

```
=INFO DEVICE $TAPE
```

This report shows:

```
                 LAST-MODIFIED     OWNER     STATUS    WARNING-MODE
$LPRINT
               18SEP87, 13:48   \*.86,255    THAWED        OFF

        086,001      R,W
        086,002      R,W
        086,003      R,W
        086,008      R,W
     \*.086,255      R,W
        255,*        R,W
```

To alter the ACL for the tape device:

```
=ALTER DEVICE $tape, ACCESS prs.harry - * ; \*.33,13 *
```

To see the new device status:

```
=INFO DEVICE $tape
```

The report shows:

```
                  LAST-MODIFIED     OWNER     STATUS      WARNING-MODE
$LPRINT
                22SEP86,  9:22   \*.86,255    THAWED         OFF

        086,001      R,W
        086,003      R,W
        086,008      R,W
     \*.033,013      R,W,      O
     \*.086,255      R,W
        255,*        R,W
```

DELETE DEVICE and SUBDEVICE Commands

DELETE DEVICE or SUBDEVICE deletes an authorization record. After an authorization record is deleted, the device or subdevice is no longer subject to Safeguard authorization checks or to Safeguard auditing.

An owner of a device, the primary owner’s group manager, and the super ID can delete a device authorization record. Similarly, an owner of a subdevice’s authorization record, the primary owner’s group manager, and the super ID can delete a subdevice authorization record.

DELETE DEVICE *device-list* [[,] WHERE WARNING-MODE]

DELETE SUBDEVICE *subdevice-list* [[,] WHERE WARNING-MODE] *device-list*

specifies one or more devices for which authorization records are to be deleted. *device-list* can be either:

```
  device-name

( device-name [ , device-name ] ... )
```

device-name

can be any device name. The name can contain wild-card characters.

subdevice-list

specifies one or more subdevices for which authorization records are to be deleted. *subdevice-list* can be either:

```
  subdevice-name

( subdevice-name [ , subdevice-name ... ] )
```

subdevice-name

can be any subdevice name. The name can contain wild-card characters.

`WHERE WARNING-MODE`

specifies that only devices or subdevices in *filename-list* that have WARNING-MODE set are to be deleted.

Example

The owner of the authorization record for a device $LASER enters this command to delete the Safeguard authorization record for a device:

```
=DELETE DEVICE $laser
```

FREEZE DEVICE and SUBDEVICE Commands

FREEZE DEVICE or SUBDEVICE temporarily suspends the access authorities granted to users on an ACL. While a device or subdevice is frozen, only the owners of the authorization record, the primary owner’s group manager, and the local super ID can access the device or subdevice.

An owner of a device, the primary owner’s group manager, and the super ID can freeze a device authorization record. Similarly, an owner of a subdevice’s authorization record, the primary owner’s group manager, and the super ID can freeze a subdevice authorization record.

Use THAW DEVICE or SUBDEVICE to restore all the access authorities granted to users on the ACL before access was frozen.

```
FREEZE DEVICE device-list [ [ , ] WHERE WARNING-MODE]


FREEZE SUBDEVICE subdevice-list [ [ , ] WHERE WARNING-MODE]
```

device-list

specifies one or more devices for which access is to be frozen. *device-list* can be either:

device-name

(*device-name* [, *device-name*] ...)

device-name

can be any device name. The name can contain wild-card characters.

subdevice-list

specifies one or more subdevices for which access is to be frozen. *subdevice-list* can be either:

subdevice-name

(*subdevice-name* [, *subdevice-name* ...])

subdevice-name

can be any subdevice name. The name can contain wild-card characters.

WHERE WARNING-MODE

specifies that only devices or subdevices in *filename-list* that have WARNING-MODE set are to be frozen.

Considerations

- Freezing a device or subdevice that is currently open has no effect on processes that have the device or subdevice open. However, if a process attempts to reopen the device or subdevice after closing it, the Safeguard software returns a security violation error (file error 48).
- While a device or subdevice is frozen, the owners of the authorization record and the primary owner’s group manager have all the access authorities (read, write, and owner) for that device or subdevice.

  The local super ID also retains ownership and has all the authority of any user or group manager unless explicitly denied.

Example

The owner of the authorization record for the device $TTYP enters this command to suspend access to the device:

```
=FREEZE DEVICE $ttyp
```

INFO DEVICE and SUBDEVICE Commands

INFO DEVICE and SUBDEVICE displays the attribute values currently stored in an authorization record. INFO DEVICE and SUBDEVICE produces two types of reports: brief and detailed. The formats for the two report types are illustrated following the syntax.

Any user can produce an INFO report on any protected device or subdevice.

```
INFO [ / OUT listfile / ] DEVICE  device-list
   [ [ , ] DETAIL ]


INFO [ / OUT listfile / ] SUBDEVICE  subdevice-list

   [ [ , ] DETAIL ]
```

OUT *listfile*

directs the INFO DEVICE or SUBDEVICE report to *listfile*. After executing the INFO command, SAFECOM redirects device output to the current OUT file.

For *listfile*, specify any file name. SAFECOM opens *listfile* and appends the INFO report to the file. If *listfile* does not exist, SAFECOM creates an EDIT file by that name and writes the INFO report to that file.

device-list

specifies one or more devices for which INFO reports are to be produced. *device-list* can be either:

device-name

(*device-name* [, *device-name*] ...)

device-name

can be any device name. The name can contain wild-card characters.

subdevice-list

specifies one or more subdevices for which INFO reports are to be produced. *subdevice-list* can be either:

subdevice-name

(*subdevice-name* [, *subdevice-name* ...])

subdevice-name

can be any subdevice name. The name can contain wild-card characters.

DETAIL

adds the *audit-spec*s defined for the device or subdevice to the INFO report. For a full description of the four *audit-spec*s, see the SET DEVICE and SUBDEVICE Commands on page 10-26.

INFO DEVICE and SUBDEVICE Brief Report

The brief INFO DEVICE and SUBDEVICE report gives information about devices and subdevices. Figure 10-1 shows the format of the brief INFO DEVICE report. The INFO SUBDEVICE format is identical except that the report gives the name of the subdevice rather than a device.

Figure 10-1. INFO DEVICE Brief Report Format

```
                 LAST-MODIFIED OWNER    STATUS      WARNING-MODE
$device
                  date, time   owner-id status        {ON|OFF}

    user-spec [DENY] authority-list
    user-spec [DENY] authority-list
       .
       .
       .
 [ NO ACCESS CONTROL LIST DEFINED! ]
```

Figure 10-1 contains these device attribute values and status fields:

$device

is the name of the device or subdevice whose existing attribute values are being displayed.

date, time

is the date and time of the last change made to this authorization record. *date* and *time* are in local civil time.

owner-id

is the user ID of the user who owns this authorization record.

status

is the current status of this device or subdevice. *status* is either FROZEN or THAWED.

WARNING-MODE
{ON|OFF}

is the current warning-mode state of this device or subdevice. ON indicates that the protection record is in warning mode. The initial value is OFF, which indicates that warning mode is disabled for this device or subdevice.

user-spec [DENY] *authority-list*

is an entry in the ACL defined for this device or subdevice. *user-spec* identifies a single user or user group. *authority-list* is a list of single-character codes that represent the access authorities granted to the user or user group identified by *user-spec*. DENY indicates that the access authorities specified with *authority-list* are specifically denied to the user or user group identified by *user-spec*.

user-spec can be any of:

```
group-num , member-num
group-num,*
*,*
\node-spec.group-num , member-num
\node-spec.group-num,*
\node-spec.*,*
```

group-num, *member-num* identifies a single local user.

group-num,* identifies all the local users in the group that has *group-num*.

, identifies all the local users at the node to which this device or subdevice is attached.

\node-spec.*group-num*, *member-num* identifies both the local user who has user ID *group-num*, *member-num* and a network user who has the same user name and user ID as that local user.

\node-spec.*group-num*,* identifies all the local users in the group identified by *group-num* and all network users whose *group-num* and *group-name* match those of the local group.

\node-spec.*,* identifies all local users on the node to which the device or subdevice is attached as well as all network users who have access to the node.

authority-list can contain these codes:

```
R - READ authority
W - WRITE authority
O - OWNER authority
```

`[ NO ACCESS CONTROL LIST DEFINED! ]`

appears for a device or subdevice that has no ACL. Use ALTER DEVICE...ACCESS or ALTER SUBDEVICE...ACCESS to define ACL entries for an existing authorization record. Only the local super ID can access a device or subdevice for which no ACL is defined.

INFO DEVICE and SUBDEVICE Detailed Report

The detailed INFO DEVICE or SUBDEVICE report includes the auditing specifications currently defined for the protected device or subdevice. The format of the detailed INFO DEVICE report appears in Figure 10-2. The format of the detailed INFO SUBDEVICE report is identical except for the subdevice name in place of the device name.

Figure 10-2. INFO DEVICE Detailed Report Format

```
                 LAST-MODIFIED OWNER    STATUS     WARNING-MODE
$device
                  date, time   owner-id status       {ON|OFF}

    user-spec [DENY] authority-list
    user-spec [DENY] authority-list
       .
       .
       .
 [ NO ACCESS CONTROL LIST DEFINED! ]

 OBJECT-TEXT-DESCRIPTION =

 AUDIT-ACCESS-PASS = a-spec  AUDIT-MANAGE-PASS = a-spec
 AUDIT-ACCESS-FAIL = a-spec  AUDIT-MANAGE-FAIL = a-spec
```

In addition to the device attribute values displayed in the brief INFO DEVICE report, the detailed INFO DEVICE report displays these attribute values:

```
AUDIT-ACCESS-PASS = a-spec  AUDIT-MANAGE-PASS = a-spec
AUDIT-ACCESS-FAIL = a-spec  AUDIT-MANAGE-FAIL = a-spec
```

These values indicate the conditions under which the Safeguard software audits attempts to open this device or subdevice and to manage this authorization record. *a-spec* can be:

`{ ALL | LOCAL | REMOTE | NONE }`

For a full description of each *a-spec*, see the appropriate *audit-spec* in SET DEVICE and SUBDEVICE Commands on page 10-26.

Example

A sample brief INFO DEVICE report for a line printer follows:

```
=INFO DEVICE $lprint
```

```
                  LAST-MODIFIED     OWNER      STATUS     WARNING-MODE
$LPRINT
                18AUG86, 17:28   \*.86,255    THAWED          OFF

        086,002 DENY R,W
        033,*        R,W
        086,*        R,W
        255,*        R,W
```

This report gives these information:

- The owner of this device authorization record is a network user who is the manager for group 86 (with user ID 86,255).
- All users who are members of group number 33 or 255 are granted both READ and WRITE authority for the device $LPRINT.
- All users who are members of group number 86, with one exception, are granted both READ and WRITE authority for $LPRINT. User ID 86,2 is specifically denied both READ and WRITE authority.

RESET DEVICE and SUBDEVICE Commands

RESET DEVICE or SUBDEVICE resets the current default attribute values to their predefined values.

When you add an authorization record, the current default attribute values are used for any attributes you do not specify with the ADD DEVICE or SUBDEVICE command. To set the default attribute values to specific values, use SET DEVICE or SUBDEVICE.

```
RESET DEVICE [ [ , ] device-attribute-keyword ]

   [ , device-attribute-keyword ] ...


RESET SUBDEVICE [ [ , ] device-attribute-keyword ]

   [ , device-attribute-keyword ] ...
```

device-attribute-keyword

sets the current default value of the specified attribute to its predefined value. The *device-attribute-keyword*s and their predefined values are:

OWNER	The user ID of the current user
ACCESS	Null (no access control list)
OBJECT-TEXT-DESCRIPTION	Null (no descriptive text or blank)
AUDIT-ACCESS-PASS	NONE (no auditing)

```
AUDIT-ACCESS-FAIL        NONE (no auditing)
AUDIT-MANAGE-PASS        NONE (no auditing)
AUDIT-MANAGE-FAIL        NONE (no auditing)
WARNING-MODE            OFF (warning mode disabled)
```

For a complete description of the *device-attribute*s, see the SET DEVICE and SUBDEVICE Commands on page 10-26.

Consideration

If you enter RESET DEVICE or RESET SUBDEVICE (or RESET when the assumed object type is DEVICE or SUBDEVICE) and you do not include any *device-attribute-keyword*, all the attributes are returned to their predefined values. The predefined values are listed before in the syntax for RESET DEVICE or SUBDEVICE.

Example

In this example, a RESET DEVICE command restores only the current default device ACL to its predefined value (that is, no ACL):

To display the current device status:

```
=SHOW DEVICE
```

The report shows:

```
TYPE          OWNER     WARNING-MODE
 DEVICE    \*.86,255        OFF

  OBJECT-TEXT-DESCRIPTION =

  AUDIT-ACCESS-PASS = ALL        AUDIT-MANAGE-PASS = REMOTE
  AUDIT-ACCESS-FAIL = NONE       AUDIT-MANAGE-FAIL = ALL

         255,255      R,W
      \*.086,255      R,W
         086,*        R,W
```

To restore the default ACL:

```
=RESET DEVICE ACCESS
```

To display the reset device status:

```
=SHOW DEVICE
```

The report shows:

```
TYPE         OWNER     WARNING-MODE
 DEVICE    \*.86,255      OFF

  OBJECT-TEXT-DESCRIPTION =

  AUDIT-ACCESS-PASS = ALL        AUDIT-MANAGE-PASS = REMOTE
  AUDIT-ACCESS-FAIL = NONE       AUDIT-MANAGE-FAIL = ALL

  NO ACCESS CONTROL LIST DEFINED!
```

SET DEVICE and SUBDEVICE Commands

SET DEVICE or SUBDEVICE establishes default values for one or more device attributes. When you add an authorization record, the default attribute values are used for any attributes you do not specify in your ADD DEVICE or SUBDEVICE command.

To display the current default values for the attribute, use the SHOW DEVICE or SUBDEVICE command.

```
SET DEVICE [ , ] { LIKE device-name | device-attribute }

   [ , device-attribute ] ...


SET SUBDEVICE [ , ] { LIKE subdevice-name | device-attribute
}

   [ , device-attribute ] ...
```

LIKE *device-name*

sets the current default *device-attribute* values to the same as the existing values for *device-name*.

device-name

identifies a device whose existing attribute values are to be the current default *device-attribute* values. *device-name* can be any device name.

LIKE *subdevice-name*

sets the current default *device-attribute* values to the same as the existing values for *subdevice-name*.

subdevice-name

identifies a subdevice whose existing attribute values are to be the current default *device-attribute* values. *subdevice-name* can be any subdevice name.

device-attribute

defines a default value for the specified device attribute. The *device-attribute*s are:

```
OWNER [owner-id]
ACCESS access-spec [ ; access-spec ] ...
OBJECT-TEXT-DESCRIPTION "[any-text]"
AUDIT-ACCESS-PASS [audit-spec]
AUDIT-ACCESS-FAIL [audit-spec]
AUDIT-MANAGE-PASS [audit-spec]
AUDIT-MANAGE-FAIL [audit-spec]
WARNING-MODE {ON|OFF}
```

```
OWNER [owner-id]
```

specifies the owner of an authorization record for a device or subdevice. *owner-id* can be either of the following:

```
[\node-spec.]group-name.member-name
[\node-spec.]group-num , member-num
```

If you omit *owner-id*, *owner-id* is set to your user ID (the user ID of the current user).

```
ACCESS access-spec [ ; access-spec ] ...
```

changes the ACL for *filename-list* by adding or deleting ACL entries or by changing the authority list of a current ACL entry.

An ACL contains as many as 50 entries that grant or deny access authorities to users and user groups.

access-spec has the following form:

```
user-list  [-] [DENY] authority-list
```

```
group-list [-] [DENY] authority-list
```

user-list

specifies users who are granted (or denied) the access authorities specified with the following *authority-list*. *user-list* can be either:

```
  net-user-spec
```

```
( net-user-spec [ , net-user-spec ] ... )
```

net-user-spec can be any of:

```
[\node-spec.]adm-group-name.user-name
[\node-spec.]adm-group-num , user-num
[\node-spec.]adm-group-name.*
[\node-spec.]adm-group-num , *
[\node-spec.]*.*
[\node-spec.]*,*
```

-

(minus-sign) operates on existing ACL entries. The minus-sign form of *access-spec* modifies the current default ACL. The *authority* entries are removed from the default ACL entries for the users specified with *user-list*.

group-list

can be either of:

 net-group-spec

(*net-group-spec* [, *net-user-spec*] ...)

net-group-spec can be any of:

GROUP [NAME][\node-spec.] *group-name*

GROUP NUMBER [\node-spec.]

node-spec

takes this form:

* | *node-name* | *node-number*

node-name

specifies the system name.

node-number

specifies the Expand node number.

adm-group-name

specifies the name of the administrative group.

adm-group-num

specifies the group number of an administrative group.

group-name

specifies the name of any group.

group-num

specifies the group number of any group.

-

(minus-sign) operates on existing ACL entries. The minus-sign form of *access-spec* modifies the current default ACL. The *authority* entries are removed from the default ACL entries for the users specified with *user-list*.

DENY

denies the users or user groups specified in the *user-list* the access authorities specified in the *authority-list*.

authority-list

specifies the access authorities to be granted (or denied) to *user-list*. *authority-list* can be any of:

```
  authority

( authority [ , authority ] ... )

  *
```

authority

can be any of:

```
R[EAD]
W[RITE]
O[WNER]
```

*

(asterisk) specifies read, write, and owner.

`OBJECT-TEXT-DESCRIPTION "[any-text]"`

allows you to store printable characters as comments. These comments are associated with the objects and are used to manage the object authorization record.

The text description field can accommodate 255 bytes of text data.

Note. The text specified in the text description field overwrites existing data, if any. Also, when LIKE clause is used with SET DEVICE and SUBDEVICE command, the OBJECT-TEXT-DESCRIPTION field is not copied with other object authorization record attributes.

The OBJECT-TEXT-DESCRIPTION attribute is supported only on systems running J06.05 and later J-series RVUs and H06.16 and later H-series RVUs and G06.32 and later G-series RVUs.

`AUDIT-ACCESS-PASS [`*audit-spec*`]`

establishes an *audit-spec* for successful attempts to access a device or subdevice. This *audit-spec* specifies the conditions under which an audit record is written to the audit file when a device or subdevice is successfully accessed.

The form of *audit-spec* is:

```
{ ALL | LOCAL | REMOTE | NONE }
```

`ALL`

All successful attempts to access the device or subdevice are audited.

`LOCAL`

Only successful attempts by local users to access the device or subdevice are audited.

`REMOTE`

Only successful attempts by remote users to access the device or subdevice audited.

`NONE`

No successful attempts to access the device or subdevice are audited.

Omitting *audit-spec* specifies NONE.

`AUDIT-ACCESS-FAIL [`*audit-spec*`]`

establishes an *audit-spec* for unsuccessful attempts to access a device or subdevice. This *audit-spec* specifies the conditions under which an audit record is written to the audit file when an attempt to access a device or subdevice fails.

The form of *audit-spec* is:

```
{ ALL | LOCAL | REMOTE | NONE }
```

`ALL`

All unsuccessful attempts to access the device or subdevice are audited.

`LOCAL`

Only unsuccessful attempts by local users to access the device or subdevice are audited.

`REMOTE`

Only unsuccessful attempts by remote users to access the device or subdevice are audited.

`NONE`

No unsuccessful attempts to access the device or subdevice are audited.

Omitting *audit-spec* specifies NONE.

`AUDIT-MANAGE-PASS [`*audit-spec*`]`

establishes an *audit-spec* for successful attempts to manage an authorization record. This *audit-spec* specifies the conditions under which an audit record is written to the audit file when an authorization record is successfully managed.

The form of *`audit-spec`* is:

```
{ ALL | LOCAL | REMOTE | NONE }
```

`ALL`

All successful management attempts are audited.

`LOCAL`

Only successful management attempts by local users are audited.

`REMOTE`

Only successful management attempts by remote users are audited.

`NONE`

No successful management attempts are audited.

Omitting *`audit-spec`* specifies NONE.

`AUDIT-MANAGE-FAIL [`*`audit-spec`*`]`

establishes an *`audit-spec`* for unsuccessful attempts to manage an authorization record. This *`audit-spec`* specifies the conditions under which an audit record is written to the audit file when an attempt to manage an authorization record fails.

The form of *`audit-spec`* is:

```
{ ALL | LOCAL | REMOTE | NONE }
```

`ALL`

All unsuccessful management attempts are audited.

`LOCAL`

Only unsuccessful management attempts by local users are audited.

`REMOTE`

Only unsuccessful management attempts by remote users are audited.

`NONE`

No unsuccessful management attempts are audited.

Omitting *`audit-spec`* specifies NONE.

`WARNING-MODE { ON | OFF }`

defines whether warning mode is enabled for the specified device or subdevice. The value is required. For more information on warning mode, see the *Safeguard Administrator's Manual*.

ON enables warning mode for the specified device or subdevice. The initial value is OFF, which disables warning mode for the specified device or subdevice.

Example

These commands define default values for a new printer:

```
=ASSUME DEVICE
=SET OWNER prs.manager
=SET AUDIT-ACCESS-PASS all , AUDIT-MANAGE-PASS local
=SET ACCESS 33,* (r,w) ; 86,* *; prs.harry DENY *; 255,* *
```

These default device attribute values are defined here:

- The owner of the authorization record for the device is the PRS manager.
- The Safeguard software audits successful access of the device and successful local management of the device’s authorization record.
- All members of groups 33, 86, and 255 can read and write to the device except PRS.HARRY, who is specifically denied access.
- All members of group 86 and 255 have OWNER authority except PRS.HARRY, who is specifically denied all authorities.

SHOW DEVICE and SUBDEVICE Commands

SHOW DEVICE or SUBDEVICE displays the current default values for the attributes.

```
SHOW [ / OUT listfile / ] { DEVICE | SUBDEVICE }
```

`OUT` *`listfile`*

directs the SHOW PROCESS report to *`listfile`*. After you execute the SHOW command, SAFECOM redirects its output to the current OUT file.

For *`listfile`*, specify any file name. SAFECOM opens *`listfile`* and appends the SHOW DEVICE or SUBDEVICE report to the file. If *`listfile`* does not exist, SAFECOM creates an EDIT file and then writes the SHOW DEVICE or SUBDEVICE report to that file.

SHOW DEVICE and SUBDEVICE Report Format

The SHOW DEVICE command displays the device attributes and their current default values in the format shown in Figure 10-3. The SHOW SUBDEVICE command output is identical, except for the word SUBDEVICE in place of the word DEVICE.

Figure 10-3. SHOW DEVICE Report Format

```
TYPE                 OWNER          WARNING-MODE
 DEVICE             gn,un             {ON|OFF}

  OBJECT-TEXT-DESCRIPTION =

  AUDIT-ACCESS-PASS = a-spec   AUDIT-MANAGE-PASS = a-spec
  AUDIT-ACCESS-FAIL = a-spec   AUDIT-MANAGE-FAIL = a-spec

     user-spec [DENY] authority-list
     user-spec [DENY] authority-list
          .        .         .
          .        .         .
  [ NO ACCESS CONTROL LIST DEFINED! ]
```

The SHOW DEVICE report displays these attribute values:

`OWNER gn, un`

is the user ID (group number and member number) of the user who will own this authorization record if a device or subdevice with these attribute values is added to Safeguard protection.

`WARNING-MODE`
`{ON|OFF}`

is the current warning-mode state of this device or subdevice. ON indicates that the protection record is in warning mode. The initial value is OFF, which indicates that warning mode is disabled for this device or subdevice.

`AUDIT-ACCESS-PASS = a-spec AUDIT-MANAGE-PASS = a-spec`
`AUDIT-ACCESS-FAIL = a-spec AUDIT-MANAGE-FAIL = a-spec`

are the conditions under which the Safeguard software will audit attempts to access this device or subdevice and attempts to manage this authorization record. For more information about these fields for *audit-spec*, see the SET DEVICE and SUBDEVICE Commands on page 10-26.

`user-spec [DENY] authority-list`

is a current default ACL entry for devices. For a full description, see INFO DEVICE and SUBDEVICE Commands on page 10-20.

`[ NO ACCESS CONTROL LIST DEFINED! ]`

indicates no default ACL entries are defined. Use SET DEVICE...ACCESS or SET SUBDEVICE...ACCESS to define default ACL entries. You can use ADD

DEVICE...ACCESS or ADD SUBDEVICE...ACCESS to define ACL entries when you create an authorization record.

△ **Caution.** If you do not specify an ACL, only the local super ID can access the device or subdevice.

Example

This SHOW DEVICE report displays the current default device attribute values for a device:

```
=SHOW DEVICE
```

```
TYPE          OWNER        WARNING-MODE
 DEVICE     255,18             OFF

 OBJECT-TEXT-DESCRIPTION =

 AUDIT-ACCESS-PASS = ALL        AUDIT-MANAGE-PASS = NONE
 AUDIT-ACCESS-FAIL = ALL        AUDIT-MANAGE-FAIL = NONE

        033,013      R,W
        033,255      R,W
        255,018      R,W
```

These current default values indicate these:

- The owner of the authorization record for a device that has these attribute values is the local super-group user with user ID 255,18.
- The Safeguard software audits all successful and unsuccessful attempts to access a device that has these attribute values.
- The users IDs 33,13 and 255,18 as well as the group 33 manager can read and write to a device that has these attribute values.

THAW DEVICE and SUBDEVICE Commands

THAW DEVICE or SUBDEVICE restores the ACL for a frozen device or subdevice. After a frozen device or subdevice is thawed, users who were granted access authority by the ACL can again access the device or subdevice.

An owner of the authorization record, the primary owner’s group manager, and the super ID can thaw a frozen device or subdevice.

THAW DEVICE or SUBDEVICE has no effect on a device or subdevice that is not frozen.

```
THAW DEVICE device-list [ [ , ] WHERE WARNING-MODE ]


THAW SUBDEVICE subdevice-list [ [ , ] WHERE WARNING-MODE ]
```

`device-list`

specifies one or more devices that are to be thawed. *`device-list`* can be either:

```
  device-name

( device-name [ , device-name ] ... )
```

`device-name`

can be any device name. The name can contain wild-card characters.

`subdevice-list`

specifies one or more subdevices that are to be thawed. *`subdevice-list`* can be either:

```
  subdevice-name

( subdevice-name [ , subdevice-name ... ] )
```

`subdevice-name`

can be any subdevice name. The name can contain wild-card characters.

`WHERE WARNING-MODE`

specifies that only devices or subdevices in *`filename-list`* that have WARNING-MODE set are to be thawed.

Example

The owner of the authorization record for the device $TAPE2 thaws the ACL for the device by entering:

```
=THAW DEVICE $tape2
```

11

Process and Subprocess Security Commands

With the SAFECOM process and subprocess security commands, any user can assume ownership of a process name by adding an authorization record for that name to the Safeguard object database. After an authorization record is added for a name, all attempts to access a process or subprocess that has the protected name are subject to Safeguard authorization checks and, optionally, to Safeguard access auditing. (You can use OBJECTTYPE PROCESS and SUBPROCESS to restrict this behavior. For more information, see Section 12, OBJECTTYPE Security Commands.)

To control access to the process name, the owner of an authorization record can create an access control list, (ACL). Accessing a process name includes creating a process that has the protected process name, opening a process that runs with the protected name, and stopping a process that runs with the protected name. The owner of a process name authorization record can also specify when the Safeguard software should audit attempts to access the process name.

This section begins with a brief overview of the Safeguard access control features for processes and subprocesses and summarizes the process security commands. Following the command summary, the process and subprocess security commands are described in detail.

Process and Subprocess Security

In a system protected by the Safeguard software, process security consists of access control for two entities:

- Program object disk files. To create a process that runs a program object disk file under standard Guardian security, a user must have EXECUTE authority for that object disk file. The owner of a protected program object file can use the SAFECOM disk file security commands to control users’ ability to run the object file.
- Process names. When a process is started, you can specify that the process run with a process name (by including the NAME option in the RUN command). If that process name is protected by the Safeguard software, the resulting process is subject to Safeguard access controls.

Process and Subprocess Access Authorities

The ACL for a process name can grant any combination of these access authorities to users and user groups:

READ Open a process or subprocess with a protected name for input operations.

WRITE Open a process or subprocess with a protected name for output operations.

CREATE Create a process with a protected name. (A user must also have EXECUTE authority for the program object disk file to execute the process.) Not applicable to subprocesses.

PURGE Stop a process with a protected name. Not applicable to subprocesses.

OWNER Manage the authorization records.

Creating a Process With a Protected Name

When a user attempts to create a process that has a protected process name, the Safeguard software checks the ACL for that process name to determine that the user has CREATE authority for the process name. If the user does have CREATE authority, the Safeguard software allows the process to be created. If the user does not have CREATE authority, the user’s process-creation request is rejected with a security violation error (file error 48).

Access to a process or subprocess that is running under a protected name is controlled by the ACL defined for that process or subprocess name.

Opening a Process or Subprocess That Has a Protected Name

When a process attempts to open another process or subprocess running under a protected name, the Safeguard software checks the ACL for the protected process name to determine whether the appropriate authority is granted to the user identified by the process accessor ID (PAID) of the process requesting the open. If the open request is for read and the user has READ authority, the request is allowed to complete successfully. If the open request is for write and the user has WRITE authority, the request is allowed to complete successfully. If the user identified by the PAID of the opening process does not have the proper authority, the Safeguard software rejects the open request with a security violation error (file error 48). For more information on process and creator accessor IDs, see the *Security Management Guide*.

If the process is opened for read, the file system allows both read and write operations. Therefore, the process itself must enforce the distinction between an open for read and an open for write.

The Safeguard software distinguishes between local and remote open requests. A remote open request is one made by a process that was created by a network user logged on to a remote system.

If a process is remote with respect to the process or subprocess that it is attempting to open, the opener's PAID must identify a network user who has been granted remote access to the process or subprocess. Otherwise, the open request is rejected with a security violation (file error 48).

For example, suppose a remote process with a PAID of 4,5 attempts to open a process running under a protected name. The ACL defined for the process running under a protected name must grant READ or WRITE authority to *.4,5, *.4,*, or *.*,*. Otherwise, the Safeguard software rejects the open request with a security violation (file error 48).

An open request that has passed a Safeguard authorization check can nevertheless fail. For example, if a process attempts to open a process that is already opened by another process that has exclusive access, the open attempt fails with file error 12 (file in use). For more information, see the *Guardian Procedure Calls Reference Manual*.

Stopping a Process With a Protected Name

If a user attempts to stop a process that is running under a protected name, the Safeguard software checks the ACL for the process name to determine whether the user has PURGE authority. If the user has PURGE authority, the Safeguard software allows the process to be stopped. If the user does not have PURGE authority, the stop request is rejected with a security violation error (file error 48). However, the user who created the process is allowed to stop the process even if an ACL is present that prevents the user from doing the same.

If you create the special NAMED and UNNAMED process protection records, certain users can be given PURGE authority for all named or unnamed processes. A user is allowed to stop any process it started as long as the process is still running under that ID. For more information, see Special NAMED and UNNAMED Process Protection Records on page 11-4.

Process and Subprocess Ownership

A process or subprocess has no authorization record until it is placed under Safeguard control. By default, any user can add a process or subprocess authorization record. For more information on how to restrict who can add process and subprocess authorization records, see PROCESS on page 12-2 or SUBPROCESS on page 12-2. Every authorization record has an OWNER attribute that contains the user ID that can manage the Safeguard access controls for the process or subprocess.

However, the user who adds the record can set the OWNER attribute to the user ID of any user (by including an OWNER specification in a SET PROCESS or SUBPROCESS or ADD PROCESS or SUBPROCESS command). Thus the owner of a process or subprocess need not be the user who added the record. The owner of a protected process or subprocess authorization record can also transfer ownership to another user by changing the OWNER attribute with the ALTER PROCESS or ALTER SUBPROCESS command.

Because the primary owner can add owners to an ACL, that individual can specify additional ownership by the OWNER authority code for ACL entries. Such OWNER authority is an independent extension of the primary owner. Additional owners can do anything that the primary owner is permitted to do. They are equal, in every way, to the primary owner. For example, they can modify the Safeguard authorization records of any process or subprocess for which they own the authorization record, and they can access any process for which they own the authorization record when that process or subprocess has been frozen.

Any user with OWNER authority on the ACL can explicitly deny a local super ID any of the authorities (including OWNER) implicitly granted to that user ID and have this denial actively enforced all of the time.

OWNER authority can be specified for all protected processes. The OWNER authority is always included whenever the * authority code is used. It can also be abbreviated as O.

With the Safeguard software, the owner of a process or subprocess can also be defined as a network user. A network user who owns an authorization record can use the Safeguard software from a remote node to control access to that process or subprocess (provided the user has remote passwords set up between the two systems).

Special NAMED and UNNAMED Process Protection Records

The process security commands allow you to create two special protection records that control who can create or stop any named or unnamed process regardless of Safeguard protection. When you create a protection record specifying NAMED as the process name, that record applies to all named processes. Similarly, UNNAMED applies to all unnamed processes. This feature is intended to allow a special group of users, such as system operators, the ability to create or stop any process.

If Safeguard's global configuration DIRECTION-PROCESS attribute has the value PROCESS-FIRST, then NAMED and UNNAMED protection records are checked first (first named/unnamed, then process, then subprocess). This is a top-down evaluation direction.

If Safeguard's global configuration DIRECTION-PROCESS attribute has the value SUBPROCESS-FIRST, then NAMED and UNNAMED protection records are checked last (first subprocess, then process, then named/unnamed). This is a bottom-up evaluation direction.

If you create the UNNAMED protection record, be aware that no users will be able to create and stop unnamed processes except users specified on the UNNAMED ACL.

If you create the NAMED protection record, it is advisable to create other process protection records.

For NAMED and UNNAMED records, the only valid access authorities are CREATE, PURGE, and OWNER authorities. READ and WRITE authorities are not valid.

If you use these special process protection records, be sure to alter your Safeguard configuration to specify FIRST-RULE for COMBINATION-PROCESS. This configuration is necessary for the NAMED and UNNAMED feature to function as intended.

Process and Subprocess Security Command Summary

Table 11-1 lists the process and subprocess security commands and gives a brief description of each.

Table 11-1. Process and Subprocess Security Command Summary (page 1 of 2)

Command	**Description**
ADD [SUB]PROCESS	Adds a process name authorization record that has the specified process or subprocess attribute values. The current default values for the process name attributes are used for any attributes not specified in the ADD PROCESS or SUBPROCESS command.
ALTER [SUB]PROCESS	Changes one or more attribute values in a process name authorization record. For all attributes except ACCESS, ALTER PROCESS or SUBPROCESS replaces the current value with the specified value. For the ACCESS attribute, ALTER PROCESS or SUBPROCESS changes the existing ACL to incorporate *access-spec*.
DELETE [SUB]PROCESS	Deletes a process name authorization record. Afterward, any user can access a process or subprocess under the deleted process name, and any process or subprocess that is accessed under that name is subject only to the standard Guardian security checks.
FREEZE [SUB]PROCESS	Temporarily suspends access to a protected process name. Only the primary owner of a process name, users with OWNER authority, the owner’s group manager, and the local super ID can create, access, or stop a process that has a frozen process name.
INFO [SUB]PROCESS	Displays the attribute values in a process name authorization record.
RESET [SUB]PROCESS	Sets one or more default values for the process or subprocess attributes to predefined values.

Table 11-1. Process and Subprocess Security Command Summary (page 2 of 2)

Command	Description
SET [SUB]PROCESS	Sets one or more default values for the process attributes to specified values. When a process name authorization record is added, the current default values for the process or subprocess attribute values are used for any attributes not specified in the ADD PROCESS or ADD SUBPROCESS command.
SHOW [SUB]PROCESS	Displays the current default values for the process or subprocess attributes.
THAW [SUB]PROCESS	Restores the access authorities granted to users in the ACL defined for a frozen process name.

Syntax of the Process and Subprocess Security Commands

The rest of this section contains individual syntax descriptions for the SAFECOM process and subprocess security commands. Commands are presented in alphabetical order, and most command descriptions contain these elements:

- A summary of the functions performed by the command, including any restrictions on who can use the command
- The syntax of the command, including descriptions of the command parameters and variables
- The format for the command listing or report (for commands that produce listings or reports)
- Considerations for the use of the command
- Examples of command usage

ADD PROCESS and SUBPROCESS Commands

ADD PROCESS or SUBPROCESS creates a Safeguard authorization record for one or more process names. After a process name authorization record is created, the access control attributes defined for the process name apply to every process that runs under the protected name. All attempts to create or stop a process with the protected name or to open a process or subprocess running under the protected name are subject to Safeguard authorization checks and, optionally, to Safeguard auditing.

Any local user can add a process name authorization record unless it is modified by OBJECTTYPE PROCESS or SUBPROCESS. For more information, see Section 12, OBJECTTYPE Security Commands.

You can use the SET PROCESS or SUBPROCESS command to establish default values for the process or subprocess attributes. You can then use ADD PROCESS or

SUBPROCESS to specify the process name to which the default values are to be applied. You can also specify values for attributes in your ADD PROCESS or SUBPROCESS command. The current default values are used for any attributes not specified in the ADD PROCESS or SUBPROCESS command.

```
ADD PROCESS process name-list [ , ]

   [ LIKE process-name | process-attribute ]

   [ , process-attribute ] ...


ADD SUBPROCESS subprocess name-list [ , ]

   [ LIKE subprocess-name | process-attribute ]

   [ , process-attribute ] ...
```

process name-list

specifies one or more process names for which authorization records are to be added. *process name-list* can be either:

process-name

(*process-name* [, *process-name*] ...)

process-name

can be any process name or one of the special names NAMED and UNNAMED to refer to all named or unnamed processes (respectively). The name cannot contain wild-card characters.

LIKE *process-name*

adopts the existing process name attribute values of *process-name* as the attribute values to be used for the authorization record or records being added.

process-name

identifies the process name whose current *process-attribute* values are to be assigned to the process authorization record or records being added. *process-name* can be any process name or one of the special names NAMED and UNNAMED to refer to any named or unnamed process (respectively).

`subprocess name-list`

specifies one or more subprocesses for which authorization records are to be added. *subprocess name-list* can be either:

```
  subprocess-name

( subprocess-name [ , subprocess-name ... ] )
```

subprocess-name

can be any subprocess name. The name cannot contain wild-card characters.

`LIKE subprocess-name`

adopts the existing process name attribute values of *subprocess-name* as the attribute values to be used for the authorization record or records being added.

subprocess-name

identifies the subprocess name whose current *process-attribute* values are to be assigned to the authorization record or records being added. *subprocess-name* can be any subprocess name.

`process-attribute`

defines an attribute value for the process or subprocess name authorization record or records being added. The *process-attribute*s are:

```
OWNER [owner-id]
ACCESS access-spec [ ; access-spec ] ...
OBJECT-TEXT-DESCRIPTION “[any-text]”
AUDIT-ACCESS-PASS [audit-spec]
AUDIT-ACCESS-FAIL [audit-spec]
AUDIT-MANAGE-PASS [audit-spec]
AUDIT-MANAGE-FAIL [audit-spec]
WARNING-MODE {ON|OFF}
```

`OWNER [owner-id]`

specifies the new owner of the authorization record. *owner-id* can be either:

```
[\node-spec.]group-name.member-name
[\node-spec.]group-num , member-num
```

If you omit *owner-id*, *owner-id* is set to your user ID.

`ACCESS access-spec [ ; access-spec ] ...`

changes the ACL for *filename-list* by adding or deleting ACL entries or by changing the authority list of a current ACL entry.

An ACL contains as many as 50 entries that grant or deny access authorities to users and user groups.

access-spec has the form:

```
user-list  [-] [DENY] authority-list
group-list [-] [DENY] authority-list
```

user-list

specifies users who are granted (or denied) the access authorities specified with the following *authority-list*. *user-list* can be either:

```
  net-user-spec

( net-user-spec [ , net-user-spec ] ... )
```

net-user-spec can be any of:

```
[\node-spec.]adm-group-name.user-name
[\node-spec.]adm-group-num , user-num
[\node-spec.]adm-group-name.*
[\node-spec.]adm-group-num , *
[\node-spec.]*.*
[\node-spec.]*,*
```

-

(minus-sign) operates on existing ACL entries. The minus-sign form of *access-spec* modifies the current default ACL. The *authority* entries are removed from the default ACL entries for the users specified with *user-list*.

group-list

can be either:

```
  net-group-spec

( net-group-spec [ , net-user-spec ] ... )
```

net-group-spec

can be any of:

```
GROUP [NAME][\node-spec.] group-name

GROUP NUMBER [\node-spec.]
```

node-spec

takes this form:

```
* | node-name | node-number
```

node-name

specifies the system name.

node-number

specifies the Expand node number.

adm-group-name

specifies the name of the administrative group.

adm-group-num

specifies the group number of an administrative group.

group-name

specifies the name of any group.

group-num

specifies the group number of any group.

-

(minus-sign) operates on existing ACL entries. The minus-sign form of *access-spec* modifies the current default ACL. The *authority* entries are removed from the default ACL entries for the users specified with *user-list*.

Note. Specifying ACCESS *access-spec* with ADD PROCESS or SUBPROCESS does not override the current default ACL (established with SET PROCESS or PROCESS). Instead, any ACL entries specified with ADD PROCESS or SUBPROCESS are added to the current default ACL. Then the entire ACL is defined for the process or subprocess name whose authorization record is being added.

DENY

denies the users or user groups specified with *user-list* the access authorities specified with *authority-list*.

authority-list

specifies the access authorities to be granted (or denied) to the user or users specified with *user-list*. *authority-list* can be any of:

```
  authority

( authority [ , authority ] ... )

  *
```

authority

is any one of:

```
R[EAD]
W[RITE]
C[REATE]
P[URGE]
O[WNER]
```

R and W are not valid for NAMED and UNNAMED processes.

C and P are not valid for subprocesses.

*

(asterisk) specifies all the process authorities (R, W, C, P, and O).

`OBJECT-TEXT-DESCRIPTION “[any-text]”`

allows you to store printable characters as comments. These comments are associated with the objects and are used to manage the object authorization record.

The text description field can accommodate 255 bytes of text data.

Note. The text specified in the text description field overwrites existing data, if any. Also, when LIKE clause is used with ADD PROCESS and SUBPROCESS command, the OBJECT-TEXT-DESCRIPTION field is not copied with other object authorization record attributes.

The OBJECT-TEXT-DESCRIPTION attribute is supported only on systems running J06.05 and later J-series RVUs and H06.16 and later H-series RVUs and G06.32 and later G-series RVUs.

`AUDIT-ACCESS-PASS [`*`audit-spec`*`]`

changes the *audit-spec* for successful attempts to access the process or subprocess name. The form of *audit-spec* is:

```
{ ALL | LOCAL | REMOTE | NONE }
```

For a description of each *audit-spec*, see the SET PROCESS and SUBPROCESS Commands on page 11-28. Omitting *audit-spec* specifies NONE.

`AUDIT-ACCESS-FAIL [`*`audit-spec`*`]`

changes the *audit-spec* for unsuccessful attempts to access the process or subprocess name. The form of *audit-spec* is:

```
{ ALL | LOCAL | REMOTE | NONE }
```

For a description of each *`audit-spec`*, see the SET PROCESS and SUBPROCESS Commands on page 11-28. Omitting *`audit-spec`* specifies NONE.

`AUDIT-MANAGE-PASS [`*`audit-spec`*`]`

changes the *`audit-spec`* for successful attempts to manage (change or read) this authorization record. The form of *`audit-spec`* is:

```
{ ALL | LOCAL | REMOTE | NONE }
```

For a description of each *`audit-spec`*, see the SET PROCESS and SUBPROCESS Commands on page 11-28. Omitting *`audit-spec`* specifies NONE.

`AUDIT-MANAGE-FAIL [`*`audit-spec`*`]`

changes the *`audit-spec`* for unsuccessful attempts to manage (change or read) this authorization record. The form of *`audit-spec`* is:

```
{ ALL | LOCAL | REMOTE | NONE }
```

For a description of each *`audit-spec`*, see the SET PROCESS and SUBPROCESS Commands on page 11-28. Omitting *`audit-spec`* specifies NONE.

`WARNING-MODE { ON | OFF }`

defines whether warning mode is enabled for the specified process or subprocess. The value is required. For details on warning mode, see the *Safeguard Administrator's Manual*.

ON enables warning mode for the specified process or subprocess. The initial value is OFF, which disables warning mode for the specified process or subprocess.

Consideration

Any attribute specifications in an ADD PROCESS or SUBPROCESS command affect only the record or records being added, but do not alter the default attribute values. This condition is also true for a LIKE clause in an ADD PROCESS or SUBPROCESS command.

Example

A user adds an authorization record for the process name $ADD:

```
=ADD PROCESS $add,OBJECT-TEXT-DESCRIPTION “Process added”,&
ACCESS \*.33,* (r,w,c); 255,* *
```

The ACCESS attribute in this ADD PROCESS command grants READ, WRITE, and CREATE authorities to all network users who are members of group 33 and grants all access authorities (READ, WRITE, CREATE, PURGE, and OWNER) to all members of the super group.

ALTER PROCESS and SUBPROCESS Commands

ALTER PROCESS or SUBPROCESS changes one or more attribute values in an authorization record.

An owner of the authorization record, the primary owner’s group manager, and the super ID can change the attribute values defined for a process or subprocess name.

Except for the ACCESS attribute, specifying a new attribute value in ALTER PROCESS or SUBPROCESS replaces the current attribute value with the specified value.

Using ALTER PROCESS or SUBPROCESS to specify a new ACCESS *access-spec* adds the new *access-spec* to the current ACL. To remove existing authorities granted to users, use the minus-sign (-) form of *access-spec*.

```
ALTER PROCESS process name-list [ , ]

   { LIKE process-name | process-attribute }

   [ , process-attribute ] ...


ALTER SUBPROCESS subprocess name-list [ , ]

   { LIKE subprocess-name | process-attribute }

   [ , process-attribute ] ...
```

process name-list

specifies one or more process names for which authorization records are to be changed. *process name-list* can be either:

```
  process-name

( process-name [ , process-name ] ... )
```

process-name

can be any process name or one of the special names NAMED and UNNAMED to refer to all named or unnamed processes (respectively). The name can contain wild-card characters.

LIKE *process-name*

adopts the existing attribute values of *process-name* as the attribute values to be used for the authorization record or records being changed. For the ACCESS attribute, LIKE adds ACL entries or authorities only to existing entries. It does not replace or delete ACL entries or authorities.

process-name

identifies the process name whose current *process-attribute* values are to be assigned to the process authorization record or records being altered. *process-name* can be any process name or one of the special names NAMED and UNNAMED to refer to any named or unnamed process (respectively).

subprocess name-list

specifies one or more subprocesses for which authorization records are to be changed. *subprocess name-list* can be either:

```
  subprocess-name

( subprocess-name [ , subprocess-name ... ] )
```

subprocess-name

can be any subprocess name. The name can contain wild-card characters.

LIKE *subprocess-name*

adopts the existing attribute values of *subprocess-name* as the attribute values to be used for the authorization record or records being changed. For the ACCESS attribute, LIKE adds ACL entries or authorities only to existing entries. It does not replace or delete ACL entries or authorities.

subprocess-name

identifies the subprocess name whose current *process-attribute* values are to be assigned to the authorization record or records being changed. *subprocess-name* can be any subprocess name.

process-attribute

changes the value of the specified process or subprocess attribute. The *process-attribute*s are:

```
OWNER [owner-id]
ACCESS access-spec [ ; access-spec ] ...
OBJECT-TEXT-DESCRIPTION “[any-text]”
RESET-OBJECT-TEXT-DESCRIPTION
AUDIT-ACCESS-PASS [audit-spec]
AUDIT-ACCESS-FAIL [audit-spec]
AUDIT-MANAGE-PASS [audit-spec]
AUDIT-MANAGE-FAIL [audit-spec]
WHERE WARNING-MODE
WARNING-MODE {ON|OFF}
```

`OWNER [`*`owner-id`*`]`

specifies the new owner of the authorization record for the process or subprocess name. *owner-id* can be either of:

```
[\node-spec.]group-name.member-name
[\node-spec.]group-num , member-num
```

If you omit *owner-id*, *owner-id* is set to your user ID.

`ACCESS` *`access-spec`* `[ ;` *`access-spec`* `] ...`

changes the ACL for *filename-list* by adding or deleting ACL entries or by changing the authority list of a current ACL entry.

An ACL contains as many as 50 entries that grant or deny access authorities to users and user groups.

access-spec has the form:

```
user-list  [-] [DENY] authority-list

group-list [-] [DENY] authority-list
```

user-list

specifies users who are granted (or denied) the access authorities specified with the following *authority-list*. *user-list* can be either:

```
  net-user-spec

( net-user-spec [ , net-user-spec ] ... )
```

net-user-spec can be any of:

```
[\node-spec.]adm-group-name.user-name
[\node-spec.]adm-group-num , user-num
[\node-spec.]adm-group-name.*
[\node-spec.]adm-group-num , *
[\node-spec.]*.*
[\node-spec.]*,*
```

-

(minus-sign) operates on existing ACL entries. The minus-sign form of *access-spec* modifies the current default ACL. The *authority* entries

are removed from the default ACL entries for the users specified with *`user-list`*.

`group-list`

can be either:

`net-group-spec`

`(` *`net-group-spec`* `[ ,` *`net-user-spec`* `] ... )`

`net-group-spec` can be any of:

`GROUP [NAME][\`*`node-spec`*`.]` *`group-nam`*`e`

`GROUP NUMBER [\`*`node-spec`*`.]`

`node-spec`

takes this form:

`*` `|` *`node-name`* `|` *`node-number`*

`node-name`

specifies the system name.

`node-number`

specifies the Expand node number.

`adm-group-name`

specifies the name of the administrative group.

`adm-group-num`

specifies the group number of an administrative group.

`group-name`

specifies the name of any group.

`group-num`

specifies the group number of any group.

-

(minus-sign) operates on existing ACL entries. The minus-sign form of *`access-spec`* modifies the current default ACL. The *`authority`* entries are removed from the default ACL entries for the users specified with *`user-list`*.

`DENY`

denies the users or user groups specified with *`user-list`* the access authorities specified with *`authority-list`*.

`authority-lis``t`

specifies the access authorities to be granted (or denied) to the user or users specified with *`user-list`*. *`authority-list`* can be any of:

```
  authority

( authority [ , authority ] ... )

  *
```

`authority`

can be any of:

```
R[EAD]
W[RITE]
C[REATE]
P[URGE]
O[WNER]
```

R and W are not valid for NAMED and UNNAMED processes.

C and P are not valid for subprocesses.

`*`

(asterisk) specifies all the process authorities (R, W, C, P, and O).

`OBJECT-TEXT-DESCRIPTION “[any-text]”`

allows you to store printable characters as comments. These comments are associated with the objects and are used to manage the object authorization record.

The text description field can accommodate 255 bytes of text data.

Note. The text specified in the text description field overwrites existing data, if any. Also, when LIKE clause is used with ALTER PROCESS and SUBPROCESS command, the OBJECT-TEXT-DESCRIPTION field is not copied with other object authorization record attributes. Also, if you specify OBJECT-TEXT-DESCRIPTION without any text in the quotation marks, the object text description for this record is removed.

The OBJECT-TEXT-DESCRIPTION attribute is supported only on systems running J06.05 and later J-series RVUs and H06.16 and later H-series RVUs and G06.32 and later G-series RVUs.

`RESET-OBJECT-TEXT-DESCRIPTION`

Resets the object description to Null.

Note. The RESET-OBJECT-TEXT-DESCRIPTION attribute is supported only on systems running J06.05 and later J-series RVUs and H06.16 and later H-series RVUs and G06.32 and later G-series RVUs.

`AUDIT-ACCESS-PASS [`*`audit-spec`*`]`

changes the *audit-spec* for successful attempts to access the process or subprocess name. The form of *audit-spec* is:

```
{ ALL | LOCAL | REMOTE | NONE }
```

For a description of each *audit-spec*, see the SET PROCESS and SUBPROCESS Commands on page 11-28. Omitting *audit-spec* specifies NONE.

`AUDIT-ACCESS-FAIL [`*`audit-spec`*`]`

changes the *audit-spec* for unsuccessful attempts to access the process or subprocess name. The form of *audit-spec* is:

```
{ ALL | LOCAL | REMOTE | NONE }
```

For a description of each *audit-spec*, see the SET PROCESS and SUBPROCESS Commands on page 11-28. Omitting *audit-spec* specifies NONE.

`AUDIT-MANAGE-PASS [`*`audit-spec`*`]`

changes the *audit-spec* for successful attempts to manage (change or read) this authorization record. The form of *audit-spec* is:

```
{ ALL | LOCAL | REMOTE | NONE }
```

For a description of each *audit-spec*, see the SET PROCESS and SUBPROCESS Commands on page 11-28. Omitting *audit-spec* specifies NONE.

`AUDIT-MANAGE-FAIL [`*`audit-spec`*`]`

changes the *audit-spec* for unsuccessful attempts to manage (change or read) this authorization record. The form of *audit-spec* is:

```
{ ALL | LOCAL | REMOTE | NONE }
```

For a description of each *audit-spec*, see the SET PROCESS and SUBPROCESS Commands on page 11-28. Omitting *audit-spec* specifies NONE.

`WHERE WARNING-MODE`

specifies that only processes or subprocesses in *`filename-list`* that have WARNING-MODE set are to be altered.

`WARNING-MODE { ON | OFF }`

defines whether warning mode is enabled for the specified process or subprocess. The value is required. For more information on warning mode, see the *Safeguard Administrator's Manual*.

ON enables warning mode for the specified process or subprocess. The initial value is OFF, which disables warning mode for the specified process or subprocess.

Consideration

Using ALTER PROCESS or SUBPROCESS to change one or more attributes for a process or subprocess that is currently running has no effect on any process currently accessing the process or subprocess.

For example, if you change the ACL for a process to deny both READ and WRITE authority to a user who has a process that is currently accessing the process, the user’s process can continue accessing the process until it closes the process. However, any attempt to reopen the process by that user results in a security violation error (file error 48).

Example

In this example, the owner of the authorization record for the process name $JAM adds an ACL entry to the record.

To display a report for $JAM:

```
=INFO PROCESS $jam
```

The report shows:

```
                         LAST-MODIFIED     OWNER    STATUS    WARNING-MODE
$JAM
                        13MAY86, 11:16     33,13    THAWED        OFF

          033,013      R,W,  P,C
          033,*        R,W,    C
          255,*        R,W,    C
```

To alter the ACL for $JAM:

```
=ALTER PROCESS $jam, ACCESS \*.prs.manager (r,w,c)
```

To display the altered report:

```
=INFO PROCESS $jam
```

The report shows:

```
                        LAST-MODIFIED     OWNER    STATUS   WARNING-MODE
$JAM
                       20AUG86, 13:44     33,13    THAWED        OFF

          033,013      R,W,  P,C
       \*.086,255      R,W,    C
          033,*        R,W,    C
          255,*        R,W,    C
```

This change allows the group manager for group 86 (who is possibly a network user) to read, write, or create processes with the protected process name.

DELETE PROCESS and SUBPROCESS Commands

DELETE PROCESS or SUBPROCESS deletes an authorization record. After an authorization record is deleted, any user can access a process or subprocess under the deleted process name, and any process that is started under that process name is not subject to Safeguard authorization checks or auditing.

An owner of the authorization record, the primary owner's group manager, and the super ID can delete the authorization record for a process or subprocess name.

```
DELETE PROCESS process name-list [ [ , ] WHERE WARNING-MODE ]


DELETE SUBPROCESS subprocess name-list [ [ , ] WHERE WARNING-
MODE ]
```

process name-list

specifies one or more process names for which authorization records are to be deleted. *process name-list* can be either:

process-name

(*process-name* [, *process-name*] ...)

process-name

can be any process name or one of the special names NAMED and UNNAMED. The name can contain wild-card characters.

subprocess name-list

specifies one or more subprocesses for which authorization records are to be deleted. *subprocess name-list* can be either:

subprocess-name

(*subprocess-name* [, *subprocess-name* ...])

subprocess-name

can be any subprocess name. The name can contain wild-card characters.

WHERE *option-list*

specifies that only processes or subprocesses in *filename-list* that have WARNING-MODE set are to be deleted.

Example

The owner of the authorization record for the process name $FIG5 deletes its authorization record:

```
=DELETE PROCESS $fig5
```

FREEZE PROCESS and SUBPROCESS Commands

FREEZE PROCESS or SUBPROCESS temporarily suspends the access authorities granted to users on the ACL in an authorization record. While a process name is frozen, access to the process name authorization record is limited to the primary owner of the process name, an owner on the ACL with the O authority, the primary owner's group manager, and the local super ID.

An owner of the authorization record, the primary owner's group manager, and the super ID can freeze process or subprocess name.

Use THAW PROCESS or SUBPROCESS to restore the access authorities granted to users on the ACL for a frozen process name.

```
FREEZE PROCESS process name-list [ [ , ] WHERE WARNING-MODE ]


FREEZE SUBPROCESS subprocess name-list [ [ , ] WHERE WARNING-
MODE]
```

process name-list

specifies one or more process names to which access is to be frozen. *process name-list* can be either:

process-name

(*process-name* [, *process-name*] ...)

process-name

identifies the process name whose current *process-attribute* values are to be assigned to the process authorization record or records being altered. *process-name* can be any process name or one of the special names NAMED and UNNAMED to refer to any named or unnamed process (respectively).

`subprocess name-list`

specifies one or more subprocesses to which access is to be frozen. *`subprocess name-list`* can be either:

```
  subprocess-name

( subprocess-name [ , subprocess-name ... ] )
```

`subprocess-name`

can be any subprocess name. The name can contain wild-card characters.

```
WHERE WARNING-MODE
```

Considerations

- Freezing the process name for a process that is currently open

  Freezing the process name of a process or subprocess that is currently running and that has been opened by other processes has no effect on these other processes until they close the process or subprocess. Then any attempt to reopen the process or subprocess returns a security violation error (file error 48).

- Accessing to a frozen process or subprocess

  While a process name is frozen, all owners of the process name authorization record and the primary owner’s group manager have all the access authorities (READ, WRITE, CREATE, PURGE, and OWNER) for that process name.

  Unless explicitly denied, the local super ID also retains ownership and has all the authority of any user or group manager on the ACL.

Example

The owner of the authorization record for the process name $LIST suspends access to this name:

```
=FREEZE PROCESS $list
```

INFO PROCESS and SUBPROCESS Commands

INFO PROCESS and SUBPROCESS displays the current attribute values. INFO PROCESS and SUBPROCESS produces two types of reports: brief and detailed. The formats for the two report types are illustrated following the syntax.

Any user can produce an INFO report for any process or subprocess name.

```
INFO [ / OUT listfile / ] PROCESS  process name-list

   [ [ , ] DETAIL ]


INFO [ / OUT listfile / ]  SUBPROCESS subprocess name-list

[ [ , ] DETAIL ]
```

`OUT` *`listfile`*

directs the INFO PROCESS or SUBPROCESS report to *`listfile`*. After executing the INFO command, SAFECOM redirects its output to the current OUT file.

For *`listfile`*, specify any file name. SAFECOM opens *`listfile`* and appends the INFO report to the file. If *`listfile`* does not exist, SAFECOM creates an EDIT file and writes the INFO report to that file.

`process name-list`

specifies one or more process names for which INFO reports are to be produced. *`process name-list`* can be either:

```
  process-name

( process-name [ , process-name ] ... )
```

`process-name`

can be any process name or one of the special names NAMED and UNNAMED. The name can contain wild-card characters.

`subprocess name-list`

specifies one or more subprocesses for which INFO reports are to be produced. *`subprocess name-list`* can be either:

```
  subprocess-name

( subprocess-name [ , subprocess-name ... ] )
```

`subprocess-name`

can be any subprocess name. The name can contain wild-card characters.

`DETAIL`

adds the *`audit-spec`*s defined for the process name to the INFO report. For a full description of the four *`audit-spec`*s, see the SET PROCESS and SUBPROCESS Commands on page 11-28.

INFO PROCESS and SUBPROCESS Brief Report

The brief INFO PROCESS or SUBPROCESS report gives you information about the process name or names you specify. Figure 11-1 shows the format of the brief INFO PROCESS report. The format of the INFO SUBPROCESS report is similar, except that the name of the subprocess replaces the name of the process.

Figure 11-1. INFO PROCESS Brief Report Format

```
                 LAST-MODIFIED OWNER    STATUS    WARNING-MODE
$process
                  date, time   owner-id status      {ON|OFF}

    user-spec [DENY] authority-list
    user-spec [DENY] authority-list
       .
       .
       .
 [ NO ACCESS CONTROL LIST DEFINED! ]
```

Figure 11-1 contains the following process attribute values and status fields:

$process

is the process name whose current attribute values are being displayed.

LAST MODIFIED *date, time*

indicates the date and time of the last change made to this process name authorization record. The times indicated by *date* and *time* are local civil time.

OWNER *owner-id*

is the user ID of the user who owns this process name.

STATUS *status*

is the current status of this process name. *status* is either FROZEN or THAWED.

WARNING-MODE
{ON|OFF}

is the current warning-mode state of this process or subprocess. ON indicates that the protection record is in warning mode. The initial value is OFF, which indicates that warning mode is disabled for this process or subprocess.

user-spec [DENY] *authority-list*

is an entry in the ACL defined for this process name. *user-spec* identifies a single user or user group. *authority-list* is a list of single-character codes that represent the access authorities granted to the user or user group identified by *user-spec*. DENY indicates that the access authorities specified in the *authority-list* are specifically denied to the user or user group identified by *user-spec*.

user-spec has the forms:

```
group-num , member-num
group-num,*
*,*
\node-spec.group-num , member-num
\node-spec.group-num,*
\node-spec.*,*
```

group-num, *member-num* identifies a single local user.

group-num,* identifies all local users in the group that has *group-num*.

. identifies all local users on this process name's node.

\node-spec.*group-num*, *member-num* identifies the local user who has user ID *group-num*, *member-num* and a network user who has the same user name and user ID as that local user.

\node-spec.*group-num*,* identifies all local users in the group that has *group-num* and all network users in that group who also have the same *group-name* as that of the local group who has *group-num*.

\node-spec.*,* identifies all local users on this process's node and all network users who have been granted access to the node.

authority-list can contain any of these codes:

R READ authority

W WRITE authority

C CREATE authority—the authority to create processes with this process name. (This item is not valid for subprocesses.)

P PURGE authority (This item is not valid for subprocesses.)

O OWNER authority

```
[ NO ACCESS CONTROL LIST DEFINED! ]
```

indicates that this process has no ACL. Use ALTER PROCESS...ACCESS to add ACL entries to an existing authorization record. Only the local super ID can access a protected process name that has no ACL.

INFO PROCESS and SUBPROCESS Detailed Report

The detailed INFO PROCESS and SUBPROCESS report includes the auditing specifications currently defined for the protected process name. Figure 11-2 on page 11-26 shows the format of the detailed INFO PROCESS report. The report for an INFO SUBPROCESS is similar, except that the name of the subprocess appears in place of the name of the process.

Figure 11-2. INFO PROCESS Detailed Report Format

```
                LAST-MODIFIED OWNER    STATUS    WARNING-MODE
$process
                  date, time   owner-id status      {ON|OFF}

    user-spec [DENY] auth-list
    user-spec [DENY] auth-list
       .
       .
       .
 [ NO ACCESS CONTROL LIST DEFINED! ]

 OBJECT-TEXT-DESCRIPTION =

 AUDIT-ACCESS-PASS = a-spec  AUDIT-MANAGE-PASS = a-spec
 AUDIT-ACCESS-FAIL = a-spec  AUDIT-MANAGE-FAIL = a-spec
```

In addition to the process attribute values displayed in the brief INFO PROCESS report, the detailed INFO PROCESS report displays these attribute values:

```
AUDIT-ACCESS-PASS = a-spec  AUDIT-MANAGE-PASS = a-spec
AUDIT-ACCESS-FAIL = a-spec  AUDIT-MANAGE-FAIL = a-spec
```

These values indicate the conditions under which the Safeguard software audits attempts to create, open or stop a process that has this process name, and attempts to change or read this authorization record. *a-spec* can be:

```
{ ALL | LOCAL | REMOTE | NONE }
```

For a full description of each *a-spec*, see the appropriate *audit-spec* under the SET PROCESS and SUBPROCESS Commands on page 11-28.

Example

A sample detailed INFO PROCESS report for a process name follows. To display the report:

```
=INFO PROCESS $audit, DETAIL
```

The report shows:

```
                   LAST-MODIFIED      OWNER     STATUS     WARNING-MODE
$AUDIT            22SEP86, 12:12    \*.86,255   THAWED         OFF

         033,013      R,W
         033,017 DENY         C
      \*.086,255      R,W,  P,C
         086,*        R,W,    C
         255,*        R,W,  P,C

 OBJECT-TEXT-DESCRIPTION =

 AUDIT-ACCESS-PASS = ALL       AUDIT-MANAGE-PASS = NONE
 AUDIT-ACCESS-FAIL = ALL       AUDIT-MANAGE-FAIL = NONE
```

This detailed INFO PROCESS report gives these information:

- The authorization record for this process name is owned by the group manager for group 86, and this manager is established as a network user.
- This record was last modified on September 22, 1986 at 12:12 p.m.
- User ID 33,13 can read and write to processes that have this process name, but user 33,17 is specifically denied authority to create processes that have this name.
- The group manager for group 86 has four access authorities for this process name (READ, WRITE, CREATE, and PURGE), and all other members of group 86 have READ, WRITE, and CREATE authority for this process name.
- All members of the super group (group number 255) have four access authorities for process name $AUDIT.
- The Safeguard software audits all attempts to create a process that has process name $AUDIT, and to open or stop a process running under this name.

RESET PROCESS and SUBPROCESS Commands

RESET PROCESS or SUBPROCESS resets the current default attribute values to their predefined values.

When you add an authorization record for a process name, the current default attribute values are used for any attributes you do not specify in the ADD PROCESS or SUBPROCESS command. (To set the default process attribute values to specific values, use the SET PROCESS or SUBPROCESS command.)

```
RESET PROCESS [ [ , ] process-attribute-keyword ]

   [ , process-attribute-keyword ] ...


RESET SUBPROCESS [ [ , ] process-attribute-keyword ]

   [ , process-attribute-keyword ] ...
```

process-attribute-keyword

sets the current default value of the specified attribute to its predefined value. The *process-attribute-keyword*s and their predefined values are:

OWNER	The user ID of the current user
ACCESS	Null (no access control list)
OBJECT-TEXT-DESCRIPTION	Null (no descriptive text or blank)
AUDIT-ACCESS-PASS	NONE (no auditing)
AUDIT-ACCESS-FAIL	NONE (no auditing)
AUDIT-MANAGE-PASS	NONE (no auditing)
AUDIT-MANAGE-FAIL	NONE (no auditing)
WARNING-MODE	OFF (warning mode disabled)

For a complete description of each *process-attribute*, see the SET PROCESS or SUBPROCESS command.

Considerations

- Specifying an attribute name without a value in an ADD or ALTER command causes the attribute to be assigned the predefined default value (as defined for the RESET command).
- If you enter the RESET PROCESS or SUBPROCESS command (or RESET when the assumed object type is PROCESS or SUBPROCESS) and you do not include any *process-attribute-keyword*, all the attributes are returned to their predefined values. The predefined values are listed before the syntax for RESET PROCESS and SUBPROCESS Commands on page 11-27.

Example

In this example, the user ID 86,2 enters a RESET PROCESS command to restore all the default process attribute values to their predefined values.

To display the default values:

```
=SHOW PROCESS
```

This report shows:

```
TYPE        OWNER     WARNING-MODE
 PROCESS     86,2          OFF

 OBJECT-TEXT-DESCRIPTION =

  AUDIT-ACCESS-PASS = ALL        AUDIT-MANAGE-PASS = LOCAL
  AUDIT-ACCESS-FAIL = REMOTE     AUDIT-MANAGE-FAIL = ALL

         086,002      R,W
         086,008      R,W
      \*.086,255      R,W
```

To restore the default values:

```
=RESET PROCESS
=SHOW PROCESS
```

This report shows:

```
TYPE        OWNER      WARNING-MODE
 PROCESS     86,2          OFF

  OBJECT-TEXT-DESCRIPTION =

  AUDIT-ACCESS-PASS = NONE       AUDIT-MANAGE-PASS = NONE
  AUDIT-ACCESS-FAIL = NONE       AUDIT-MANAGE-FAIL = NONE

  NO ACCESS CONTROL LIST DEFINED!
```

SET PROCESS and SUBPROCESS Commands

SET PROCESS or SUBPROCESS establishes default values for one or more attributes. When you add an authorization record, the default attribute values are used for any attribute you do not specify with the ADD PROCESS or SUBPROCESS command.

To display the current default attribute values, use the SHOW PROCESS or SUBPROCESS command.

```
SET PROCESS process name-list [ , ]

   { LIKE process-name | process-attribute }

   [ , process-attribute ] ...


SET SUBPROCESS subprocess name-list [ , ]

   { LIKE subprocess-name | process-attribute }

   [ , process-attribute ] ...
```

`LIKE` *`process-name`*

sets the current default *`process-attribute`* values to be the same as those currently defined for *`process-name`*.

`process-name`

identifies a process name whose current attribute values are to be the current default *`process-attribute`* values. *`process-name`* can be any process name or one of the special names NAMED or UNNAMED.

`LIKE` *`subprocess-name`*

sets the current default *`process-attribute`* values to the same as those currently defined for *`subprocess-name`*.

`subprocess-name`

identifies a subprocess name whose current attribute values are to be the current default *`process-attribute`* values. *`subprocess-name`* can be any process name.

`process-attribute`

defines a default value for the specified process or subprocess. The *`process-attribute`* variables are:

```
OWNER [owner-id]
ACCESS access-spec [ ; access-spec ] ...
OBJECT-TEXT-DESCRIPTION "[any-text]"
AUDIT-ACCESS-PASS [audit-spec]
AUDIT-ACCESS-FAIL [audit-spec]
AUDIT-MANAGE-PASS [audit-spec]
```

```
AUDIT-MANAGE-FAIL [audit-spec]
WARNING-MODE {ON|OFF}
```

`OWNER [owner-id]`

specifies the owner of a process or subprocess name. *owner-id* can be either:

```
[\node-spec.]group-name.member-name
[\node-spec.]group-num , member-num
```

If you omit *owner-id*, *owner-id* is set to the user ID of the current user.

`ACCESS access-spec [ ; access-spec ] ...`

changes the ACL for *filename-list* by adding or deleting ACL entries or by changing the authority list of a current ACL entry.

An ACL contains as many as 50 entries that grant or deny access authorities to users and user groups.

access-spec has the form:

```
user-list  [-] [DENY] authority-list

group-list [-] [DENY] authority-list
```

user-list

specifies those users who are granted (or denied) the access authorities specified with the following *authority-list*. *user-list* can be either:

```
  net-user-spec

( net-user-spec [ , net-user-spec ] ... )
```

net-user-spec can be any of:

```
[\node-spec.]adm-group-name.user-name
[\node-spec.]adm-group-num , user-num
[\node-spec.]adm-group-name.*
[\node-spec.]adm-group-num , *
[\node-spec.]*.*
[\node-spec.]*,*
```

-

(minus-sign) operates on existing ACL entries. The minus-sign form of *access-spec* modifies the current default ACL. The *authority* entries are removed from the default ACL entries for the users specified with *user-list*.

group-list

can be either:

net-group-spec

(*net-group-spec* [, *net-user-spec*] ...)

net-group-spec can be any of:

GROUP [NAME][\node-spec.] *group-name*

GROUP NUMBER [\node-spec.]

node-spec

takes this form:

* | *node-name* | *node-number*

node-name

specifies the system name.

node-number

specifies the Expand node number.

adm-group-name

specifies the name of the administrative group.

adm-group-num

specifies the group number of an administrative group.

group-name

specifies the name of any group.

group-num

specifies the group number of any group.

-

(minus-sign) operates on existing ACL entries. The minus-sign form of *access-spec* modifies the current default ACL. The *authority* entries are removed from the default ACL entries for the users specified with *user-list*.

DENY

specifically denies the *user-list* the access authorities specified with the following:

authority-list

specifies the access authorities to be granted (or denied) to the user or users specified with *user-list*. *authority-list* can be:

```
  authority

( authority [ , authority ]  ... )

  *
```

authority

can be any of:

```
R[EAD]
W[RITE]
C[REATE]
P[URGE]
O[WNER]
```

R and W are not valid for NAMED and UNNAMED processes.

C and P are not valid for subprocesses.

*

specifies all the access authorities (R, W, C, P, and O).

`OBJECT-TEXT-DESCRIPTION "[any-text]"`

allows you to store printable characters as comments. These comments are associated with the objects and are used to manage the object authorization record.

The text description field can accommodate 255 bytes of text data.

Note. The text specified in the text description field overwrites existing data, if any. Also, when LIKE clause is used with SET PROCESS and SUBPROCESS command, the OBJECT-TEXT-DESCRIPTION field is not copied with other object authorization record attributes.

The OBJECT-TEXT-DESCRIPTION attribute is supported only on systems running J06.05 and later J-series RVUs and H06.16 and later H-series RVUs and G06.32 and later G-series RVUs.

`AUDIT-ACCESS-PASS [`*`audit-spec`*`]`

establishes an *audit-spec* for successful attempts to access a process or subprocess name. This *audit-spec* specifies the conditions under which an audit record is written to the audit file when a process name is successfully accessed. (A name is accessed when a process or subprocess is created with that name or when a process or subprocess running with that name is opened or stopped.)

The form of *audit-spec* is:

`{ ALL | LOCAL | REMOTE | NONE }`

`ALL`

All successful access attempts are audited.

`LOCAL`

Only successful access attempts made by local users are audited.

`REMOTE`

Only successful access attempts made by remote users are audited.

`NONE`

No successful access attempts are audited.

Omitting *audit-spec* specifies NONE.

AUDIT-ACCESS-FAIL [*audit-spec*]

establishes an *audit-spec* for unsuccessful attempts to access a process or subprocess name. This *audit-spec* specifies the conditions under which an audit record is written to the audit file when an attempt to access a process name fails. (A name is accessed when a process or subprocess is created with that name or when a process or subprocess running with that name is opened or stopped.)

The form of *audit-spec* is:

`{ ALL | LOCAL | REMOTE | NONE }`

`ALL`

All unsuccessful access attempts are audited.

`LOCAL`

Only unsuccessful access attempts made by local users are audited.

`REMOTE`

Only unsuccessful access attempts made by remote users are audited.

`NONE`

No unsuccessful access attempts are audited.

Omitting *audit-spec* specifies NONE.

`AUDIT-MANAGE-PASS [`*`audit-spec`*`]`

establishes an *`audit-spec`* for successful attempts to change or read an authorization record. This *`audit-spec`* specifies the conditions under which an audit record is written to the audit file when an authorization record is successfully managed.

The form of *`audit-spec`* is:

```
{ ALL | LOCAL | REMOTE | NONE }
```

`ALL`

All successful management attempts are audited.

`LOCAL`

Only successful management attempts made by local users are audited.

`REMOTE`

Only successful management attempts made by remote users are audited.

`NONE`

No successful management attempts are audited.

Omitting *`audit-spec`* specifies NONE.

`AUDIT-MANAGE-FAIL [`*`audit-spec`*`]`

establishes an *`audit-spec`* for unsuccessful attempts to change or read an authorization record. This *`audit-spec`* specifies the conditions under which an audit record is written to the audit file when an attempt to manage an authorization record fails.

The form of *`audit-spec`* is:

```
{ ALL | LOCAL | REMOTE | NONE }
```

`ALL`

All unsuccessful management attempts are audited.

`LOCAL`

Only unsuccessful management attempts made by local users are audited.

`REMOTE`

Only unsuccessful management attempts made by remote users are audited.

NONE

No unsuccessful management attempts are audited.

Omitting *audit-spec* specifies NONE.

WARNING-MODE { ON | OFF }

defines whether warning mode is enabled for the specified process or subprocess. The value is required. For more information on warning mode, see the *Safeguard Administrator's Manual*.

ON enables warning mode for the specified process or subprocess. The initial value is OFF, which disables warning mode for the specified process or subprocess.

Example

User PRS.HARRY (with user ID 86,2) enters the following SET PROCESS commands to specify default ACL entries and audit specifications before creating a process name authorization record with ADD PROCESS:

```
=ASSUME PROCESS
=SET OWNER 86,2
=SET ACCESS 86,2 (r,w,p,c) ; 86,1 DENY c; 86,* (r,w,c)
=SET ACCESS 33,255 (r,w,c)
=SET AUDIT-ACCESS-PASS all , AUDIT-ACCESS-FAIL local
=SHOW
```

This report shows:

```
TYPE           OWNER      WARNING-MODE
 PROCESS       86,2            OFF

  OBJECT-TEXT-DESCRIPTION =

  AUDIT-ACCESS-PASS = ALL       AUDIT-MANAGE-PASS = NONE
  AUDIT-ACCESS-FAIL = LOCAL     AUDIT-MANAGE-FAIL = NONE

         033,255      R,W,    C
         086,001 DENY         C
         086,002      R,W,  P,C
         086,*        R,W,    C
```

These SET PROCESS commands specify:

- The owner of the process name authorization record is user ID 86,2 (PRS.HARRY).
- PRS.HARRY has all four access authorities (READ, WRITE, CREATE, and PURGE).
- User ID 86,1 is denied CREATE authority to the process name.
- All members of group 86 (excluding users 86,2 and 86,1) have READ, WRITE, and CREATE access authorities.

- The group manager for group 33 has READ, WRITE, and CREATE access authorities.
- Two auditing specifications establish that audit records are written each time any successful attempt or any local unsuccessful attempt is made to create a process that has this process name or to open or stop a process running under this process name.

To specify the process name to have these default attribute values (and to create the authorization record), use ADD PROCESS.

SHOW PROCESS and SUBPROCESS Commands

SHOW PROCESS or SUBPROCESS displays the current default attribute values.

When you add an authorization record, the current default attribute values are used for any attributes you do not specify in the ADD PROCESS or SUBPROCESS command. To set the default attribute values to specific values, use the SET PROCESS or SUBPROCESS command.

```
SHOW [ / OUT listfile / ] { PROCESS | SUBPROCESS }
```

OUT *listfile*

directs the SHOW PROCESS or SUBPROCESS report to *listfile*. After executing the SHOW command, SAFECOM redirects its output to the current OUT file.

For *listfile*, specify any file name. SAFECOM opens *listfile* and appends the SHOW PROCESS report to the file. If *listfile* does not exist, SAFECOM creates an EDIT file and writes the SHOW PROCESS report to that file.

SHOW PROCESS and SUBPROCESS Report Format

The SHOW PROCESS command displays the current default attribute values in the format shown in Figure 11-3. The SHOW SUBPROCESS command output is similar, except that a subprocess name appears in place of a process name.

Figure 11-3. SHOW PROCESS Report Format

```
TYPE              OWNER      WARNING-MODE
 PROCESS          gn,un        {ON|OFF}

  OBJECT-TEXT-DESCRIPTION =


  AUDIT-ACCESS-PASS = a-spec AUDIT-MANAGE-PASS = a-spec
  AUDIT-ACCESS-FAIL = a-spec AUDIT-MANAGE-FAIL = a-spec

     user-spec [DENY] authority-list
     user-spec [DENY] authority-list
          .        .         .
          .        .         .
  [ NO ACCESS CONTROL LIST DEFINED! ]
```

The SHOW PROCESS report displays these attribute values:

`OWNER` *`gn,un`*

is the user ID (group number and member number) of the user who will own this process name authorization record if a process name having these attribute values is added to Safeguard protection.

`WARNING-MODE`
`{ON|OFF}`

is the current warning-mode state of this process or subprocess. ON indicates that the protection record is in warning mode. The initial value is OFF, which indicates that warning mode is disabled for this process or subprocess.

`AUDIT-ACCESS-PASS =` *`a-spec`* `AUDIT-MANAGE-PASS =` *`a-spec`*
`AUDIT-ACCESS-FAIL =` *`a-spec`* `AUDIT-MANAGE-FAIL =` *`a-spec`*

are the conditions under which the Safeguard software will audit attempts to access this process name and attempts to manage this authorization record. These fields are described in detail for *`audit-spec`* under the SET PROCESS or SUBPROCESS command.

`user-spec` `[DENY]` *`auth-list`*

is a current default ACL entry for processes. For a full description, see INFO PROCESS and SUBPROCESS Brief Report on page 11-24.

`[ NO ACCESS CONTROL LIST DEFINED! ]`

indicates no ACL entries have been defined in the current default attribute values. Use SET PROCESS...ACCESS or SET SUBPROCESS...ACCESS to define default ACL entries. You can use ADD PROCESS...ACCESS or ADD

SUBPROCESS...ACCESS to define ACL entries when you create an authorization record.

△ **Caution.** If you do not specify an ACL for a process, only the local super ID can access the process or subprocess name.

Example

The SHOW PROCESS report displays the current default process attribute values during a SAFECOM session. For example:

```
=SHOW PROCESS
```

```
TYPE        OWNER     WARNING-MODE
 PROCESS    86,2          OFF

  OBJECT-TEXT-DESCRIPTION =

  AUDIT-ACCESS-PASS = ALL        AUDIT-MANAGE-PASS = ALL
  AUDIT-ACCESS-FAIL = ALL        AUDIT-MANAGE-FAIL = ALL

  NO ACCESS CONTROL LIST DEFINED!
```

This display shows that no ACL is defined, but each of the four auditing specifications is set to ALL. The owner of the potential process name authorization record is PRS.HARRY, who has user ID 86,2.

THAW PROCESS and SUBPROCESS Commands

THAW PROCESS or SUBPROCESS restores the ACL for a frozen process name. After a frozen process name is thawed, users who were granted access authority by the ACL for the process name can once again access the process name.

An owner of the authorization record, the primary owner’s group manager, and the super ID can thaw a frozen process or subprocess name.

THAW PROCESS or SUBPROCESS has no effect on a process name that is not frozen.

```
THAW PROCESS process name-list [ [ , ] WHERE option-list ]


THAW SUBPROCESS subprocess name-list [ [ , ] WHERE WARNING-
MODE ]
```

process name-list

specifies one or more process names that are to be thawed. *process name-list* can be either:

process-name

(*process-name* [, *process-name*] ...)

process-name

can be any process name or one of the special names NAMED and UNNAMED. The name can contain wild-card characters.

subprocess name-list

specifies one or more subprocess names to be thawed. *subprocess name-list* can be either:

subprocess-name

(*subprocess-name* [, *subprocess-name* ...])

subprocess-name

can be any subprocess name. The name can contain wild-card characters.

WHERE WARNING-MODE

specifies that only processes or subprocesses in *filename-list* that have WARNING-MODE set are to be thawed.

Example

The owner of the authorization record for the process name $CHCK thaws the ACL for the process name:

```
=THAW PROCESS $chck
```

12

OBJECTTYPE Security Commands

Safeguard OBJECTTYPE security allows a security administrator to define the user or groups of users who can add new subjects or objects to the Safeguard database.

Each kind of subject and object (such as DISKFILE, DEVICE, or USER) can be given a corresponding OBJECTTYPE protection record. For example, the protection record to control adding new DISKFILEs is an entry for OBJECTTYPE DISKFILE. However, authorities granted on the access control list (ACL) for OBJECTTYPE DISKFILE do not represent permissions for individual disk files but rather the ability to add new disk files to the Safeguard database.

When a user attempts an ADD command (for example, ADD DISKFILE), the Safeguard software first checks for the presence of an authorization record for the corresponding OBJECTTYPE (in this case, OBJECTTYPE DISKFILE). If no record exists, the Safeguard software proceeds according to default rules, which are shown in Table 12-1 on page 12-2. However, if a record exists for the corresponding OBJECTTYPE, the Safeguard software consults the ACL for that OBJECTTYPE. If the user has not been granted C (CREATE) authority on the ACL, the ADD command fails with a security violation (file error 48).

Protection records for OBJECTTYPEs are similar to protection records for individual objects: the initial owner can grant additional ownership (through the O authority on the ACL), the owner can give ownership away, the owner can freeze or thaw the protection record, and the owner can establish selective auditing criteria. Owners can even delete the protection record for an OBJECTTYPE to restore the operation of the ADD command for that OBJECTTYPE back to the default rules.

Because the OBJECTTYPE records alter the behavior of the Safeguard ADD command, consider carefully the consequences of changing the Safeguard software from the default behavior by adding an OBJECTTYPE record. Table 12-1 lists the default behaviors.

Because the OBJECTTYPE records are in themselves *pseudo-objects*, an additional OBJECTTYPE record exists to control the creation of new OBJECTTYPE records. This additional record is the OBJECTTYPE OBJECTTYPE record. Only users granted CREATE authority on the OBJECTTYPE OBJECTTYPE ACL (if present) can create other OBJECTTYPE records. Only the owner and other users granted OWNER authority on the OBJECTTYPE OBJECTTYPE ACL can manage the OBJECTTYPE OBJECTTYPE record.

OBJECTTYPE DISKFILE has no effect on default protection for a user’s disk files. It only controls who can execute the ADD DISKFILE command.

Table 12-1. Defaults for Undefined OBJECTTYPE ACLs

Type of Object	Who Can Place an Object Under Safeguard Control
ALIAS	Group manager of underlying user ID. Also must be the owner of underlying user ID or owner’s group manager.
DEVICE	Any local super group member
DISKFILE	Local owner of the existing file
DISKFILE-PATTERN	Any local user
GROUP	Any local super group member
OBJECTTYPE	Any local super group member
PROCESS	Any local user
SUBDEVICE	Any local super group member
SUBPROCESS	Any local user
SUBVOLUME	Any local user
USER	Guardian rules apply: ● The local super ID can create any local user ID. ● The local group manager can create any local group member user ID.
VOLUME	Any local super-group member

Note. OBJECTTYPE USER controls who can add users, aliases, and groups.

OBJECTTYPE Access Authorities

The ACL defined for an OBJECTTYPE can grant any of these access authorities to users and user groups:

CREATE	Add an authorization record for an object of this type
OWNER	Manage the OBJECTTYPE authorization record

Note. Starting with H06.24/J06.13 RVUs, the OBJECTTYPE USER is granted additional access permissions, WRITE (W) and PURGE (P), along with the existing CREATE (C) and OWNER (O) permissions. Members having the WRITE (W) permission on OBJECTTYPE USER can modify any subject records. Members having the PURGE (P) permission on OBJECTTYPE USER can purge any subject records.

Note. Starting with H06.26/J06.15 RVUs, the OBJECTTYPE DISKFILE/VOLUME/SUBVOLUME is granted additional access permissions, WRITE (W) and PURGE (P), along with the existing CREATE (C) and OWNER (O) permissions. Members having the WRITE (W) permission on OBJECTTYPE DISKFILE/VOLUME/SUBVOLUME can modify the respective DISKFILE/VOLUME/SUBVOLUME protection records. Members having the PURGE (P) permission on OBJECTTYPE DISKFILE/VOLUME/SUBVOLUME can purge the respective DISKFILE/VOLUME/SUBVOLUME protection records.

OBJECTTYPE Security Command Summary

Table 12-2 on page 12-3 lists the OBJECTTYPE security commands and gives a brief description of each.

Table 12-2. OBJECTTYPE Security Command Summary (page 1 of 2)

Command	**Description**
ADD OBJECTTYPE	Adds an OBJECTTYPE authorization record with the specified OBJECTTYPE attribute values. If you do not specify attribute values, the current default is used. By default, only a member of the local super group can add an authorization record for an object type.
ALTER OBJECTTYPE	Changes one or more attribute values in an OBJECTTYPE authorization record. For all attributes except ACCESS, ALTER OBJECTTYPE replaces the current value with the specified value. For the ACCESS attribute, ALTER OBJECTTYPE changes the existing ACL to incorporate *access-spec*.
DELETE OBJECTTYPE	Deletes an OBJECTTYPE authorization record. Afterward, requests to create an authorization record for any object of the specified object type are subject to the rules described in Table 12-1.
FREEZE OBJECTTYPE	Temporarily disables authorities granted to users who have OBJECTTYPE access. When an OBJECTTYPE is frozen, only the primary owner, the primary owner’s group manager, owners on the ACL, and the local super ID can create authorization records for that type of object.
INFO OBJECTTYPE	Displays the existing attribute values in an OBJECTTYPE authorization record.
RESET OBJECTTYPE	Sets default OBJECTTYPE attribute values to the predefined values of the SET command.

Table 12-2. OBJECTTYPE Security Command Summary (page 2 of 2)

Command	Description
SET OBJECTTYPE	Sets OBJECTTYPE attribute values to specified default values.
SHOW OBJECTTYPE	Displays the current default values of the OBJECTTYPE attributes.
THAW OBJECTTYPE	Reenables a frozen OBJECTTYPE. Then user IDs with appropriate entries on the OBJECTTYPE ACL can create authorization records once again.

Syntax of OBJECTTYPE Security Commands

The remainder of this section describes each OBJECTTYPE security command in detail. Commands are presented in alphabetical order, and descriptions contain these elements:

- A summary of the command’s function, including the restrictions on who can use the command
- The command syntax, including descriptions of the command parameters and variables
- The format for any command listing or report
- Considerations for the use of the command
- Examples of command usage

ADD OBJECTTYPE Command

ADD OBJECTTYPE creates a Safeguard authorization record for a class of objects. After an OBJECTTYPE authorization record is created, every attempt to create an authorization record for an object of that type is subject to Safeguard authorization checks and, optionally, to Safeguard auditing. By default, only a member of the local super-group can add an OBJECTTYPE authorization record for a class of objects.

You can use SET OBJECTTYPE to establish default attributes and then use ADD OBJECTTYPE to name the object class to which the default attributes are to be applied. You can also specify values for the OBJECTTYPE attributes in the ADD OBJECTTYPE command. The current default values are used for any attributes not specified.

```
ADD OBJECTTYPE objecttype-list [ , ]

   [ LIKE objecttype-spec | objecttype-attribute ]

   [ , objecttype-attribute ] ...
```

`objecttype-list`

specifies one or more object types for which authorization records are to be added. *`objecttype-list`* can be either:

```
  objecttype-spec

( objecttype-spec [ , objecttype-spec ] ... )
```

`objecttype-spec`

can be any object class or type, including OBJECTTYPE:

```
DEVICE
DISKFILE
DISKFILE-PATTERN
SAVED-DISCFILE-PATTERN
OBJECTTYPE
PROCESS
SUBDEVICE
SUBPROCESS
SUBVOLUME
USER
VOLUME
```

Note. OBJECTTYPE USER also controls who can use the ADD ALIAS and ADD GROUP commands.

`LIKE` *`objecttype-spec`*

adopts the existing OBJECTTYPE attribute values of *`objecttype-spec`* as the attribute values to be used for the authorization record or records being added.

`objecttype-spec`

identifies the OBJECTTYPE whose current *`objecttype-attribute`* values are to be assigned to the OBJECTTYPE authorization record or records being added. *`objecttype-spec`* can be any object class or type name.

`objecttype-attribute`

defines an OBJECTTYPE attribute value for the OBJECTTYPE authorization record or records being added. The permitted *`objecttype-attribute`*s are:

```
OWNER [owner-id]
ACCESS access-spec [ ; access-spec ] ...
OBJECT-TEXT-DESCRIPTION “[any-text]”
AUDIT-ACCESS-PASS [audit-spec]
AUDIT-ACCESS-FAIL [audit-spec]
AUDIT-MANAGE-PASS [audit-spec]
AUDIT-MANAGE-FAIL [audit-spec]
```

OWNER [*owner-id*]

specifies the new owner of the class of objects. The *owner-id* can be either:

```
[\node-spec.]group-name.member-name
[\node-spec.]group-num , member-num
```

If you omit *owner-id*, *owner-id* is set to your user ID.

ACCESS *access-spec* [; *access-spec*] ...

changes the ACL for *filename-list* by adding or deleting ACL entries or by changing the authority list of a current ACL entry.

An ACL contains as many as 50 entries that grant or deny access authorities to users and user groups.

access-spec has the form:

```
user-list  [-] [DENY] authority-list

group-list [-] [DENY] authority-list
```

user-list

specifies those users who are granted (or denied) the access authorities specified with the following *authority-list*. *user-list* can be either:

```
  net-user-spec

( net-user-spec [ , net-user-spec ] ... )
```

net-user-spec can be any of:

```
[\node-spec.]adm-group-name.user-name
[\node-spec.]adm-group-num , user-num
[\node-spec.]adm-group-name.*
[\node-spec.]adm-group-num , *
[\node-spec.]*.*
[\node-spec.]*,*
```

-

(minus-sign) operates on existing ACL entries. The minus-sign form of *access-spec* modifies the current default ACL. The *authority* entries are removed from the default ACL entries for the users specified with *user-list*.

group-list

can be either:

net-group-spec

(*net-group-spec* [, *net-user-spec*] ...)

net-group-spec can be any of:

GROUP [NAME][\node-spec.] *group-nam*e

GROUP NUMBER [\node-spec.]

node-spec

takes this form:

* | *node-name* | *node-number*

node-name

specifies the system name.

node-number

specifies the Expand node number.

adm-group-name

specifies the name of the administrative group.

adm-group-num

specifies the group number of an administrative group.

group-name

specifies the name of any group.

group-num

specifies the group number of any group.

-

(minus-sign) operates on existing ACL entries. The minus-sign form of *access-spec* modifies the current default ACL. The *authority* entries are removed from the default ACL entries for the users specified with *user-list*.

DENY

denies the user IDs or user groups specified by *user-list* the access authorities specified by *authority-list*.

authority-list

specifies the access authorities to be granted (or denied) to *user-list*. *authority-list* can be any one of:

```
  authority

( authority [ , authority ] ... )

  *
```

authority

is one of the following:

```
C[REATE]
O[WNER]
```

`*`

(asterisk) specifies CREATE and OWNER.

Note. Starting with H06.24/J06.13 RVUs, the OBJECTTYPE USER is granted additional access permissions, WRITE (W) and PURGE (P), along with the existing CREATE (C) and OWNER (O) permissions. Members having the WRITE (W) permission on OBJECTTYPE USER can modify any subject records. Members having the PURGE (P) permission on OBJECTTYPE USER can purge any subject records.

Note. Starting with H06.26/J06.15 RVUs, the OBJECTTYPE DISKFILE/VOLUME/SUBVOLUME is granted additional access permissions, WRITE (W) and PURGE (P), along with the existing CREATE (C) and OWNER (O) permissions. Members having the WRITE (W) permission on OBJECTTYPE DISKFILE/VOLUME/SUBVOLUME can modify the respective DISKFILE/VOLUME/SUBVOLUME protection records. Members having the PURGE (P) permission on OBJECTTYPE DISKFILE/VOLUME/SUBVOLUME can purge the respective DISKFILE/VOLUME/SUBVOLUME protection records.

Note. Specifying ACCESS *access-spec* with ADD OBJECTTYPE does not override the current default ACL (established with SET OBJECTTYPE). Instead, any ACL entries specified with ADD OBJECTTYPE modify the template of current default settings.

`OBJECT-TEXT-DESCRIPTION “[any-text]”`

allows you to store printable characters as comments. These comments are associated with the objects and are used to manage the object authorization record.

The text description field can accommodate 255 bytes of text data.

Note. The text specified in the text description field overwrites existing data, if any. Also, when LIKE clause is used with ADD OBJECTTYPE command, the OBJECT-TEXT-DESCRIPTION field is not copied with other object authorization record attributes.

The OBJECT-TEXT-DESCRIPTION attribute is supported only on systems running J06.05 and later J-series RVUs and H06.16 and later H-series RVUs and G06.32 and later G-series RVUs.

`AUDIT-ACCESS-PASS [`*`audit-spec`*`]`

changes the *`audit-spec`* for successful attempts to add an authorization record for an object of this type. The form of *`audit-spec`* is:

```
{ ALL | LOCAL | REMOTE | NONE }
```

For a description of the *`audit-spec`*s, see the SET OBJECTTYPE Command on page 12-23. Omitting *`audit-spec`* specifies NONE.

`AUDIT-ACCESS-FAIL [`*`audit-spec`*`]`

changes the *`audit-spec`* for unsuccessful attempts to add an authorization record for an object this type. The form of *`audit-spec`* is:

```
{ ALL | LOCAL | REMOTE | NONE }
```

For a description of the *`audit-spec`*s, see the SET OBJECTTYPE Command on page 12-23. Omitting *`audit-spec`* specifies NONE.

`AUDIT-MANAGE-PASS [`*`audit-spec`*`]`

changes the *`audit-spec`* for successful attempts to manage this authorization record. The form of *`audit-spec`* is:

```
{ ALL | LOCAL | REMOTE | NONE }
```

For a description of the *`audit-spec`*s, see the SET OBJECTTYPE Command on page 12-23. Omitting *`audit-spec`* specifies NONE.

`AUDIT-MANAGE-FAIL [`*`audit-spec`*`]`

changes the *`audit-spec`* for unsuccessful attempts to manage this authorization record. The form of *`audit-spec`* is:

```
{ ALL | LOCAL | REMOTE | NONE }
```

For a description of the *`audit-spec`*s, see the SET OBJECTTYPE Command on page 12-23. Omitting *`audit-spec`* specifies NONE.

For a complete description of the *`objecttype-attribute`*s, see the SET OBJECTTYPE Command on page 12-23.

Considerations

- Additional owners can modify the authorization record.

  In addition to the primary owner, the primary owner’s group manager, and the local super ID, any user ID that has an ACL entry granting OWNER authority can also modify the OBJECTTYPE authorization record.

- Attributes in an ADD command affect only the record added.

  Any attribute specifications in an ADD OBJECTTYPE command affect only the authorization record being created and do not change the current default OBJECTTYPE attribute values. This condition is also true for a LIKE clause in an ADD OBJECTTYPE command.

Example

You can use a LIKE *objecttype-name* clause to copy all attribute values for one class of objects from another class of objects. Then you can specify in the same command line that one or more attribute values will be different.

This sample command adds an authorization record for DEVICE that has the same OBJECTTYPE attribute values (and ACL) as PROCESS, except for the OWNER attribute:

```
=ADD OBJECTTYPE device, LIKE process, OWNER super.bob
```

ALTER OBJECTTYPE Command

ALTER OBJECTTYPE changes one or more attribute values in an OBJECTTYPE authorization record. The owner, the primary owner’s group manager, and the super ID can change an OBJECTTYPE authorization record. In addition, any user ID that has an ACL entry granting it O[WNER] authority can modify the OBJECTTYPE authorization record.

Except for the ACCESS attribute, new object type attribute values specified in an ALTER OBJECTTYPE command replace the existing attribute value. Specifying a new ACCESS *access-spec* adds the new *access-spec* to the object type’s existing ACL. To remove authorities previously granted to user IDs, use the minus-sign (-) form of *access-spec*.

```
ALTER OBJECTTYPE objecttype-list [ , ]

   { LIKE objecttype-spec | objecttype-attribute }

   [ , objecttype-attribute ] ...
```

objecttype-list

specifies one or more object types whose existing *objecttype-attribute* values are to be changed. All object types specified must already have Safeguard authorization records (created with the ADD OBJECTTYPE command).

objecttype-list can be either:

```
  objecttype-spec

( objecttype-spec [ , objecttype-spec ] ... )
```

objecttype-spec

can be the name of any class of objects, including OBJECTTYPE:

```
DEVICE
DISKFILE
SAVED-DISCFILE-PATTERN
OBJECTTYPE
PROCESS
SUBDEVICE
SUBPROCESS
SUBVOLUME
USER
VOLUME
```

LIKE *objecttype-spec*

changes the attribute values of *objecttype-list* to the same as the existing attribute values for *objecttype-spec*. For the ACCESS attribute, LIKE only adds ACL entries or adds authorities to existing entries. It does not replace or delete ACL entries or authorities.

objecttype-spec

identifies the class of objects whose existing *objecttype-attribute* values are to be assigned to the OBJECTTYPE authorization record being changed. *objecttype-spec* can be any object class name.

objecttype-attribute

changes the existing value of the specified object-class attribute for the object type being changed. The *objecttype-attribute*s values are:

```
OWNER [owner-id]
ACCESS access-spec [ ; access-spec ] ...
OBJECT-TEXT-DESCRIPTION “[any-text]”
RESET-OBJECT-TEXT-DESCRIPTION
AUDIT-ACCESS-PASS [audit-spec]
AUDIT-ACCESS-FAIL [audit-spec]
AUDIT-MANAGE-PASS [audit-spec]
AUDIT-MANAGE-FAIL [audit-spec]
```

`OWNER [`*`owner-id`*`]`

specifies the new owner of the class of objects. The *`owner-id`* can be either:

```
[\node-spec.]group-name.member-name
[\node-spec.]group-num , member-num
```

If you omit *`owner-id`*, *`owner-id`* is set to your user ID.

`ACCESS` *`access-spec`* `[ ;` *`access-spec`* `] ...`

changes the ACL for *`filename-list`* by adding or deleting ACL entries or by changing the authority list of a current ACL entry.

An ACL contains as many as 50 entries that grant or deny access authorities to users and user groups.

`access-spec` has the form:

```
user-list  [-] [DENY] authority-list

group-list [-] [DENY] authority-list
```

`user-list`

specifies those users who are granted (or denied) the access authorities specified with the following *`authority-list`*. *`user-list`* can be either:

```
  net-user-spec

( net-user-spec [ , net-user-spec ] ... )
```

`net-user-spec` can be any of:

```
[\node-spec.]adm-group-name.user-name
[\node-spec.]adm-group-num , user-num
[\node-spec.]adm-group-name.*
[\node-spec.]adm-group-num , *
[\node-spec.]*.*
[\node-spec.]*,*
```

-

(minus-sign) operates on existing ACL entries. The minus-sign form of *`access-spec`* modifies the current default ACL. The *`authority`* entries are removed from the default ACL entries for the users specified with *`user-list`*.

group-list

can be either:

net-group-spec

(*net-group-spec* [, *net-user-spec*] ...)

net-group-spec can be any of:

GROUP [NAME][\node-spec.] *group-nam*e

GROUP NUMBER [\node-spec.]

node-spec

takes this form:

* | *node-name* | *node-number*

node-name

specifies the system name.

node-number

specifies the Expand node number.

adm-group-name

specifies the name of the administrative group.

adm-group-num

specifies the group number of an administrative group.

group-name

specifies the name of any group.

group-num

specifies the group number of any group.

-

(minus-sign) operates on existing ACL entries. The minus-sign form of *access-spec* modifies the current default ACL. The *authority* entries are removed from the default ACL entries for the users specified with *user-list*.

DENY

denies the user IDs or user groups specified by *user-list* the access authorities specified by *authority-list*.

authority-list

specifies the access authorities to be granted (or denied) to *user-list*. *authority-list* can be any one of:

```
  authority

( authority [ , authority ] ... )

  *
```

authority

is one of the following:

```
C[REATE]
O[WNER]
```

*

(asterisk) specifies CREATE and OWNER.

Note. Starting with H06.24/J06.13 RVUs, the OBJECTTYPE USER is granted additional access permissions, WRITE (W) and PURGE (P), along with the existing CREATE (C) and OWNER (O) permissions. Members having the WRITE (W) permission on OBJECTTYPE USER can modify any subject records. Members having the PURGE (P) permission on OBJECTTYPE USER can purge any subject records.

Note. Starting with H06.26/J06.15 RVUs, the OBJECTTYPE DISKFILE/VOLUME/SUBVOLUME is granted additional access permissions, WRITE (W) and PURGE (P), along with the existing CREATE (C) and OWNER (O) permissions. Members having the WRITE (W) permission on OBJECTTYPE DISKFILE/VOLUME/SUBVOLUME can modify the respective DISKFILE/VOLUME/SUBVOLUME protection records. Members having the PURGE (P) permission on OBJECTTYPE DISKFILE/VOLUME/SUBVOLUME can purge the respective DISKFILE/VOLUME/SUBVOLUME protection records.

Note. Specifying ACCESS *access-spec* with ADD OBJECTTYPE does not override the current default ACL (established with SET OBJECTTYPE). Instead, any ACL entries specified with ADD OBJECTTYPE modify the template of current default settings.

`OBJECT-TEXT-DESCRIPTION “[any-text]”`

allows you to store printable characters as comments. These comments are associated with the objects and are used to manage the object authorization record.

The text description field can accommodate 255 bytes of text data.

Note. The text specified in the text description field overwrites existing data, if any. Also, when LIKE clause is used with ALTER OBJECTTYPE command, the OBJECT-TEXT-DESCRIPTION field is not copied with other object authorization record attributes. Also, if you specify OBJECT-TEXT-DESCRIPTION without any text in the quotation marks, the object text description for this record is removed.

The OBJECT-TEXT-DESCRIPTION attribute is supported only on systems running J06.05 and later J-series RVUs and H06.16 and later H-series RVUs and G06.32 and later G-series RVUs.

`RESET-OBJECT-TEXT-DESCRIPTION`

Resets the object description to Null.

Note. The RESET-OBJECT-TEXT-DESCRIPTION attribute is supported only on systems running J06.05 and later J-series RVUs and H06.16 and later H-series RVUs and G06.32 and later G-series RVUs.

`AUDIT-ACCESS-PASS [`*`audit-spec`*`]`

changes the *`audit-spec`* for successful attempts to add an authorization record for a specific object. The form of *`audit-spec`* is:

```
{ ALL | LOCAL | REMOTE | NONE }
```

For a description of the *`audit-spec`*s, see the SET OBJECTTYPE Command on page 12-23. Omitting *`audit-spec`* specifies NONE.

`AUDIT-ACCESS-FAIL [`*`audit-spec`*`]`

changes the *`audit-spec`* for unsuccessful attempts to add an authorization record for a specific object. The form of *`audit-spec`* is:

```
{ ALL | LOCAL | REMOTE | NONE }
```

For a description of the *`audit-spec`*s, see the SET OBJECTTYPE Command on page 12-23. Omitting *`audit-spec`* specifies NONE.

`AUDIT-MANAGE-PASS [`*`audit-spec`*`]`

changes the *`audit-spec`* for successful attempts to manage this authorization record. The form of *`audit-spec`* is:

```
{ ALL | LOCAL | REMOTE | NONE }
```

For a description of the *`audit-spec`* variables, see the SET OBJECTTYPE Command on page 12-23. Omitting *`audit-spec`* specifies NONE.

AUDIT-MANAGE-FAIL [*audit-spec*]

changes the *audit-spec* for unsuccessful attempts to manage this authorization record. The form of *audit-spec* is:

```
{ ALL | LOCAL | REMOTE | NONE }
```

For a description of the *audit-spec*s, see the SET OBJECTTYPE Command on page 12-23. Omitting *audit-spec* specifies NONE.

Example

This command transfers ownership of the DISKFILE object type to the user with user ID 86,13 and allows all users who are members of group number 86 to add new disk file authorization records:

```
=ALTER OBJECTTYPE DISKFILE , OWNER 86,13, ACCESS 86,* c
```

DELETE OBJECTTYPE Command

DELETE OBJECTTYPE deletes an OBJECTTYPE authorization record. After an OBJECTTYPE authorization record is deleted, that class of objects is subject to Safeguard authorization checks as described in Table 12-1 on page 12-2.

The primary owner, the primary owner’s group manager, and the super ID can delete an OBJECTTYPE authorization record. In addition, any user ID that has an ACL entry granting it OWNER authority can also delete the OBJECTTYPE authorization record.

```
DELETE OBJECTTYPE objecttype-list
```

objecttype-list

specifies one or more OBJECTTYPEs for which authorization records are to be deleted. *objecttype-list* can be either:

```
  objecttype-spec

( objecttype-spec [ , objecttype-spec ] ... )
```

objecttype-spec

can be any object class name, including OBJECTTYPE:

```
DEVICE
DISKFILE
DISKFILE-PATTERN
SAVED-DISCFILE-PATTERN
OBJECTTYPE
PROCESS
SUBDEVICE
SUBPROCESS
SUBVOLUME
```

```
USER
VOLUME
```

Example

As owner of the object class DEVICE, you can enter the command to delete the Safeguard authorization record for OBJECTTYPE DEVICE:

```
=DELETE OBJECTTYPE device
```

FREEZE OBJECTTYPE Command

FREEZE OBJECTTYPE temporarily suspends the authorities granted to user IDs listed on an object-class ACL. While the object class is frozen, only the primary owner, the owner's group manager, an owner on the ACL, and the local super ID can create authorization records for that type of object.

Use the THAW OBJECTTYPE command to reenable all the ACL authorities granted to user IDs before the object class was frozen.

```
FREEZE OBJECTTYPE objecttype-list
```

objecttype-list

specifies the object class for which access is to be frozen. *objecttype-list* can be either:

```
  objecttype-spec

( objecttype-spec [ , objecttype-spec ] ... )
```

objecttype-spec

can be any object-class name, including OBJECTTYPE:

```
DEVICE
DISKFILE
DISKFILE-PATTERN
SAVED-DISCFILE-PATTERN
OBJECTTYPE
PROCESS
SUBDEVICE
SUBPROCESS
SUBVOLUME
USER
VOLUME
```

Consideration

While a class of objects is frozen, the primary owner, the owner's group manager, and an owner on the ACL are implicitly granted all the access authorities.

The local super ID also retains ownership and has all the authority of any user or group manager unless explicitly denied.

Example

To disable access to the object-class DEVICE, the owner enters:

```
=FREEZE OBJECTTYPE device
```

INFO OBJECTTYPE Command

INFO OBJECTTYPE displays the attribute values currently stored in an OBJECTTYPE authorization record and produces two types of reports: brief and detailed. The format for each report is illustrated after the following syntax description.

Any user can produce an INFO report on any type of object.

```
INFO [ / OUT listfile / ] OBJECTTYPE objecttype-list

   [ [ , ] DETAIL ]
```

`OUT`

directs the INFO OBJECTTYPE report to *listfile*. After it executes the INFO command, SAFECOM redirects its output to the current OUT file.

listfile

For *listfile*, specify any file name. SAFECOM opens the *listfile* and appends the INFO report to the file. If *listfile* does not exist, SAFECOM creates it as an EDIT-format file.

objecttype-list

specifies the object class for which INFO reports are to be produced. *objecttype-list* can be either:

```
  objecttype-spec

( objecttype-spec [ , objecttype-spec ] ... )
```

objecttype-spec

can be any object-class name, including OBJECTTYPE:

```
DEVICE
DISKFILE
DISKFILE-PATTERN
SAVED-DISCFILE-PATTERN
OBJECTTYPE
PROCESS
SUBDEVICE
SUBPROCESS
```

SUBVOLUME
USER
VOLUME

DETAIL

adds the *audit-specs* defined for the object type to the INFO report. For a full description of the four *audit-spec*s, see the SET OBJECTTYPE Command on page 12-23.

INFO OBJECTTYPE Brief Report

The brief INFO OBJECTTYPE report displays the attribute values currently stored for this object class. Figure 12-1 shows the format of the brief INFO OBJECTTYPE report.

Figure 12-1. INFO OBJECTTYPE Brief Report Format

```
                LAST-MODIFIED OWNER    STATUS
objecttype
                  date,time   owner-id status

   user-spec [DENY] authority-list
   user-spec [DENY] authority-list
      .
      .
      .
[ NO ACCESS CONTROL LIST DEFINED! ]
```

Figure 12-1 contains these OBJECTTYPE attribute values and status fields:

objecttype

is the name of the object class whose existing attribute values are being displayed.

LAST MODIFIED TIME
date, time

is the date and time of the last change made to this OBJECTTYPE authorization record. *date* and *time* are in local civil time.

OWNER
owner-id

is the user ID of the person who owns this OBJECTTYPE authorization record.

STATUS
status

indicates the current status of this object class. *status* is either FROZEN or THAWED.

user-spec [DENY] *authority-list*

is an entry in the ACL defined for this object class. *user-spec* identifies a single user or user group. *authority-list* is a list of single-character codes that represent the access authorities granted to the user or user group identified by *user-spec*. DENY indicates that the access authorities specified with *authority-list* are specifically denied to the user or user group identified by *user-spec*.

user-spec can be any of:

```
group-num, member-num
group-num, *
*,*
\node-spec.group-num, member-num
\node-spec.group-num, *
\node-spec.*,*
```

group-num, *member-num* identifies a single local user.

group-num,* identifies all the local users in the group that has *group-num*.

, identifies all the local users.

\node-spec.*group-num*, *member-num* identifies both the local user with user ID *group-num*, *member-num* and a network user with the same user name and user ID as that local user.

\node-spec.*group-num*,* identifies all the local users in the group identified by *group-num* and all network users whose *group-num* and *group-name* match those of the local group.

\node-spec.*,* identifies all local users as well as all network users with access to the local node.

authority-list for this object type can contain one of these codes:

```
C - CREATE authority
O - OWNER  authority
```

Note. Starting with H06.24/J06.13 RVUs, the OBJECTTYPE USER is granted additional access permissions, WRITE (W) and PURGE (P), along with the existing CREATE (C) and OWNER (O) permissions. Members having the WRITE (W) permission on OBJECTTYPE USER can modify any subject records. Members having the PURGE (P) permission on OBJECTTYPE USER can purge any subject records.

Note. Starting with H06.26/J06.15 RVUs, the OBJECTTYPE DISKFILE/VOLUME/SUBVOLUME is granted additional access permissions, WRITE (W) and PURGE (P), along with the existing CREATE (C) and OWNER (O) permissions. Members having the WRITE (W) permission on OBJECTTYPE DISKFILE/VOLUME/SUBVOLUME can modify the respective DISKFILE/VOLUME/SUBVOLUME protection records. Members having the PURGE (P) permission on OBJECTTYPE DISKFILE/VOLUME/SUBVOLUME can purge the respective DISKFILE/VOLUME/SUBVOLUME protection records.

`NO ACCESS CONTROL LIST DEFINED!`

> appears for an object class that has no ACL. Use ALTER OBJECTTYPE...ACCESS to define ACL entries for an existing object-class authorization record.

△ **Caution.** If you do not specify an ACL for an object class, only the local super ID can add an authorization record for an object of that object class.

INFO OBJECTTYPE Detailed Report

The detailed INFO OBJECTTYPE report includes the auditing specifications currently defined for the protected OBJECTTYPE. Figure 12-2 shows the format of the detailed INFO OBJECTTYPE report.

Figure 12-2. INFO OBJECTTYPE Detailed Report Format

```
                LAST-MODIFIED OWNER    STATUS
objecttype
                  date,time   owner-id status

   user-spec [DENY] authority-list
   user-spec [DENY] authority-list
        .
        .
[ NO ACCESS CONTROL LIST DEFINED! ]

 OBJECT-TEXT-DESCRIPTION =

 AUDIT-ACCESS-PASS = a-spec  AUDIT-MANAGE-PASS = a-spec
 AUDIT-ACCESS-FAIL = a-spec  AUDIT-MANAGE-FAIL = a-spec
```

In addition to the OBJECTTYPE attribute values displayed in the brief INFO OBJECTTYPE report, the detailed INFO OBJECTTYPE report displays these attribute values:

```
AUDIT-ACCESS-PASS = a-spec  AUDIT-MANAGE-PASS = a-spec
AUDIT-ACCESS-FAIL = a-spec  AUDIT-MANAGE-FAIL = a-spec
```

These values indicate the conditions under which the Safeguard software audits attempts to create an authorization record for any specific object in this object class, and attempts to manage this authorization record. *`a-spec`* can be:

```
{ ALL | LOCAL | REMOTE | NONE }
```

For a full description of each *`a-spec`*, see the appropriate *`audit-spec`* under the SET OBJECTTYPE Command on page 12-23.

Example

To generate a brief INFO OBJECTTYPE report for the object-class device:

```
=INFO OBJECTTYPE device
```

RESET OBJECTTYPE Command

RESET OBJECTTYPE returns the current default OBJECTTYPE attribute values to their predefined values.

When you add an authorization record for an object class, the current default OBJECTTYPE attribute values are used for any attributes you do not specify with the SET OBJECTTYPE or ADD OBJECTTYPE commands.

```
RESET OBJECTTYPE [ [ , ] objecttype-attribute-keyword ]

   [ , objecttype-attribute-keyword ] ...
```

`objecttype-attribute-keyword`

sets the current default value of the specified attribute to its predefined value. The *`objecttype-attribute-keywords`* and their predefined values are:

```
OWNER                    - The user ID of the current user
ACCESS                   - Null (no ACL)
OBJECT-TEXT-DESCRIPTION - Null (no descriptive text or blank)
AUDIT-ACCESS-PASS        - NONE (no auditing)
AUDIT-ACCESS-FAIL        - NONE (no auditing)
AUDIT-MANAGE-PASS        - NONE (no auditing)
AUDIT-MANAGE-FAIL       - NONE (no auditing)
```

For a complete description of the *`objecttype-attribute`* values, see the SET OBJECTTYPE Command on page 12-23.

Consideration

If you enter RESET OBJECTTYPE but do not include an *`objecttype-attribute-keyword`*, all the object-class attributes return to their predefined values.

Example

To display the current attribute values:

```
=SHOW OBJECTTYPE
```

A brief report shows:

```
TYPE          OWNER
 OBJECTTYPE \*.86,255

 OBJECT-TEXT-DESCRIPTION =

  AUDIT-ACCESS-PASS = ALL        AUDIT-MANAGE-PASS = REMOTE
  AUDIT-ACCESS-FAIL = NONE       AUDIT-MANAGE-FAIL = ALL

         255,255      C,O
      \*.086,255      C,O
         086,*        C,O
```

To restore the default object-class ACL to its predefined value (that is, no ACL):

```
=RESET OBJECTTYPE ACCESS
```

To display the new attribute values:

```
=SHOW OBJECTTYPE
```

A brief report shows:

```
TYPE          OWNER
 OBJECTTYPE \*.86,255

  OBJECT-TEXT-DESCRIPTION =

  AUDIT-ACCESS-PASS = ALL        AUDIT-MANAGE-PASS = REMOTE
  AUDIT-ACCESS-FAIL = NONE       AUDIT-MANAGE-FAIL = ALL

  NO ACCESS CONTROL LIST DEFINED!
```

SET OBJECTTYPE Command

SET OBJECTTYPE establishes default values for one or more object-class attributes. These default values become a template so that when you add an authorization record for an object class, the default values are used for any attributes not specified in your ADD OBJECTTYPE command.

To display the current default OBJECTTYPE attribute values, use the SHOW OBJECTTYPE command.

```
SET OBJECTTYPE [ , ]

   { LIKE objecttype-spec | objecttype-attribute }

   [ , objecttype-attribute ] ...
```

LIKE *objecttype-spec*

sets the current default *objecttype-attribute* values to the existing *objecttype-name-spec* values.

objecttype-spec

identifies a class of objects whose existing attribute values are to become the default *objecttype-attribute* values. *objecttype-spec* can be any object-class name, including OBJECTTYPE:

```
DEVICE
DISKFILE
DISKFILE-PATTERN
SAVED-DISCFILE-PATTERN
OBJECTTYPE
PROCESS
SUBDEVICE
SUBPROCESS
SUBVOLUME
USER
VOLUME
```

objecttype-attribute

defines a default value for the specified object-class attribute. The *objecttype-attribute* values are:

```
OWNER [owner-id]
ACCESS access-spec [ ; access-spec ] ...
OBJECT-TEXT-DESCRIPTION "[any-text]"
AUDIT-ACCESS-PASS [audit-spec]
AUDIT-ACCESS-FAIL [audit-spec]
AUDIT-MANAGE-PASS [audit-spec]
AUDIT-MANAGE-FAIL [audit-spec]
```

OWNER [*owner-id*]

specifies the owner of a class of objects. *owner-id* can be either:

```
[\node-spec.]group-name.member-name
[\node-spec.]group-num , member-num
```

If you omit *owner-id*, *owner-id* is set to your user ID (the user ID of the current SAFECOM user).

ACCESS *access-spec* [; *access-spec*] ...

changes the ACL for *filename-list* by adding or deleting ACL entries or by changing the authority list of a current ACL entry.

An ACL contains as many as 50 entries that grant or deny access authorities to users and user groups.

access-spec has the form:

```
user-list  [-] [DENY] authority-list
```

```
group-list [-] [DENY] authority-list
```

`user-list`

specifies users who are granted (or denied) the access authorities specified with the following *`authority-list`*. *`user-list`* can be either of these:

```
  net-user-spec

( net-user-spec [ , net-user-spec ] ... )
```

`net-user-spec` can be any of:

```
[\node-spec.]adm-group-name.user-name
[\node-spec.]adm-group-num , user-num
[\node-spec.]adm-group-name.*
[\node-spec.]adm-group-num , *
[\node-spec.]*.*
[\node-spec.]*,*
```

-

(minus-sign) operates on existing ACL entries. The minus-sign form of *`access-spec`* modifies the current default ACL. The *`authority`* entries are removed from the default ACL entries for the users specified with *`user-list`*.

`group-list`

can be either:

```
  net-group-spec

( net-group-spec [ , net-user-spec ] ... )
```

`net-group-spec` can be any of:

```
GROUP [NAME][\node-spec.] group-name

GROUP NUMBER [\node-spec.]
```

`node-spec`

takes this form:

```
* | node-name | node-number
```

`node-name`

specifies the system name.

`node-number`

specifies the Expand node number.

adm-group-name

specifies the name of the administrative group.

adm-group-num

specifies the group number of an administrative group.

group-name

specifies the name of any group.

group-num

specifies the group number of any group.

-

(minus-sign) operates on existing ACL entries. The minus-sign form of *access-spec* modifies the current default ACL. The *authority* entries are removed from the default ACL entries for the users specified with *user-list*.

DENY

denies the user IDs or user groups specified with *user-list* the access authorities specified with *authority-list*.

authority-list

specifies the access authorities granted (or denied) to *user-list*. *authority-list* can be any of:

```
  authority
( authority [ , authority ] ... )
  *
```

authority

can be one of the following:

```
C[REATE]
O[WNER]
```

*

(asterisk) specifies CREATE and OWNER.

Note. Starting with H06.24/J06.13 RVUs, the OBJECTTYPE USER is granted additional access permissions, WRITE (W) and PURGE (P), along with the existing CREATE (C) and OWNER (O) permissions. Members having the WRITE (W) permission on OBJECTTYPE USER can modify any subject records. Members having the PURGE (P) permission on OBJECTTYPE USER can purge any subject records.

Note. Starting with H06.26/J06.15 RVUs, the OBJECTTYPE DISKFILE/VOLUME/SUBVOLUME is granted additional access permissions, WRITE (W) and PURGE (P), along with the existing CREATE (C) and OWNER (O) permissions. Members having the WRITE (W) permission on OBJECTTYPE DISKFILE/VOLUME/SUBVOLUME can modify the respective DISKFILE/VOLUME/SUBVOLUME protection records. Members having the PURGE (P) permission on OBJECTTYPE DISKFILE/VOLUME/SUBVOLUME can purge the respective DISKFILE/VOLUME/SUBVOLUME protection records.

`OBJECT-TEXT-DESCRIPTION "[any-text]"`

allows you to store printable characters as comments. These comments are associated with the objects and are used to manage the object authorization record.

The text description field can accommodate 255 bytes of text data.

Note. The text specified in the text description field overwrites existing data, if any. Also, when LIKE clause is used with SET OBJECTTYPE command, the OBJECT-TEXT-DESCRIPTION field is not copied with other object authorization record attributes.

The OBJECT-TEXT-DESCRIPTION attribute is supported only on systems running J06.05 and later J-series RVUs and H06.16 and later H-series RVUs and G06.32 and later G-series RVUs.

`AUDIT-ACCESS-PASS [`*`audit-spec`*`]`

establishes an *audit-spec* for successful attempts to add an authorization record for a specific object. This *audit-spec* specifies the conditions under which an audit record is written to the object-audit file.

The form of *audit-spec* is:

`{ ALL | LOCAL | REMOTE | NONE }`

`ALL`

All successful attempts to add an authorization record are audited.

`LOCAL`

Only successful attempts made by local users to add an authorization record are audited.

`REMOTE`

Only successful attempts made by remote users to add an authorization record are audited.

`NONE`

No successful attempts to add an authorization record are audited.

Omitting *audit-spec* specifies NONE.

`AUDIT-ACCESS-FAIL [`*`audit-spec`*`]`

establishes an *audit-spec* for unsuccessful attempts to add an authorization record for a specific object. This *audit-spec* specifies the conditions under which an audit record is written to the object-audit file.

The form of *audit-spec* is:

```
{ ALL | LOCAL | REMOTE | NONE }
```

`ALL`

All unsuccessful attempts to add an authorization record are audited.

`LOCAL`

Only unsuccessful attempts made by local users to add an authorization record are audited.

`REMOTE`

Only unsuccessful attempts made by remote users to add an authorization record are audited.

`NONE`

No unsuccessful attempts to add an authorization record are audited.

Omitting *audit-spec* specifies NONE.

`AUDIT-MANAGE-PASS [`*`audit-spec`*`]`

establishes an *audit-spec* for successful attempts to manage an OBJECTTYPE authorization record. This *audit-spec* specifies the conditions under which an audit record is written to the audit file when an attempt to manage an OBJECTTYPE authorization record is successful.

The form of *audit-spec* is:

```
{ ALL | LOCAL | REMOTE | NONE }
```

`ALL`

All successful management attempts are audited.

`LOCAL`

Only successful management attempts by local users are audited.

`REMOTE`

Only successful management attempts by remote users are audited.

`NONE`

No successful management attempts are audited.

Omitting *`audit-spec`* specifies NONE.

`AUDIT-MANAGE-FAIL [`*`audit-spec`*`]`

establishes an *`audit-spec`* for unsuccessful attempts to manage an objecttype-authorization record. This *`audit-spec`* specifies the conditions under which an audit record is written to the audit file when an attempt to manage an OBJECTTYPE authorization record fails.

The form of *`audit-spec`* is:

`{ ALL | LOCAL | REMOTE | NONE }`

`ALL`

All unsuccessful management attempts are audited.

`LOCAL`

Only unsuccessful management attempts made by local users are audited.

`REMOTE`

Only unsuccessful management attempts made by remote users are audited.

`NONE`

No unsuccessful management attempts are audited.

Omitting *`audit-spec`* specifies NONE.

Example

These commands define default values for a new object class:

```
=SET OBJECTTYPE OWNER prs.manager
=SET OBJECTTYPE AUDIT-ACCESS-PASS all, &
=AUDIT-MANAGE-PASS local
=SET OBJECTTYPE ACCESS 33,*  (c,o); (86,*, 255,*) *
=SET OBJECTTYPE ACCESS prs.harry DENY *
```

The default object-class attribute values defined here are:

- The object-class owner is the manager of the PRS group.
- The Safeguard software audits all successful attempts to add an authorization record for a specific object, as well as successful local attempts to manage the authorization record for that type.

- All members of groups 33, 86, and 255 can create and own the object type (except user PRS.HARRY, who is specifically denied access to the object type).

SHOW OBJECTTYPE Command

SHOW OBJECTTYPE displays the current default values for the OBJECTTYPE attributes.

```
SHOW [ / OUT listfile / ] OBJECTTYPE
```

`OUT`

directs the SHOW OBJECTTYPE report to *`listfile`*. After it executes the SHOW command, SAFECOM redirects its output to the current OUT file.

`listfile`

For *`listfile`*, specify any file name. SAFECOM opens the *`listfile`* and appends the SHOW OBJECTTYPE report to that file. If *`listfile`* does not exist, SAFECOM creates it as an EDIT-format file.

SHOW OBJECTTYPE Report Format

Figure 12-3 shows the format of the SHOW OBJECTTYPE report.

Figure 12-3. SHOW OBJECTTYPE Report Format

```
TYPE                OWNER
 OBJECTTYPE         gn,un

  OBJECT-TEXT-DESCRIPTION =

  AUDIT-ACCESS-PASS = a-spec   AUDIT-MANAGE-PASS = a-spec
  AUDIT-ACCESS-FAIL = a-spec   AUDIT-MANAGE-FAIL = a-spec

     user-spec [DENY] authority-list
     user-spec [DENY] authority-list
          .        .         .
          .        .         .

   [ NO ACCESS CONTROL LIST DEFINED! ]
```

The SHOW OBJECTTYPE report displays the following attribute values:

`OWNER` *`gn,un`*

is the group number and member number of the user who will own this OBJECTTYPE authorization record.

```
AUDIT-ACCESS-PASS = a-spec  AUDIT-MANAGE-PASS = a-spec
AUDIT-ACCESS-FAIL = a-spec  AUDIT-MANAGE-FAIL = a-spec
```

are the conditions under which the Safeguard software will audit attempts to create authorization records for any specific objects and attempts to manage this authorization record. For more information about these fields for *audit-spec*, see the SET OBJECTTYPE Command on page 12-23.

user-spec [DENY] *authority-list*

is a current default ACL entry for the object class. For a full description, see INFO OBJECTTYPE Brief Report on page 12-19.

`[ NO ACCESS CONTROL LIST DEFINED! ]`

indicates no default ACL entries are defined. Use SET OBJECTTYPE...ACCESS to define default ACL entries. You can use ADD OBJECTTYPE...ACCESS to define ACL entries when you create an authorization record.

△ **Caution.** If you do not specify an ACL for an object class, only the local super ID can add an authorization record for an object of that object class.

Example

This SHOW OBJECTTYPE report displays the current default object type attribute values for a class of objects. To display the report:

```
=SHOW OBJECTTYPE
```

The report shows:

```
TYPE             OWNER
 OBJECTTYPE      255,18

  OBJECT-TEXT-DESCRIPTION =

  AUDIT-ACCESS-PASS = ALL         AUDIT-MANAGE-PASS = NONE
  AUDIT-ACCESS-FAIL = ALL         AUDIT-MANAGE-FAIL = NONE

         033,013      C,O
         033,255      C,O
         255,018      C,O
```

These current default values indicate that:

- The owner of an object type that has these attribute values is the local super-group member with user ID 255,18.
- The Safeguard software audits all successful and unsuccessful attempts to create an authorization record for any specific object belonging to an object class defined by a future ADD OBJECTTYPE command.

- The users with user IDs 33,13 and 255,18 as well as the group manager for group 33 have CREATE and OWNER authorities for the authorization record to be created for this object class.

THAW OBJECTTYPE Command

THAW OBJECTTYPE reenables the ACL for a frozen object class. The authority granted the user is reinstated.

The primary owner, the primary owner's group manager, and the local super ID can thaw a frozen object class. In addition, any user ID with OWNER authority on the ACL can also thaw the object-class authorization record.

THAW OBJECTTYPE has no effect on an object class that is not frozen.

```
THAW OBJECTTYPE objecttype-list
```

objecttype-list

specifies the object classes to be thawed. *objecttype-list* can be either:

```
  objecttype-spec

( objecttype-spec [ , objecttype-spec ] ... )
```

objecttype-spec

can be any object type name, including OBJECTTYPE:

```
DEVICE
DISKFILE
DISKFILE-PATTERN
SAVED-DISCFILE-PATTERN
OBJECTTYPE
PROCESS
SUBDEVICE
SUBPROCESS
SUBVOLUME
USER
VOLUME
```

Example

To thaw the ACL for OBJECTTYPE DEVICE:

```
=THAW OBJECTTYPE DEVICE
```

13 Security Group Commands

Safeguard security group commands allow a security administrator to define security groups of users who can execute certain restricted commands. The security group commands are similar to OBJECTTYPE commands.

Note. In prior product versions, the Safeguard security groups were managed by GROUP commands. GROUP commands are now used to manage file-sharing groups, as described in Section 7, Group Commands. Security groups are now managed with the SECURITY-GROUP commands, as described in this section.

The security groups, SECURITY-ADMINISTRATOR, SYSTEM-OPERATOR, SECURITY-OSS-ADMINISTRATOR, SECURITY-PRV-ADMINISTRATOR, SECURITY-AUDITOR, SECURITY-MEDIA-ADMIN, and SECURITY-PERSISTENCE-ADMIN can be added to the Safeguard database. These security groups do not exist until they are added using the ADD SECURITY-GROUP command.

Note.

- The SECURITY-OSS-ADMINISTRATOR security group is supported only on systems running G06.29 and later G-series RVUs, and H06.08 and later H-series RVUs.
- The SECURITY-PRV-ADMINISTRATOR security group is supported only on systems running J06.11 and later J-series RVUs, and H06.22 and later H-series RVUs.
- The SECURITY-AUDITOR security group is supported only on systems running J06.13 and later J-series RVUs, and H06.24 and later H-series RVUs.
- The SECURITY-MEDIA-ADMIN security group is supported only on systems running J06.15 and later J-series RVUs, and H06.26 and later H-series RVUs.
- The SECURITY-PERSISTENCE-ADMIN security group is supported only on systems running J06.16 and later J-series RVUs, and H06.27 and later H-series RVUs.

Until the security groups are added, all super-group members can execute audit service commands, TERMINAL commands, EVENT-EXIT-PROCESS commands, and the ALTER SAFEGUARD and STOP SAFEGUARD commands. Creating the security groups allows you to restrict use of these commands by designating the specific users who are allowed to execute the commands. After a security group is created, only users with EXECUTE authority on the access control list (ACL) can use the commands restricted to that security group.

Note.

- It is recommended that SUPER.SUPER must not be added to the SECURITY-OSS-ADMINISTRATOR or SECURITY-PRV-ADMINISTRATOR security groups.
- SECURITY-OSS-ADMINISTRATOR or SECURITY-PRV-ADMINISTRATOR security groups must not be added by a SUPER.SUPER, but can be added by any SUPER.*. This prevents SUPER.SUPER from gaining ownership of the security groups.
- SUPER.SUPER must be denied access to SECURITY-OSS-ADMINISTRATOR or SECURITY-PRV-ADMINISTRATOR Security groups using Safeguard ACLs. For example, `alter sec-group sec-prv-admin`, `access super.super deny *`.

Members of the SECURITY-ADMINISTRATOR security group can execute these restricted commands:

```
ALTER SAFEGUARD
STOP SAFEGUARD
ADD EVENT-EXIT-PROCESS
ALTER EVENT-EXIT-PROCESS
DELETE EVENT-EXIT-PROCESS
ADD AUDIT POOL
ALTER AUDIT POOL
ALTER AUDIT SERVICE
DELETE AUDIT POOL
SELECT
ADD TERMINAL
ALTER TERMINAL
DELETE TERMINAL
FREEZE TERMINAL
THAW TERMINAL
```

Members of the SYSTEM-OPERATOR security group can execute these restricted commands:

```
ADD AUDIT POOL
ALTER AUDIT POOL
DELETE AUDIT POOL
NEXTFILE
RELEASE
SELECT
FREEZE TERMINAL
THAW TERMINAL
```

The SECURITY-OSS-ADMINISTRATOR security group designates a list of users that are granted additional OSS security management privileges over normal users for the operations:

```
acl(ACL_SET)
chown(2)
chmod(2)
chdir(2)
opendir(3)
```

Note. Only the locally authenticated users who are part of the SECURITY-OSS-ADMINISTRATOR security group are granted the above specified privileges, not the remotely authenticated users.

Membership in the SECURITY-OSS-ADMINISTRATOR security group are flagged in the user’s environment during initial logon.

The SECURITY-PRV-ADMINISTRATOR security group designates a list of users that are granted additional OSS security management privileges over normal users for the setfilepriv(2) operation.

Membership in the SECURITY-PRV-ADMINISTRATOR security group are flagged in the user’s environment during initial logon.

The SECURITY-AUDITOR security group designates a list of users, who are not SUPER.SUPER, record owner or record owner's group manager to view the subject and group records. Users who are part of this group will have read only privileges for the subject and group records.

The SECURITY-MEDIA-ADMIN security group designates a list of users, who are responsible for management of the tape subsystem and have the permission to execute the tape management commands.

The SECURITY-PERSISTENCE-ADMIN security group designates a list of users who have the same privileges as that of super-group users for managing persistence processes.

Like the ADD OBJECTTYPE command, the ADD SECURITY-GROUP command can be used only by super-group members. Once an authorization record for a security group has been added to the Safeguard database, the record’s primary owner, the owner’s group manager, and any user with OWNER authority on the ACL can use other security group commands to manage the security group authorization record.

Security Group Access Authorities

The ACL defined for a security group can grant either of these access authorities to users and user groups:

EXECUTE Execute the set of commands restricted to the security group

OWNER Manage the security group authorization record

Security Group Command Summary

Table 13-1 lists the SECURITY-GROUP commands and gives a brief description of each.

Table 13-1. Security-Group Command Summary (page 1 of 2)

Command	**Description**
ADD SECURITY-GROUP	Adds a security group authorization record with the specified group attribute values. If you do not specify attribute values, the current defaults are used. Only a member of the local super group can add an authorization record for a security group.
ALTER SECURITY-GROUP	Changes one or more attribute values in a security group authorization record. For all attributes except ACCESS, the ALTER SECURITY-GROUP command replaces the current value with the specified value. For the ACCESS attribute, ALTER SECURITY-GROUP changes the existing ACL to incorporate *access-spec*.
DELETE SECURITY-GROUP	Deletes a security group authorization record. Afterward, only local super-group members can execute the restricted commands.

Table 13-1. Security-Group Command Summary (page 2 of 2)

Command	Description
FREEZE SECURITY-GROUP	Temporarily disables authorities granted to users who have security group access. Only the owners of a security group authorization record, the primary owner’s group manager, and the local super ID can execute the restricted commands.
INFO SECURITY-GROUP	Displays the existing attribute values of a security group authorization record.
RESET SECURITY-GROUP	Sets one or more default security group attribute values to the predefined values of the SET command.
SET SECURITY-GROUP	Sets one or more security group attribute values to specified default values.
SHOW SECURITY-GROUP	Displays the current default values of the security group attributes.
THAW SECURITY-GROUP	Reenables a frozen security group. User IDs with EXECUTE authority on the security group ACL can execute the restricted commands once again.

Syntax of Security Group Commands

The remainder of this section describes each security group command in detail. Commands are presented in alphabetical order, and descriptions contain these elements:

- A summary of the command’s function, including the restrictions on who can use the command
- The command syntax, including descriptions of the command parameters and variables
- The format for any command listing or report
- Considerations for the use of the command
- Examples of command usage

ADD SECURITY-GROUP Command

ADD SECURITY-GROUP creates an authorization record for one or more security groups. Only a member of the local super group can add an authorization record for a security group.

You can specify values for the security group attributes in the ADD SECURITY-GROUP command. The current default values are used for any attributes not specified. These default values are established with the SET command.

```
ADD SECURITY-GROUP sec-group-list [ , ]

   [ LIKE sec-group-spec | sec-group-attribute ]

   [ , sec-group-attribute ] ...
```

sec-group-list

specifies one or more security groups for which an authorization record is to be added. *sec-group-list* can be either:

```
  sec-group-spec

( sec-group-spec [ , sec-group-spec ] ... )
```

sec-group-spec

can be either of:

```
SECURITY-ADMINISTRATOR
SYSTEM-OPERATOR
SECURITY-OSS-ADMINISTRATOR
SECURITY-PRV-ADMINISTRATOR
SECURITY-AUDITOR
SECURITY-MEDIA-ADMIN
SECURITY-PERSISTENCE-ADMIN
```

LIKE *sec-group-spec*

adopts the existing group attribute values of *sec-group-spec* as the attribute values to be used for the authorization record or records being added.

sec-group-spec

identifies the security group whose current *sec-group-attribute* values are to be assigned to the security group authorization record or records being added. *sec-group-spec* can be any security group name.

sec-group-attribute

defines a security group attribute value for the security group authorization record or records being added.

The permitted *sec-group-attribute*s are:

```
OWNER [owner-id]
ACCESS access-spec [ ; access-spec ] ...
OBJECT-TEXT-DESCRIPTION “[any-text]”
AUDIT-ACCESS-PASS [audit-spec]
AUDIT-ACCESS-FAIL [audit-spec]
AUDIT-MANAGE-PASS [audit-spec]
AUDIT-MANAGE-FAIL [audit-spec]
```

`OWNER [`*`owner-id`*`]`

specifies the new owner of this security group authorization record. The *owner-id* can be either:

```
[\node-spec.]group-name.member-name
[\node-spec.]group-num , member-num
```

If you omit *owner-id*, *owner-id* is set to your user ID.

`ACCESS `*`access-spec`*` [ ; `*`access-spec`*` ] ...`

changes the ACL for *filename-list* by adding or deleting ACL entries or by changing the authority list of a current ACL entry.

An ACL contains as many as 50 entries that grant or deny access authorities to users and user groups.

access-spec has the following form:

```
user-list  [-] [DENY] authority-list
group-list [-] [DENY] authority-list
```

user-list

specifies users who are granted (or denied) the access authorities specified with the following *authority-list*. *user-list* can be either:

```
  net-user-spec

( net-user-spec [ , net-user-spec ] ... )
```

net-user-spec can be any of:

```
[\node-spec.]adm-group-name.user-name
[\node-spec.]adm-group-num , user-num
[\node-spec.]adm-group-name.*
[\node-spec.]adm-group-num , *
[\node-spec.]*.*
[\node-spec.]*,*
```

-

(minus-sign) operates on existing ACL entries. The minus-sign form of *access-spec* modifies the current default ACL. The *authority* entries are removed from the default ACL entries for the users specified with *user-list*.

group-list

can be either:

net-group-spec

`(` *net-group-spec* `[ ,` *net-user-spec* `] ... )`

net-group-spec can be any of:

`GROUP [NAME][\`*node-spec*`.]` *group-name*e

`GROUP NUMBER [\`*node-spec*`]`

node-spec

takes this form:

`* |` *node-name* `|` *node-number*

node-name

specifies the system name.

node-number

specifies the Expand node number.

adm-group-name

specifies the name of the administrative group.

adm-group-num

specifies the group number of an administrative group.

group-name

specifies the name of any group.

group-num

specifies the group number of any group.

-

(minus-sign) operates on existing ACL entries. The minus-sign form of *access-spec* modifies the current default ACL. The *authority* entries are removed from the default ACL entries for the users specified with *user-list*.

DENY

denies the user IDs or user groups specified by *user-list* the access authorities specified by *authority-list*.

authority-list

specifies the access authorities to be granted (or denied) to *user-list*. *authority-list* can be either:

```
  authority

( authority [ , authority ] ... )
```

authority

is either:

```
E[XECUTE]
O[WNER]
```

`OBJECT-TEXT-DESCRIPTION “[any-text]”`

allows you to store printable characters as comments. These comments are associated with the objects and are used to manage the object authorization record.

The text description field can accommodate 255 bytes of text data.

Note. The text specified in the text description field overwrites existing data, if any. Also, when LIKE clause is used with ADD SECURITY-GROUP command, the OBJECT-TEXT-DESCRIPTION field is not copied with other object authorization record attributes.

The OBJECT-TEXT-DESCRIPTION attribute is supported only on systems running J06.05 and later J-series RVUs and H06.16 and later H-series RVUs and G06.32 and later G-series RVUs.

`AUDIT-ACCESS-PASS [`*audit-spec*`]`

changes the *audit-spec* for successful attempts to execute a restricted command. You need not specify AUDIT-ACCESS-PASS because the Safeguard software automatically audits all attempts to execute restricted commands.

`AUDIT-ACCESS-FAIL [`*audit-spec*`]`

changes the *audit-spec* for unsuccessful attempts to execute a restricted command. You need not specify AUDIT-ACCESS-FAIL because the Safeguard software automatically audits all attempts to execute restricted commands.

`AUDIT-MANAGE-PASS [`*audit-spec*`]`

changes the *audit-spec* for successful attempts to manage this authorization record. The form of *audit-spec* is:

```
{ ALL | LOCAL | REMOTE | NONE }
```

For a description of the *audit-spec* variables, see the SET SECURITY-GROUP Command on page 13-25. Omitting *audit-spec* specifies NONE.

`AUDIT-MANAGE-FAIL [`*`audit-spec`*`]`

changes the *`audit-spec`* for unsuccessful attempts to manage this authorization record. The form of *`audit-spec`* is:

```
{ ALL | LOCAL | REMOTE | NONE }
```

For a description of the *`audit-spec`*s, see the SET SECURITY-GROUP Command on page 13-25. Omitting *`audit-spec`* specifies NONE.

Note. Specifying ACCESS *`access-spec`* with ADD SECURITY-GROUP does not override the current default ACL (established with SET SECURITY-GROUP). Instead, any ACL entries specified with ADD SECURITY-GROUP modify the template of current default settings.

For a complete description of the *`group-attribute`*s, see the SET SECURITY-GROUP Command on page 13-25.

Considerations

- Additional owners can modify the authorization record.

  In addition to the owner, the primary owner’s group manager, and the local super ID, any user ID that has an ACL entry granting OWNER authority can also modify the security group authorization record.

- Attributes in an ADD command affect only the record added.

  Any attribute specifications in an ADD SECURITY-GROUP command affect only the authorization record being created and do not change the current default group attribute values. This condition is also true for a LIKE clause in an ADD SECURITY-GROUP command.

Example

You can use a LIKE *`sec-group-name`* clause to copy all attribute values for one security group from another security group. Then you can specify in the same command line that one or more attribute values will be different.

This sample command adds an authorization record for the SYSTEM-OPERATOR security group that has the same group attribute values (and ACL) as the SECURITY-ADMINISTRATOR security group, except for the OWNER attribute. It also allows you to add object text description:

```
=ADD SECURITY-GROUP sys-oper, OBJECT-TEXT-DESCRIPTION “Added a &
record”,LIKE sec-admin, OWNER super.sue
```

You can define membership in the SECURITY-OSS-ADMINISTRATOR security group by adding an authorization record for that group. For example, this command creates the authorization record for the SECURITY-OSS-ADMINISTRATOR security group and allows you to enter object text description:

```
=ADD SECURITY-GROUP SECURITY-OSS-ADMINISTRATOR,  &
OBJECT-TEXT-DESCRIPTION “Added a record”,&
```

```
OWNER SUPER.TEST,&
AUDIT-ACCESS NONE,&
AUDIT-MANAGE-PASS ALL,&
ACCESS TEST1.USER1 (E,O); TEST1.USER2 (E); TEST1.USER3(O)
```

You can add the SECURITY-PRV-ADMINISTRATOR security group protection record:

```
=ADD SECURITY-GROUP SECURITY-PRV-ADMINISTRATOR, ACCESS SECGRP.*
(E)
```

You can add the SECURITY-AUDITOR security group protection record:

```
=ADD SECURITY-GROUP SECURITY-AUDITOR, ACCESS SECGRP.* (E)
```

You can add the SECURITY-MEDIA-ADMIN security group protection record:

```
=ADD SECURITY-GROUP SECURITY-MEDIA-ADMIN, OWNER SUPER.SUPER,
ACCESS 255,* *
```

You can add the SECURITY-PERSISTENCE-ADMIN security group protection record:

```
=ADD SECURITY-GROUP SECURITY-PERSISTENCE-ADMIN, OWNER
SUPER.SUPER, ACCESS 255,* *
```

ALTER SECURITY-GROUP Command

ALTER SECURITY-GROUP changes one or more attribute values in a security group authorization record. Both the owner and the primary owner’s group manager can change a security group authorization record. In addition, any user ID that has an ACL entry granting it OWNER authority can also modify the security group authorization record.

Except for the ACCESS attribute, new group attribute values specified in an ALTER SECURITY-GROUP command replace the existing attribute values. Specifying a new ACCESS *access-spec* adds the new *access-spec* to the security group’s existing ACL. To remove authorities previously granted to user IDs, use the minus-sign (-) form of *access-spec*.

```
ALTER SECURITY-GROUP sec-group-list [ , ]

   { LIKE sec-group-spec | sec-group-attribute  }

   [ , sec-group-attribute ] ...
```

sec-group-list

specifies one or more security groups whose existing *sec-group-attribute* values are to be changed. All security groups specified must already have Safeguard authorization records (created with the ADD SECURITY-GROUP command).

sec-group-list can be either:

sec-group-spec

(*sec-group-spec* [, *sec-group-spec*] ...)

sec-group-spec

can be either:

```
SECURITY-ADMINISTRATOR
SYSTEM-OPERATOR
SECURITY-OSS-ADMINISTRATOR
SECURITY-PRV-ADMINISTRATOR
SECURITY-AUDITOR
SECURITY-MEDIA-ADMIN
SECURITY-PERSISTENCE-ADMIN
```

LIKE *sec-group-spec*

changes the attribute values of *sec-group-list* to the same as the existing attribute values for *sec-group-spec*. For the ACCESS attribute, LIKE adds ACL entries or adds authorities only to existing entries. It does not replace or delete ACL entries or authorities.

sec-group-spec

identifies the security group whose existing *sec-group-attribute* values are to be assigned to the security group authorization record being changed. *sec-group-spec* can be any valid security group name.

sec-group-attribute

changes the existing value of the specified group attribute for the security group being changed. The *sec-group-attribute* values are:

OWNER [*owner-id*]
ACCESS *access-spec* [; *access-spec*] ...
OBJECT-TEXT-DESCRIPTION “[any-text]”
RESET-OBJECT-TEXT-DESCRIPTION
AUDIT-ACCESS-PASS [*audit-spec*]
AUDIT-ACCESS-FAIL [*audit-spec*]
AUDIT-MANAGE-PASS [*audit-spec*]
AUDIT-MANAGE-FAIL [*audit-spec*]

OWNER [*owner-id*]

specifies the new owner of the security group authorization record. The *owner-id* can be either:

[\node-spec.]*group-name*.*member-name*
[\node-spec.]*group-num* , *member-num*

If you omit *owner-id*, *owner-id* is set to your user ID.

`ACCESS` *`access-spec`* `[ ;` *`access-spec`* `] ...`

changes the ACL for *`filename-list`* by adding or deleting ACL entries or by changing the authority list of a current ACL entry.

An ACL contains as many as 50 entries that grant or deny access authorities to users and user groups.

`access-spec` has the form:

```
user-list  [-] [DENY] authority-list
group-list [-] [DENY] authority-list
```

`user-list`

specifies users who are granted (or denied) the access authorities specified with the following *`authority-list`*. *`user-list`* can be either:

```
  net-user-spec

( net-user-spec [ , net-user-spec ] ... )
```

`net-user-spec` can be any of:

```
[\node-spec.]adm-group-name.user-name
[\node-spec.]adm-group-num , user-num
[\node-spec.]adm-group-name.*
[\node-spec.]adm-group-num , *
[\node-spec.]*.*
[\node-spec.]*,*
```

`-`

(minus-sign) operates on existing ACL entries. The minus-sign form of *`access-spec`* modifies the current default ACL. The *`authority`* entries are removed from the default ACL entries for the users specified with *`user-list`*.

`group-list`

can be either:

```
  net-group-spec

( net-group-spec [ , net-user-spec ] ... )
```

`net-group-spec` can be any of:

```
GROUP [NAME][\node-spec.] group-name

GROUP NUMBER [\node-spec.]
```

`node-spec`

takes this form:

```
* | node-name | node-number
```

node-name

specifies the system name.

node-number

specifies the Expand node number.

adm-group-name

specifies the name of the administrative group.

adm-group-num

specifies the group number of an administrative group.

group-name

specifies the name of any group.

group-num

specifies the group number of any group.

-

(minus-sign) operates on existing ACL entries. The minus-sign form of *access-spec* modifies the current default ACL. The *authority* entries are removed from the default ACL entries for the users specified with *user-list*.

DENY

denies the user IDs or user groups specified by *user-list* the access authorities specified by *authority-list*.

authority-list

specifies the access authorities to be granted (or denied) to *user-list*. *authority-list* can be either:

```
  authority

( authority [ , authority ] ... )
```

authority

is either:

```
E[XECUTE]
O[WNER]
```

`OBJECT-TEXT-DESCRIPTION “[any-text]”`

allows you to store printable characters as comments. These comments are associated with the objects and are used to manage the object authorization record.

The text description field can accommodate 255 bytes of text data.

Note. The text specified in the text description field overwrites existing data, if any. Also, when LIKE clause is used with ALTER SECURITY-GROUP command, the OBJECT-TEXT-DESCRIPTION field is not copied with other object authorization record attributes. Also, if you specify OBJECT-TEXT-DESCRIPTION without any text in the quotation marks, the object text description for this record is removed.

The OBJECT-TEXT-DESCRIPTION attribute is supported only on systems running J06.05 and later J-series RVUs and H06.16 and later H-series RVUs and G06.32 and later G-series RVUs.

`RESET-OBJECT-TEXT-DESCRIPTION`

Resets the object description to Null.

Note. The RESET-OBJECT-TEXT-DESCRIPTION attribute is supported only on systems running J06.05 and later J-series RVUs, H06.16 and later H-series RVUs, and G06.32 and later G-series RVUs.

`AUDIT-ACCESS-PASS [`*`audit-spec`*`]`

changes the *audit-spec* for successful attempts to execute restricted commands. You need not specify AUDIT-ACCESS-PASS because all attempts to execute restricted commands are audited automatically.

`AUDIT-ACCESS-FAIL [`*`audit-spec`*`]`

changes the *audit-spec* for unsuccessful attempts to execute restricted commands. You need not specify AUDIT-ACCESS-FAIL because all attempts to execute restricted commands are audited automatically.

`AUDIT-MANAGE-PASS [`*`audit-spec`*`]`

changes the *audit-spec* for successful attempts to manage this authorization record. The form of *audit-spec* is:

```
{ ALL | LOCAL | REMOTE | NONE }
```

For a description of the *audit-spec*s, see the SET SECURITY-GROUP Command on page 13-25. Omitting *audit-spec* specifies NONE.

AUDIT-MANAGE-FAIL [*audit-spec*]

changes the *audit-spec* for unsuccessful attempts to manage this authorization record. The form of *audit-spec* is:

```
{ ALL | LOCAL | REMOTE | NONE }
```

For a description of the *audit-spec*s, see the SET SECURITY-GROUP Command on page 13-25. Omitting *audit-spec* specifies NONE.

Example

This command transfers ownership of the SECURITY-ADMINISTRATOR security group to the user with user ID 12,4 and allows all users who are members of group number 12 to execute the commands restricted to the SECURITY-ADMINISTRATOR security group. It also allows you to enter text description:

```
=ALTER SECURITY-GROUP sec-admin, OBJECT-TEXT-DESCRIPTION &
“Record altered”,OWNER 12,4, ACCESS 12,* e
```

Ownership of a group authorization record can be transferred to another user by the ALTER command. For example,

```
=ALTER SECURITY-GROUP SECURITY-OSS-ADMINISTRATOR, &
ACCESS TEST1.USER1 – (E); TEST1.USER4 (E), &
AUDIT-MANAGE ALL
```

To alter the SECURITY-PRV-ADMINISTRATOR security group to enable auditing of pass or fail protection record management operations:

```
=ALTER SECURITY-GROUP SECURITY-PRV-ADMINISTRATOR, AUDIT-MANAGE
ALL
```

To alter the SECURITY-AUDITOR security group to enable auditing of pass or fail protection record management operations:

```
=ALTER SECURITY-GROUP SECURITY-AUDITOR, AUDIT-MANAGE ALL
```

To alter the SECURITY-MEDIA-ADMIN security group protection record, use the following command:

```
=ALTER SECURITY-GROUP SECURITY-MEDIA-ADMIN, ACCESS TEST.MGR E
```

To alter the SECURITY-PERSISTENCE-ADMIN security group protection record, use the following command:

```
=ALTER SECURITY-GROUP SECURITY-PERSISTENCE-ADMIN, ACCESS
TEST.USER1 E
```

DELETE SECURITY-GROUP Command

DELETE SECURITY-GROUP deletes a security group authorization record. After a security group authorization record is deleted, members of the local super group are the only users who can execute the commands restricted to that security group.

```
DELETE SECURITY-GROUP sec-group-list
```

sec-group-list

specifies one or more security groups for which authorization records are to be deleted. *sec-group-list* can be either:

```
  sec-group-spec

( sec-group-spec [ , sec-group-spec ] ... )
```

sec-group-spec

can be either:

```
SECURITY-ADMINISTRATOR
SYSTEM-OPERATOR
SECURITY-OSS-ADMINISTRATOR
SECURITY-PRV-ADMINISTRATOR
SECURITY-AUDITOR
SECURITY-MEDIA-ADMIN
SECURITY-PERSISTENCE-ADMIN
```

Example

As owner of the SYSTEM-OPERATOR security group, you can enter the command to delete the Safeguard authorization record for that security group:

```
=DELETE SECURITY-GROUP system-operator
```

To delete the SECURITY-OSS-ADMINISTRATOR security group, use the DELETE SECURITY-GROUP command. For example, this command deletes the SECURITY-OSS-ADMINISTRATOR security group:

```
=DELETE SECURITY-GROUP SECURITY-OSS-ADMINISTRATOR
```

To delete the SECURITY-PRV-ADMINISTRATOR security group, use the following command:

```
=DELETE SECURITY-GROUP SECURITY-PRV-ADMINISTRATOR
```

To delete the SECURITY-AUDITOR security group, use the following command:

```
=DELETE SECURITY-GROUP SECURITY-AUDITOR
```

To delete the SECURITY-MEDIA-ADMIN security group protection record, use the following command:

```
=DELETE SECURITY-GROUP SECURITY-MEDIA-ADMIN
```

To delete the SECURITY-PERSISTENCE-ADMIN security group protection record, use the following command:

```
=DELETE SECURITY-GROUP SECURITY-PERSISTENCE-ADMIN
```

FREEZE SECURITY-GROUP Command

FREEZE SECURITY-GROUP temporarily suspends the authorities granted to user IDs listed on a security group ACL. While the security group is frozen, only the primary owner, the primary owner’s group manager, an owner on the ACL, and the local super ID can execute the commands restricted to that security group.

Use the THAW SECURITY-GROUP command to reenable all the ACL authorities granted to user IDs before the security group was frozen.

```
FREEZE SECURITY-GROUP sec-group-list
```

sec-group-list

specifies the security group for which access is to be frozen. *sec-group-list* can be either:

```
  sec-group-spec

( sec-group-spec [ , sec-group-spec ] ... )
```

sec-group-spec

can be either:

```
SECURITY-ADMINISTRATOR
SYSTEM-OPERATOR
SECURITY-OSS-ADMINISTRATOR
SECURITY-PRV-ADMINISTRATOR
SECURITY-AUDITOR
SECURITY-MEDIA-ADMIN
SECURITY-PERSISTENCE-ADMIN
```

Consideration

While a security group is frozen, the primary owner, the primary owner’s group manager, and an owner on the ACL are implicitly granted all access authorities. The local super ID also retains ownership.

Example

To disable access authorities granted to members of the SECURITY-ADMINISTRATOR security group, enter this command:

```
=FREEZE SECURITY-GROUP sec-admin
```

The SECURITY-OSS-ADMINISTRATOR security group can be frozen by the primary owner or by any user with OWNER authority on the access control list for the group. For example,

```
=FREEZE SECURITY-GROUP SECURITY-OSS-ADMINISTRATOR
```

To freeze the SECURITY-PRV-ADMINISTRATOR security group, use the following command:

```
=FREEZE SECURITY-GROUP SECURITY-PRV-ADMINISTRATOR
```

To freeze the SECURITY-AUDITOR security group, use the following command:

```
=FREEZE SECURITY-GROUP SECURITY-AUDITOR
```

To freeze the SECURITY-MEDIA-ADMIN security group protection record, use the following command:

```
=FREEZE SECURITY-GROUP SECURITY-MEDIA-ADMIN
```

To freeze the SECURITY-PERSISTENCE-ADMIN security group protection record, use the following command:

```
=FREEZE SECURITY-GROUP SECURITY-PERSISTENCE-ADMIN
```

INFO SECURITY-GROUP Command

INFO SECURITY-GROUP displays the attribute values currently stored in a security group authorization record and produces two types of reports: brief and detailed. The format of each report is illustrated after these syntax description.

Any user can produce an INFO report on any security group.

```
INFO [ / OUT listfile / ] SECURITY-GROUP [ , ] sec-group-list

   [ [ , ] DETAIL ]
```

`OUT`

directs the INFO SECURITY-GROUP report to *`listfile`*. After it executes the INFO command, SAFECOM redirects its output to the current OUT file.

`listfile`

For *`listfile`*, specify any file name. SAFECOM opens *`listfile`* and appends the INFO report to the file. If *`listfile`* does not exist, SAFECOM creates it as an EDIT-format file.

sec-group-list

specifies the security group for which INFO reports are to be produced. *sec-group-list* can be either:

```
  sec-group-spec

( sec-group-spec [ , sec-group-spec ] ... )
```

sec-group-spec

can be either:

```
SECURITY-ADMINISTRATOR
SYSTEM-OPERATOR
SECURITY-OSS-ADMINISTRATOR
SECURITY-PRV-ADMINISTRATOR
SECURITY-AUDITOR
SECURITY-MEDIA-ADMIN
SECURITY-PERSISTENCE-ADMIN
```

DETAIL

adds the *audit-spec*s defined for the security group to the INFO report. For a full description of the four *audit-spec*s, see the SET SECURITY-GROUP Command on page 13-25.

INFO SECURITY-GROUP Brief Report

The brief INFO SECURITY-GROUP report displays the attribute values currently stored for this security group. Figure 13-1 on page 13-19 shows the format of the brief INFO SECURITY-GROUP report.

Figure 13-1. INFO SECURITY-GROUP Brief Report Format

```
                LAST-MODIFIED OWNER    STATUS
sec-group
                 date, time   owner-id status

   user-spec [DENY] authority-list
   user-spec [DENY] authority-list
      .
      .
      .
[ NO ACCESS CONTROL LIST DEFINED! ]
```

Figure 13-1 contains these SECURITY-GROUP attribute values and status fields:

sec-group

is the name of the security group whose existing attribute values are being displayed.

```
LAST MODIFIED TIME
date, time
```

is the date and time of the last change made to this security group authorization record. *date* and *time* are in local civil time.

```
OWNER
owner-id
```

is the user ID of the person who owns this security group authorization record.

```
STATUS
status
```

is the current status of this security group. *status* is either FROZEN or THAWED.

```
user-spec [DENY] authority-list
```

is an entry in the ACL defined for this security group. *user-spec* identifies a single user or user group. *authority-list* is a list of single-character codes that represent the access authorities granted to the user or user group identified by *user-spec*. DENY indicates that the access authorities specified with *authority-list* are specifically denied to the user or user group identified by *user-spec*.

user-spec can be any of:

```
group-num, member-num
group-num, *
*,*
\node-spec.group-num, member-num
\node-spec.group-num, *
\node-spec.*,*
```

group-num, *member-num* identifies a single local user.

group-num,* identifies all the local users in the group that has *group-num*.

, identifies all the local users.

\node-spec.*group-num*, *member-num* identifies both the local user with user ID *group-num*, *member-num* and a network user with the same user name and user ID as that local user.

\node-spec.*group-num*,* identifies all the local users in the group identified by *group-num* and all network users whose *group-num* and *group-name* match those of the local group.

\node-spec.*,* identifies all local users as all network users with access to the local node.

authority-list for this object type can contain either of these codes:

```
E - EXECUTE authority
O - OWNER authority
```

```
NO ACCESS CONTROL LIST DEFINED!
```

appears for a security group that has no ACL. Use ALTER SECURITY-GROUP . . . ACCESS to define ACL entries for an existing security group authorization record.

△ **Caution.** If you do not specify an ACL for a security group, only the local super ID can execute commands restricted to that security group.

INFO SECURITY-GROUP Detailed Report

The detailed INFO SECURITY-GROUP report includes the auditing specifications currently defined for the security group. Figure 13-2 shows the format of the detailed INFO SECURITY-GROUP report.

Figure 13-2. INFO SECURITY-GROUP Detailed Report Format

```
                LAST-MODIFIED OWNER    STATUS
sec-group
                 date, time   owner-id status

   user-spec [DENY] authority-list
   user-spec [DENY] authority-list
        .
        .
[ NO ACCESS CONTROL LIST DEFINED! ]


 OBJECT-TEXT-DESCRIPTION =

 AUDIT-ACCESS-PASS = a-spec  AUDIT-MANAGE-PASS = a-spec
 AUDIT-ACCESS-FAIL = a-spec  AUDIT-MANAGE-FAIL = a-spec
```

In addition to the security group attribute values displayed in the brief INFO SECURITY-GROUP report, the detailed INFO SECURITY-GROUP report displays these attribute values:

```
AUDIT-ACCESS-PASS = a-spec  AUDIT-MANAGE-PASS = a-spec
AUDIT-ACCESS-FAIL = a-spec  AUDIT-MANAGE-FAIL = a-spec
```

These values indicate the conditions under which the Safeguard software audits attempts to execute a restricted command and attempts to manage this authorization record. *a-spec* can be:

```
{ ALL | LOCAL | REMOTE | NONE }
```

For a full description of each *a-spec*, see the appropriate *audit-spec* under the SET SECURITY-GROUP Command on page 13-25.

Example

To generate a brief INFO SECURITY-GROUP report for the group SECURITY-ADMINISTRATOR:

```
=INFO SECURITY-GROUP security-administrator
```

The report shows:

```
                 LAST-MODIFIED      OWNER      STATUS
SECURITY-ADMINISTRATOR
                 18AUG86, 17:28   \*.86,255    THAWED

        086,002 DENY E,O
        033,*        E,O
        086,*        E,O
        255,*        E,O
```

The report shows that:

- The owner of this security group authorization record is a network user who is the manager for group 86 (with user ID 86,255).
- All users who are members of group number 33 or 255 are granted both EXECUTE and OWNER authority for the security group SECURITY-ADMINISTRATOR.
- All users who are members of group number 86, with one exception, are granted both EXECUTE and OWNER authority for this security group. User ID 86,2 is specifically denied both EXECUTE and OWNER authority.

The output of the INFO command is influenced by the session command, DISPLAY, in particular the setting of DISPLAY USER AS and DISPLAY AS COMMANDS.

To display the SECURITY-OSS-ADMINISTRATOR security group protection record:

```
=DISPLAY USER AS NAME
```

To verify the results:

```
=INFO SECURITY-GROUP SECURITY-OSS-ADMINISTRATOR
```

```
                  LAST-MODIFIED     OWNER             STATUS
SECURITY-OSS-ADMINISTRATOR
                  1FEB05, 13:20     SUPER.SUPER       THAWED
```

To display the SECURITY-PRV-ADMINISTRATOR security group protection record:

```
=DISPLAY USER AS NAME
```

To verify the results:

```
=INFO SECURITY-GROUP SECURITY-PRV-ADMINISTRATOR
```

```
                  LAST-MODIFIED     OWNER             STATUS
SECURITY-PRV-ADMINISTRATOR
                  26NOV10, 18:14     SUPER.SUPER       THAWED
```

To display the SECURITY-AUDITOR security group protection record:

```
=DISPLAY USER AS NAME
```

To verify the results:

```
=INFO SECURITY-GROUP SECURITY-AUDITOR
```

```
                  LAST-MODIFIED     OWNER             STATUS
SECURITY-AUDITOR
                  1MAY10, 13:20     SUPER.SUPER       THAWED

GROUP  \*.SEC2                                        E
```

To display the SECURITY-MEDIA-ADMIN security group protection record, use the following command:

```
=DISPLAY USER AS NAME
```

To verify the results:

```
=INFO SECURITY-GROUP SECURITY-MEDIA-ADMIN
```

```
                  LAST-MODIFIED     OWNER            STATUS
SECURITY-MEDIA-ADMIN
                  13JUL12, 13:20    SUPER.SUPER      THAWED

GROUP_MEDIA1.USER1                           DENY    E,O
GROUP GROUP_MEDIA2                                   E,O
GROUP GROUP_MEDIA1                                   E,O
```

To display the SECURITY-PERSISTENCE-ADMIN security group protection record, use the following command:

```
=DISPLAY USER AS NAME
```

To verify the results:

```
=INFO SECURITY-GROUP SECURITY-PERSISTENCE-ADMIN
```

```
                  LAST-MODIFIED       OWNER          STATUS
SECURITY-PERSISTENCE-ADMIN
                 18MAR13, 18:23      SUPER.SUPER     THAWED

                 TEST.USER1                          E
 GROUP           SUPER                               E,   O
```

RESET SECURITY-GROUP Command

RESET SECURITY-GROUP returns the default group attribute values to their predefined values.

When you add an authorization record for a security group, the current default group attribute values are used for any attributes you do not specify with the SET SECURITY-GROUP or ADD SECURITY-GROUP commands.

```
RESET SECURITY-GROUP [ [ , ] sec-group-attribute-keyword ]

   [ , sec-group-attribute-keyword ] ...
```

group-attribute-keyword

sets the current default value of the specified attribute to its predefined value. The *group-attribute-keyword*s and their predefined values are:

OWNER	The user ID of the current user
ACCESS	Null (no ACL)
OBJECT-TEXT-DESCRIPTION	Null (no descriptive text or blank)
AUDIT-ACCESS-PASS	NONE (no auditing)
AUDIT-ACCESS-FAIL	NONE (no auditing)
AUDIT-MANAGE-PASS	NONE (no auditing)
AUDIT-MANAGE-FAIL	NONE (no auditing)

For a complete description of the *group-attribute* values, see the SET SECURITY-GROUP Command.

Consideration

If you enter RESET SECURITY-GROUP but do not include a *sec-group-attribute-keyword*, all the security group attributes return to their predefined values.

Example

To display the current attribute values:

```
=SHOW SECURITY-GROUP
```

A brief report shows:

```
TYPE                     OWNER
 SECURITY-GROUP    \*.86,255

  OBJECT-TEXT-DESCRIPTION =

  AUDIT-ACCESS-PASS = ALL        AUDIT-MANAGE-PASS = REMOTE
  AUDIT-ACCESS-FAIL = NONE       AUDIT-MANAGE-FAIL = ALL

         255,255      E,O
     \*.086,255      E,O
         086,*        E,O
```

To restore the default group ACL to its predefined value (that is, no ACL):

```
=RESET SECURITY-GROUP ACCESS
```

To display the new attribute values:

```
=SHOW SECURITY-GROUP
```

A brief report shows:

```
TYPE                 OWNER
 SECURITY-GROUP    \*.86,255

  OBJECT-TEXT-DESCRIPTION =

  AUDIT-ACCESS-PASS = ALL        AUDIT-MANAGE-PASS = REMOTE
  AUDIT-ACCESS-FAIL = NONE       AUDIT-MANAGE-FAIL = ALL

 NO ACCESS CONTROL LIST DEFINED!
```

SET SECURITY-GROUP Command

SET SECURITY-GROUP establishes default values for one or more security group attributes. These default values become a template so that when you add an authorization record for a security group, the default values are used for any attributes not specified in your ADD SECURITY-GROUP command.

To display the current default security group attribute values, use the SHOW SECURITY-GROUP command.

```
SET SECURITY-GROUP [ , ]

   { LIKE sec-group-spec | sec-group-attribute }

   [ , sec-group-attribute ] ...
```

`LIKE` *`sec-group-spec`*

sets the current default *`sec-group-attribute`* values to the existing *`sec-group-spec`* values.

`sec-group-spec`

can be either:

```
SECURITY-ADMINISTRATOR
SYSTEM-OPERATOR
SECURITY-OSS-ADMINISTRATOR
SECURITY-PRV-ADMINISTRATOR
SECURITY-AUDITOR
SECURITY-MEDIA-ADMIN
SECURITY-PERSISTENCE-ADMIN
```

sec-group-attribute

defines a default value for the specified group attribute. The *sec-group-attribute* values are:

```
OWNER [owner-id]
ACCESS access-spec [ ; access-spec ] ...
OBJECT-TEXT-DESCRIPTION "[any-text]"
AUDIT-ACCESS-PASS [audit-spec]
AUDIT-ACCESS-FAIL [audit-spec]
AUDIT-MANAGE-PASS [audit-spec]
AUDIT-MANAGE-FAIL [audit-spec]
```

OWNER [*owner-id*]

specifies the owner of a security group. *owner-id* can be either:

```
[\node-spec.]group-name.member-name
[\node-spec.]group-num , member-num
```

If you omit *owner-id*, *owner-id* is set to your user ID (the user ID of the current SAFECOM user).

ACCESS *access-spec* [; *access-spec*] ...

changes the ACL for *filename-list* by adding or deleting ACL entries or by changing the authority list of a current ACL entry.

An ACL contains as many as 50 entries that grant or deny access authorities to users and user groups.

access-spec has the form:

```
user-list  [-] [DENY] authority-list
```

```
group-list [-] [DENY] authority-list
```

user-list

specifies users who are granted (or denied) the access authorities specified with the following *authority-list*. *user-list* can be either:

```
  net-user-spec
```

```
( net-user-spec [ , net-user-spec ] ... )
```

net-user-spec can be any of:

```
[\node-spec.]adm-group-name.user-name
[\node-spec.]adm-group-num , user-num
[\node-spec.]adm-group-name.*
[\node-spec.]adm-group-num , *
[\node-spec.]*.*
[\node-spec.]*,*
```

-

(minus-sign) operates on existing ACL entries. The minus-sign form of *access-spec* modifies the current default ACL. The *authority* entries are removed from the default ACL entries for the users specified with *user-list*.

group-list

can be either:

net-group-spec

(*net-group-spec* [, *net-user-spec*] ...)

net-group-spec can be any of:

GROUP [NAME][\node-spec.] *group-name*e

GROUP NUMBER [\node-spec.]

node-spec

takes this form:

* | *node-name* | *node-number*

node-name

specifies the system name.

node-number

specifies the Expand node number.

adm-group-name

specifies the name of the administrative group.

adm-group-num

specifies the group number of an administrative group.

group-name

specifies the name of any group.

group-num

specifies the group number of any group.

-

(minus-sign) operates on existing ACL entries. The minus-sign form of *access-spec* modifies the current default ACL. The *authority* entries

are removed from the default ACL entries for the users specified with *`user-list`*.

`DENY`

denies the user IDs or user groups specified with *`user-list`* the access authorities specified with *`authority-list`*.

`authority-list`

specifies the access authorities granted (or denied) to *`user-list`*. *`authority-list`* can be any of:

```
  authority

( authority [ , authority ] ... )

  *
```

`authority`

can be either:

```
E[XECUTE]
O[WNER]
```

`*`

(asterisk) specifies both EXECUTE and OWNER.

`OBJECT-TEXT-DESCRIPTION "[any-text]"`

allows you to store printable characters as comments. These comments are associated with the objects and are used to manage the object authorization record.

The text description field can accommodate 255 bytes of text data.

Note. The text specified in the text description field overwrites existing data, if any. Also, when LIKE clause is used with SET SECURITY-GROUP command, the OBJECT-TEXT-DESCRIPTION field is not copied with other object authorization record attributes.

The OBJECT-TEXT-DESCRIPTION attribute is supported only on systems running J06.05 and later J-series RVUs and H06.16 and later H-series RVUs and G06.32 and later G-series RVUs.

`AUDIT-ACCESS-PASS [`*`audit-spec`*`]`

establishes an *`audit-spec`* for successful attempts to execute a restricted command. You need not specify AUDIT-ACCESS-PASS because the Safeguard software automatically audits all attempts to execute restricted commands.

`AUDIT-ACCESS-FAIL [`*`audit-spec`*`]`

establishes an *`audit-spec`* for unsuccessful attempts to execute a restricted command. You need not to specify AUDIT-ACCESS-FAIL because the Safeguard software automatically audits all attempts to execute restricted commands.

`AUDIT-MANAGE-PASS [`*`audit-spec`*`]`

establishes an *`audit-spec`* for successful attempts to manage a security group-authorization record. This *`audit-spec`* specifies the conditions under which an audit record is written to the audit file when an attempt to manage a security group authorization record is successful.

The form of *`audit-spec`* is:

```
{ ALL | LOCAL | REMOTE | NONE }
```

`ALL`

All successful management attempts are audited.

`LOCAL`

Only successful management attempts by local users are audited.

`REMOTE`

Only successful management attempts by remote users are audited.

`NONE`

No successful management attempts are audited.

Omitting *`audit-spec`* specifies NONE.

`AUDIT-MANAGE-FAIL [`*`audit-spec`*`]`

establishes an *`audit-spec`* for unsuccessful attempts to manage a security group-authorization record. This *`audit-spec`* specifies the conditions under which an audit record is written to the audit file when an attempt to manage a security group-authorization record fails.

The form of *`audit-spec`* is:

```
{ ALL | LOCAL | REMOTE | NONE }
```

`ALL`

All unsuccessful management attempts are audited.

`LOCAL`

Only unsuccessful management attempts made by local users are audited.

`REMOTE`

Only unsuccessful management attempts made by remote users are audited.

`NONE`

No unsuccessful management attempts are audited.

Omitting *`audit-spec`* specifies NONE.

Example

These commands define default values for a new security group:

```
=SET SECURITY-GROUP OWNER prs.manager
=SET SECURITY-GROUP AUDIT-ACCESS-PASS all, &
=AUDIT-MANAGE-PASS local
=SET SECURITY-GROUP ACCESS 33,*  (e,o); (86,*, 255,*) *
=SET SECURITY-GROUP ACCESS prs.harry DENY *
```

The default group attribute values defined in this example are:

- The security group owner is the manager of the PRS group.
- The Safeguard software audits all successful attempts to execute a restricted command, as well as successful local attempts to manage a security group authorization record.
- All members of groups 33, 86, and 255 can execute restricted commands and manage security group authorization records (except for user PRS.HARRY, who is specifically denied all access).

The SET command is a SAFECOM environmental command that establishes default values for the attributes. These values are used whenever the ADD command does not explicitly state the value of the attribute.

To set all SECURITY-GROUP protection record attributes like those set in the SECURITY-OSS-ADMINISTRATOR security group:

```
=SET SECURITY-GROUP LIKE SECURITY-OSS-ADMINISTRATOR
```

To set all SECURITY-GROUP protection record attributes like those set in the SECURITY-PRV-ADMINISTRATOR security group:

```
=SET SECURITY-GROUP LIKE SECURITY-PRV-ADMINISTRATOR
```

To set all SECURITY-GROUP protection record attributes like those set in the SECURITY-AUDITOR security group:

```
=SET SECURITY-GROUP LIKE SECURITY-AUDITOR
```

To set all SECURITY-GROUP protection record attributes like those set in the SECURITY-MEDIA-ADMIN security group, use the following command:

```
=SET SECURITY-GROUP LIKE SECURITY-MEDIA-ADMIN
```

To set all SECURITY-GROUP protection record attributes like those set in the SECURITY-PERSISTENCE-ADMIN security group, use the following command:

```
=SET SECURITY-GROUP LIKE SECURITY-PERSISTENCE-ADMIN
```

SHOW SECURITY-GROUP Command

SHOW SECURITY-GROUP displays the current default values for the SECURITY-GROUP attributes.

```
SHOW [ / OUT listfile / ] SECURITY-GROUP
```

`OUT`

directs the SHOW SECURITY-GROUP report to *`listfile`*. After it executes the SHOW command, SAFECOM redirects its output to the current OUT file.

`listfile`

For *`listfile`*, specify any file name. SAFECOM opens the *`listfile`* and appends the SHOW SECURITY-GROUP report to that file. If *`listfile`* does not exist, SAFECOM creates it as an EDIT-format file.

SHOW SECURITY-GROUP Report Format

Figure 13-3 shows the format of the SHOW SECURITY-GROUP report.

Figure 13-3. SHOW SECURITY-GROUP Report Format

```
TYPE                 OWNER
SECURITY-GROUP       gn,un

OBJECT-TEXT-DESCRIPTION =

AUDIT-ACCESS-PASS = a-spec   AUDIT-MANAGE-PASS = a-spec
AUDIT-ACCESS-FAIL = a-spec   AUDIT-MANAGE-FAIL = a-spec

 user-spec [DENY] authority-list
 user-spec [DENY] authority-list
      .        .          .
      .        .          .

[ NO ACCESS CONTROL LIST DEFINED! ]
```

The SHOW SECURITY-GROUP report displays these attribute values:

`OWNER` *`gn,un`*

is the group number and member number of the user who will own this security group authorization record.

`AUDIT-ACCESS-PASS =` *`a-spec`* `AUDIT-MANAGE-PASS =` *`a-spec`*
`AUDIT-ACCESS-FAIL =` *`a-spec`* `AUDIT-MANAGE-FAIL =` *`a-spec`*

are the conditions under which the Safeguard software will audit attempts to execute restricted commands and attempts to manage this authorization record. For more information about these fields for *`audit-spec`*, see the SET SECURITY-GROUP Command on page 13-25.

`user-spec` `[DENY]` *`authority-list`*

is a current default ACL entry for the security group. For a full description, see INFO SECURITY-GROUP Brief Report on page 13-19.

`[ NO ACCESS CONTROL LIST DEFINED! ]`

specifies no default ACL entries are defined. Use SET SECURITY-GROUP. . . ACCESS to define default ACL entries. You can use ADD SECURITY-GROUP. . . ACCESS to define ACL entries when you create a security group-authorization record.

△ **Caution.** If you do not specify an ACL for a security group, only the local super ID can execute commands restricted to that security group.

Example

This SHOW SECURITY-GROUP report displays the current default security group attribute values. To display the report:

```
=SHOW SECURITY-GROUP
```

The report shows:

```
TYPE                          OWNER
 SECURITY-GROUP              255,18

  OBJECT-TEXT-DESCRIPTION =

  AUDIT-ACCESS-PASS = ALL        AUDIT-MANAGE-PASS = NONE
  AUDIT-ACCESS-FAIL = ALL        AUDIT-MANAGE-FAIL = NONE

         033,013      E,O
         033,255      E,O
         255,018      E,O
```

These current default values indicate that:

- The owner of a security group that has these attribute values is the local super-group member with user ID 255,18.
- The Safeguard software audits all successful and unsuccessful attempts to execute commands restricted by a future ADD SECURITY-GROUP command.

- The users with user IDs 33,13 and 255,18 as well as the group manager for group 33 have EXECUTE and OWNER authorities for the authorization record to be created for this security group.

THAW SECURITY-GROUP Command

THAW SECURITY-GROUP reenables the ACL for a frozen security group. The authorities granted the users on the ACL are reinstated.

The owner of a security group authorization record, the primary owner's group manager, and the local super ID can thaw a frozen group. Any user ID with O[WNER] authority can also thaw the authorization record.

THAW SECURITY-GROUP has no effect on a security group that is not frozen.

```
THAW SECURITY-GROUP sec-group-list
```

sec-group-list

specifies the security groups to be thawed. *sec-group-list* can be either:

```
  sec-group-spec

( sec-group-spec [ , sec-group-spec ] ... )
```

sec-group-spec

can be either:

```
SECURITY-ADMINISTRATOR
SYSTEM-OPERATOR
SECURITY-OSS-ADMINISTRATOR
SECURITY-PRV-ADMINISTRATOR
SECURITY-AUDITOR
SECURITY-MEDIA-ADMIN
SECURITY-PERSISTENCE-ADMIN
```

Example

To thaw the SYSTEM-OPERATOR ACL:

```
=THAW SECURITY-GROUP system-operator
```

The SECURITY-OSS-ADMINISTRATOR security group can be thawed by the primary owner or by any user with OWNER authority on the access control list for the group.

To thaw the group:

```
=THAW SECURITY-GROUP SECURITY-OSS-ADMINISTRATOR
```

To verify the results:

```
=INFO SECURITY-GROUP SECURITY-OSS-ADMINISTRATOR
```

The display shows:

```
                  LAST-MODIFIED        OWNER     STATUS
SECURITY-OSS-ADMINISTRATOR
                 14MAR06,  1:29        255,255   THAWED

                240,001            E
                240,002                   O
                240,003            E      O
                255,025            E      O
```

The SECURITY-PRV-ADMINISTRATOR security group can be thawn by the primary owner or by any user with OWNER authority on the access control list for the group.

To thaw the group:

```
=THAW SECURITY-GROUP SECURITY-PRV-ADMINISTRATOR
```

To verify the results:

```
=INFO SECURITY-GROUP SECURITY-PRV-ADMINISTRATOR
```

The display shows:

```
                  LAST-MODIFIED        OWNER     STATUS
SECURITY-PRV-ADMINISTRATOR
                 8DEC10,  12:26        SUPER.SUPER   THAWED

               NORMAL.ANORSPA            E
               SUPER.ASUPSPA             E,     O
```

The SECURITY-AUDITOR security group can be thawed by the primary owner or by any user with OWNER authority on the access control list for the group.

To thaw the group:

```
=THAW SECURITY-GROUP SECURITY-AUDITOR
```

To verify the results:

```
=INFO SECURITY-GROUP SECURITY-AUDITOR
```

The display shows:

```
                  LAST-MODIFIED        OWNER     STATUS
SECURITY-AUDITOR
                14MAR11, 1:29       255,255  THAWED

                 240,001               E
                 240,002                  O
                 240,003               E  O
                 255,025               E  O
```

The SECURITY-MEDIA-ADMIN security group can be thawed by the primary owner or by any user with OWNER authority on the access control list for the group.

To thaw the group:

```
=THAW SECURITY-GROUP SECURITY-MEDIA-ADMIN
```

To verify the results:

```
=INFO SECURITY-GROUP SECURITY-MEDIA-ADMIN
```

The display shows:

```
                  LAST-MODIFIED        OWNER     STATUS
SECURITY-MEDIA-ADMIN
                 14FEB13, 1:29      255,255  THAWED

                  240,001              E
                  240,002                 O
                  240,003              E  O
                  255,025              E  O
```

The SECURITY-PERSISTENCE-ADMIN security group can be thawed by the primary owner or by any user with OWNER authority on the access control list for the group.

To thaw the group:

```
=THAW SECURITY-GROUP SECURITY-PERSISTENCE-ADMIN
```

To verify the results:

```
=INFO SECURITY-GROUP SECURITY-PERSISTENCE-ADMIN
```

The display shows:

```
                  LAST-MODIFIED        OWNER     STATUS
SECURITY-PERSISTENCE-ADMIN
                 14FEB13, 1:29      255,255  THAWED

                  240,001              E
                  240,002                O
                  240,003              E O
                  255,025              E O
```

14 Terminal Security Commands

The terminal commands allow a security administrator to add and manage terminal definition records. When you add a terminal definition record, the Safeguard software takes control of the logon dialog at that terminal. When you define a terminal, you can also specify a particular command interpreter to be started automatically at the terminal after user authentication. Terminal definitions can be added selectively for some or all of the terminals on your system.

The automatic starting of specific command interpreter is available only at a Safeguard terminal. Even though this features can also be specified in a user authentication record and in a Safeguard configuration record, it is enforced only at terminals controlled by the Safeguard software.

Previously, an extended logon dialog was available only at Safeguard terminals. Effective with D30, the TACL command interpreter also provides these extended features as long as Safeguard is running.

You cannot specify access authorities with the terminal commands. These commands do not allow you to specify an access control list (ACL). To specify an ACL for a terminal, you must use DEVICE or SUBDEVICE commands, depending on how terminals are named on your system.

Except for the INFO TERMINAL command, use of the terminal commands is restricted to security group members. INFO TERMINAL can be executed by any user. If you have not defined the SECURITY-ADMINISTRATOR and SYSTEM-OPERATOR groups, any super-group member can use the terminal commands. (For information about how to define security groups, see Section 13, Security Group Commands.)

Terminals controlled by the Safeguard software can also be configured for exclusive access, which insures that any user who is logged on to a Safeguard terminal has exclusive access to the terminal until the user logs off. For more information, see TERMINAL-EXCLUSIVE-ACCESS { ON | OFF } on page 16-26.

Terminal Command Summary

Table 14-1 lists the terminal commands and gives a brief description of each.

Table 14-1. Terminal Command Summary (page 1 of 2)

Command	Description
ADD TERMINAL	Adds a terminal definition record with the specified terminal attribute values.
ALTER TERMINAL	Changes one or more attribute values in a terminal definition record.
DELETE TERMINAL	Deletes a terminal definition record.

Table 14-1. Terminal Command Summary (page 2 of 2)

Command	**Description**
FREEZE TERMINAL	Temporarily disables a terminal from accepting the LOGON command.
INFO TERMINAL	Displays the existing attribute values in a terminal definition record.
THAW TERMINAL	Reenables a frozen terminal so that it accepts the LOGON command.

Syntax of Terminal Commands

The remainder of this section describes each terminal command in detail. Terminal commands are presented in alphabetical order, and descriptions contain these elements:

- A summary of the command's function, including the restrictions on who can use the command
- The command syntax, including descriptions of the command parameters and variables
- The format for any command listing or report
- Considerations for the use of the command
- Examples of command usage

ADD TERMINAL Command

The ADD TERMINAL command adds a terminal definition record for a specified terminal. You can specify only one terminal name in an ADD TERMINAL command.

If you have defined a SECURITY-ADMINISTRATOR security group, only members of that group can use the ADD terminal command.

```
ADD TERMINAL  terminal-name [ , ]

   [ LIKE terminal-name | term-attribute ]

   [ , term-attribute ] ...
```

`TERMINAL`

specifies TERMINAL as the object type of the ADD command. Omit this option if TERMINAL is the assumed object type. (For more information on assumed object types, see the ASSUME Command on page 4-3.)

terminal-name

specifies the terminal to be controlled by the Safeguard software. *terminal-name* is a network name with the following form:

```
[\system.]$device[.#subdevice]
```

If you omit `\system`, your current default system name is used. If you omit *#subdevice*, no subdevice name is assumed.

LIKE *terminal-name*

adopts the existing terminal definition for *terminal-name* as the definition for the terminal being added in this command. LIKE defines values for these terminal attributes:

```
PROG [prog-filename]
LIB [lib-filename]
CPU [cpu-number]
SWAP [$vol[.subvol.filename]]
PRI [priority]
PARAM-TEXT [startup-param-text]
```

term-attribute

defines the command interpreter to be started after a user is authenticated at this terminal. *term-attribute* can be any of:

```
PROG [prog-filename]
LIB [lib-filename]
CPU [cpu-number | ANY]
PNAME [process-name]
SWAP [$vol[.subvol.filename]]
PRI [priority]
PARAM-TEXT [startup-param-text]
```

PROG [*prog-filename*]

specifies the command interpreter to be started after a user is authenticated at this terminal. *prog-filename* is the name of the command interpreter's object file. It must be a local file name.

If you omit *prog-filename*, the other *term-attribute*s in this record are not meaningful.

If *prog-filename* is omitted in the terminal definition record and no CI-PROG is specified in the authentication record of the user who is logging on, the Safeguard software starts the CI-PROG (with associated parameters) defined in the Safeguard configuration record.

LIB [*lib-filename*]

specifies the library file to be used with the command interpreter started at this terminal after user authentication. *lib-filename* must be a local file name.

If you omit *lib-filename*, no library file is used.

CPU [*cpu-number* | ANY]

specifies the number of the CPU in which the command interpreter is to run. If you specify *ANY*, any CPU will be used.

If you omit *cpu-number*, any CPU will be used.

PNAME [*process-name*]

specifies the process name to be assigned to the command interpreter started at this terminal after user authentication. *process-name* must be a local process name.

If you omit *process-name* from the terminal definition record, the Safeguard software generates a process name.

SWAP [$*vol*[*.subvol.filename*]]

specifies the name of the volume or file to be used as the swap volume or file for the command interpreter started at this terminal. $*vol* must be a local volume name. You can optionally supply a subvolume name and file name.

If you omit $*vol*, the same volume that contains the PROG object file is used.

PRI [*priority*]

specifies the priority at which the command interpreter is to run at this terminal.

If you omit *priority* from the terminal definition record, the value of CI-PRI in the Safeguard configuration record is used.

PARAM-TEXT [*startup-param-text*]

specifies the data to be supplied as the startup message text for the command interpreter started at this terminal. If you specify the PARAM-TEXT attribute, it must be the last attribute in the command string.

If you omit *startup-param-text*, no startup parameter text is used.

Considerations

- You must stop any process running at the terminal before you add a terminal definition record for that terminal. If you do not stop the other process, the Safeguard software competes with it for control of the terminal.
- After you add a terminal definition record, that terminal is disabled (frozen) until you execute a THAW TERMINAL command to enable the terminal.
- The PROG specified in this record is overridden by any CI-PROG (and associated parameters) specified in the authentication record of the user who is logging on.

- When you add a terminal on a remote system (*system.device*), you must ensure that the terminal is completely accessible to the super ID. For example, the appropriate remote passwords must be established, and the terminal must not have an ACL that denies access to the super ID.
- If you specify a PNAME, be sure it is unique for each terminal. For this reason, LIKE does not include the PNAME attribute.
- With a normal Safeguard configuration, you should be able to add up to about 450 terminal definition records. If you attempt to add more than the maximum number of terminals, SAFECOM issues an error message and rejects the ADD TERMINAL command.

Examples

1. These commands add a terminal definition for the terminal $TFOX.#T014 and enable the terminal:

   ```
   =ADD TERMINAL $tfox.#t014
   =THAW TERMINAL $tfox.#t014
   ```

   This command uses the values from the Safeguard configuration record for the command interpreter to be started at the terminal after a user is authenticated.

2. The following command adds a terminal definition for the terminal $TFOX.#T009. It specifies that a command interpreter is to be started from the object file WORDS.FORMAT after any user is authenticated at the terminal:

   ```
   =ADD TERMINAL $tfox.#t009, PROG words.format
   ```

3. The following command adds a terminal definition for the terminal $TFOX.#T010. This terminal is to have the same definition as terminal $TFOX.#T009, except that the command interpreter will use a library file named BOOKS.LIB:

   ```
   =ADD TERM $tfox.#t010, LIKE $tfox.#t009, LIB books.lib
   ```

4. This example assumes that a TACL process is currently running at the terminal named $TC02.#C12. The example shows the use of TACL commands to check the terminal's status and stop the process running at the terminal before adding the terminal to the Safeguard database:

   ```
   1>STATUS *, TERM $tc02.#c12
   ```

```
Process          Pri PFR %WT Userid  Program file          Hometerm
$PTC02 B  6,44  150     001   4,122 $SYSTEM.SYS06.TACL    $TC02.#C12
$PTC02    7,49  150   R 001   4,122 $SYSTEM.SYS06.TACL    $TC02.#C12
```

```
2>STOP $PTC02
3>SAFECOM
=ADD TERM $tc02.#c12
```

ALTER TERMINAL Command

The ALTER TERMINAL command changes one or more terminal attribute values in a terminal definition record. You can specify only one terminal name in an ALTER TERMINAL command, but that name can contain wild-card characters.

If you have defined a SECURITY-ADMINISTRATOR security group, only members of that group can use the ALTER terminal command.

```
ALTER TERMINAL  terminal-spec [ , ]

   { LIKE terminal-name | term-attribute }

   [ , term-attribute ] ...
```

TERMINAL

specifies TERMINAL as the object type of the ALTER command. Omit this option if TERMINAL is the assumed object type. (For more information on assumed object types, see the ASSUME Command on page 4-3.)

terminal-spec

specifies the terminal or terminals whose definition is to be changed. *terminal-spec* has the following form:

```
[\system-name.]$device[.#subdevice]
```

LIKE *terminal-name*

changes the terminal attributes for this terminal to match the existing specifications for *terminal-name*.

LIKE sets the values for only the following terminal attributes:

```
PROG prog-filename
LIB lib-filename
CPU cpu-number
SWAP $vol[.subvol.filename]
PRI priority
PARAM-TEXT startup-param-text
```

term-attribute

changes the value of a terminal attribute. *term-attribute* can be any of:

```
PROG [prog-filename]
LIB [lib-filename]
CPU [cpu-number | ANY]
PNAME [process-name]
SWAP [$vol[.subvol.filename]]
PRI [priority]
PARAM-TEXT [startup-param-text]
```

For a complete description of each terminal attribute, see the ADD TERMINAL Command on page 14-2.

Considerations

- If you specify a PNAME attribute, be sure is unique for each terminal. For this reason, LIKE does not include the PNAME attribute.

Examples

The following command alters the terminal definition for the terminal $TFOX.#T009. It specifies that TACL is to be the command interpreter, that it is to run in CPU 3, and that the startup parameter text is 4. TACL interprets this text as the backup CPU number.

```
=ALTER TERMINAL $tfox.#t009, PROG $system.system.tacl, &
=CPU 3, PARAM-TEXT 4
```

DELETE TERMINAL Command

The DELETE TERMINAL command deletes a terminal definition record for a specified terminal. You can specify only one terminal name in a DELETE TERMINAL command, but that name can contain wild-card characters.

If you have defined a SECURITY-ADMINISTRATOR security group, only members of that group can use the DELETE TERMINAL command.

```
DELETE TERMINAL  terminal-spec
```

`TERMINAL`

specifies TERMINAL as the object type of the DELETE command. Omit this option if TERMINAL is the assumed object type. (For more information on assumed object types, see the ASSUME Command on page 4-3.)

`terminal-spec`

specifies the terminal or terminals whose definition is to be deleted.

Considerations

- The terminal must be frozen with a FREEZE TERMINAL command before you can delete it with a DELETE TERMINAL command.
- After you delete a terminal definition record, that terminal is not usable until you start another command interpreter to handle the logon dialog at the terminal.

Examples

To delete the terminal definition record for terminal $TCO2.#A14:

```
=DELETE TERMINAL $tc02.#a14
```

FREEZE TERMINAL Command

The FREEZE TERMINAL command freezes a terminal definition record so that the logon dialog at that terminal becomes disabled. Only one terminal name can be specified in a FREEZE TERMINAL command, but that name can contain wild-card characters.

If you have defined SECURITY-ADMINISTRATOR and SYSTEM-OPERATOR security groups, use of FREEZE TERMINAL is restricted to the members of those security groups.

```
FREEZE TERMINAL  terminal-spec
```

`TERMINAL`

specifies TERMINAL as the object type of the FREEZE command. Omit this option if TERMINAL is the assumed object type. (For more information on assumed object types, see the ASSUME Command on page 4-3.)

`terminal-spec`

specifies the terminal or terminals whose definition record is to be frozen.

Consideration

- If no terminal definition record exists for the terminal, FREEZE TERMINAL has no effect.

Examples

1. To freeze the terminal definition record for $TF11.#C09 so that the terminal does not accept logon attempts:

   ```
   =FREEZE TERMINAL $tf11.#c09
   ```

2. To freeze all terminals whose names begin with $TF11.#C:

   ```
   =FREEZE TERMINAL $tf11.#c*
   ```

INFO TERMINAL Command

The INFO TERMINAL command shows the terminal attributes stored in a specified terminal definition record. Only one terminal name can be specified in an INFO TERMINAL command, but that name can contain wild-card characters.

Any user can execute the INFO TERMINAL command.

```
INFO [ / OUT listfile / ] TERMINAL [ , ] terminal-spec
```

`TERMINAL`

specifies TERMINAL as the object type of the INFO command. Omit this option if TERMINAL is the assumed object type. (For more information on assumed object types, see the ASSUME Command on page 4-3.)

`OUT`

directs the INFO TERMINAL report to *`listfile`*. After it executes the INFO command, SAFECOM redirects its output to the current OUT file.

`listfile`

For *`listfile`*, specify any file name. SAFECOM opens the *`listfile`* and appends the INFO report to the file. If *`listfile`* does not exist, SAFECOM creates it as an EDIT-format file.

`terminal-spec`

specifies the terminal or terminals whose definition is to be displayed.

INFO TERMINAL Report Format

Figure 14-1 shows the format of the INFO TERMINAL report.

Figure 14-1. INFO TERMINAL Report Format

```
TERMINAL   terminal-name                          STATUS   status

  PROG   = prog-filename
  LIB    = lib-filename
  PNAME  = process-name
  SWAP   = $vol[.subvol.filename]
  CPU    = cpu-number
  PRI    = priority

  PARAM-TEXT = startup-param-text
```

The report contains these terminal attribute values and status fields:

`TERMINAL` *`terminal-name`*

is the name of the terminal whose attributes are being displayed.

`STATUS` *`status`*

is the status of the terminal, either FROZEN or THAWED.

`PROG =` *`prog-filename`*

is the name of the object file of the command interpreter started at this terminal.

`LIB` *`lib-filename`*

is the name of the library file used with the command interpreter.

`CPU {` *`cpu-number`* `| ANY }`

is the number of the CPU in which the command interpreter runs.

`PNAME` *`process-name`*

is the process name assigned to the command interpreter that runs at this terminal.

`SWAP $`*`vol`*`[`*`.subvol.filename`*`]`

is the name of the volume or file used as the swap volume or swap file for the command interpreter.

`PRI` *`priority`*

is the priority at which the command interpreter runs.

`PARAM-TEXT` *`startup-param-text`*

is the data supplied as the startup message text for the command interpreter.

Examples

To display the terminal attributes for the terminal name $TF11.#C09:

```
=INFO TERMINAL $TF11.#C09
```

```
TERMINAL  $TF11.#C09                     STATUS  FROZEN

 PROG  = $SYSTEM.SYSTEM.TACL
 LIB   = * NONE *
 PNAME = * NONE *
 SWAP  = * NONE *
 CPU   = ANY
 PRI   = 150

 PARAM-TEXT =
```

THAW TERMINAL Command

The THAW TERMINAL command reenables a terminal whose terminal definition record is frozen. Only one terminal name can be specified in a THAW TERMINAL command, but that name can contain wild-card characters.

If you have defined SECURITY-ADMINISTRATOR and SYSTEM-OPERATOR security groups, use of THAW TERMINAL is restricted to the members of those security groups.

```
THAW TERMINAL  terminal-spec
```

`TERMINAL`

specifies TERMINAL as the object type of the THAW command. Omit this option if TERMINAL is the assumed object type. (For more information on assumed object types, see the ASSUME Command on page 4-3.)

`terminal-spec`

specifies the terminal whose definition is to be thawed.

Considerations

- If no terminal definition record exists for the terminal, THAW TERMINAL has no effect.
- If the terminal definition record is not frozen, THAW TERMINAL has no effect.

Examples

To thaw the terminal $TF11.#C03 so that it accepts logon attempts:

```
=THAW TERMINAL $tf11.#c03
```

15 Event-Exit-Process Commands

The event-exit-process commands allow a security administrator to configure and manage the security event exit process.

A security event-exit process is a user-written process that is allowed to participate in security policy enforcement. Depending on how the event-exit process is configured, the Safeguard subsystem passes it requests for authorization, authentication, and password changes. The event-exit process rules on the request and returns the ruling to the Safeguard subsystem for interpretation and enforcement.

This section describes the commands used to configure the event-exit process, provides design considerations for designing and writing an event-exit process, and documents the interprocess messages exchanged between the Safeguard subsystem and the event-exit process.

If you have defined a SECURITY-ADMINISTRATOR security group, only members of that security group can use the ADD, ALTER, and DELETE EVENT-EXIT-PROCESS commands. If you have not defined the SECURITY-ADMINISTRATOR group, any super-group member can use these commands.

Any user can execute the INFO EVENT-EXIT-PROCESS command.

Event-Exit-Process Command Summary

Table 15-1 lists the event-exit-process commands and gives a brief description of each.

Table 15-1. Event-Exit-Process Command Summary

Command	Description
ADD EVENT-EXIT-PROCESS	Adds an event-exit-process configuration record.
ALTER EVENT-EXIT-PROCESS	Changes one or more attribute values of the event-exit process-configuration.
DELETE EVENT-EXIT-PROCESS	Deletes an event-exit-process configuration record.
INFO EVENT-EXIT-PROCESS	Displays the existing attribute values defined for the event-exit-process configuration.

Syntax of Event-Exit-Process Commands

The remainder of this section describes each event-exit-process command in detail. Commands are presented in alphabetical order, and descriptions contain these elements:

- A summary of the command's function, including the restrictions on who can use the command

- The command syntax, including descriptions of the command parameters and variables
- The format for any command listing or report
- Considerations for the use of the command
- Examples of command usage

In addition, this section contains these information about the event-exit process:

- The format of interprocess messages exchanged between the Safeguard subsystem and the event-exit process
- Programming considerations for writing an event-exit process

ADD EVENT-EXIT-PROCESS Command

The ADD EVENT-EXIT-PROCESS command adds an event-exit configuration record and optionally specifies values for one or more of the event-exit-configuration attributes. Attributes not specified in the ADD command assume the default values given in the following attribute descriptions.

If you have defined a SECURITY-ADMINISTRATOR security group, only members of that group can use the ADD EVENT-EXIT-PROCESS command. If you have not defined the SECURITY-ADMINISTRATOR security group, any super-group member can use this command.

```
ADD EVENT-EXIT-PROCESS name  [ [ , ] exit-attribute ]

   [ , exit-attribute ] ...
```

name

specifies the name of the event-exit configuration record. The name can be up to 32 alphanumeric characters long. It can also include hyphen (-) and underscore (_) characters. The first character must be alphabetic or an underscore.

exit-attribute

specifies the name of the event-exit-process attribute to be set. The *exit-attribute*s are:

```
ENABLED { ON | OFF }
RESPONSE-TIMEOUT [ n [ SECONDS ] ]
TIMEOUT-ALL-AUTHZREQ { ON | OFF }
ENABLE-AUTHENTICATION-EVENT { ON | OFF }
ENABLE-AUTHORIZATION-EVENT { ON | OFF }
ENABLE-PASSWORD-EVENT { ON | OFF }
PROG [ prog-filename ]
LIB [ lib-filename ]
SWAP [ $vol  [ subvol.filename ] ]
PNAME [ process-name ]
```

```
CPU [ cpu-number | ANY ]
PRI [ priority ]
PARAM-TEXT [ startup-param-text ]
```

`ENABLED { ON | OFF }`

defines whether the security event exit is enabled. ON indicates that the event exit is enabled and that the Safeguard software is to start the event-exit process and send designated security event messages to the process. If another process with the same name is running when the event-exit process is enabled, Safeguard kills that process before starting the event-exit process. OFF indicates that the Safeguard software is not to start the event-exit process.

The PROG object file must be specified before (or in the same command) ENABLED is set to ON.

The default value is OFF. If you omit this attribute, it is set to the default value.

`RESPONSE-TIMEOUT [ n [ SECONDS ] ]`

specifies the number of seconds, from 1 through 900, that the Safeguard software is to wait for the event-exit process to respond to an event.

If a timeout occurs and the request is for a password-quality or authorization event from an undeniable user, the event proceeds with the check performed by the Safeguard software. Locally authenticated super-group members are undeniable users. All other users are considered deniable users.

If a timeout occurs and the request is for a password-quality event from a deniable user, the request is denied.

If a timeout occurs when the attribute TIMEOUT-ALL-AUTHZREQ is set to ON, and the request is for an authorization event from a deniable user, then the request is denied. If the attribute TIMEOUT-ALL-AUTHZREQ is set to OFF, then the deniable user waits indefinitely with neither approval nor denial.

If a timeout occurs and the request is for authentication by any user, the request is denied.

The default value is five seconds. If you omit this attribute, it is set to the default value.

`TIMEOUT-ALL-AUTHZREQ { ON | OFF }`

specifies whether authorization request from a deniable user will be timed out. A time out occurs if the response is not recieved from SEEP within the window indicated by the RESPONSE-TIMEOUT attribute. ON indicates that authorization requests from all users are timed out. OFF indicates that only requests from undeniable users are subject to timeout and the deniable users wait indefinitely with neither approval nor denial to proceed with the request.

Locally authenticated super-group members are treated as undeniable and all other users are considered as deniable.

Note. The TIMEOUT-ALL-AUTHZREQ attribute is supported only on systems running H06.26 and later H-series RVUs and J06.15 and later J-series RVUs.

`ENABLE-AUTHENTICATION-EVENT { ON | OFF }`

specifies whether authentication events are to be sent to the event-exit process. ON indicates that the events are sent to the event-exit process when it is enabled. For a complete list of events sent when ENABLE-AUTHENTICATION-EVENT is ON, see Design Considerations on page 15-24. For more information, see ENABLE-PASSWORD-EVENT { ON | OFF } on page 15-4.

The default value is OFF. If you omit this attribute, it is set to the default value.

`ENABLE-AUTHORIZATION-EVENT { ON | OFF }`

specifies whether authorization events are to be sent to the event-exit process. ON indicates that the events will be sent to the event-exit process when it is enabled. For a complete list of events that are sent when ENABLE-AUTHORIZATION-EVENT is ON, see Design Considerations on page 15-24.

The default value is OFF. If you omit this attribute, it is set to the default value.

`ENABLE-PASSWORD-EVENT { ON | OFF }`

specifies whether password change events are to be sent to the event-exit process for a password-quality check. ON indicates that the events are sent to the event-exit process when it is enabled.

If ENABLE-PASSWORD-EVENT is ON and ENABLE-AUTHENTICATION-EVENT is also ON, password changes that occur during a logon dialog are not sent to the password-quality exit. In this instance, the password-quality exit is invoked only for password changes from the PASSWORD program and from the Safeguard ADD USER, ALTER USER, ADD ALIAS, and ALTER ALIAS commands.

If ENABLE-PASSWORD-EVENT is ON and ENABLE-AUTHENTICATION-EVENT is OFF, all password change events are sent to the password-quality exit for evaluation. For more information, see Design Considerations on page 15-24.

The default value is OFF. If you omit this attribute, it is set to the default value.

`PROG [`*`prog-filename`*`]`

specifies the name of the object program file to be run when the ENABLED attribute is set to ON. It must be a local file name. *`prog-filename`* must be specified before the ENABLED attribute can be set to ON.

The default value is no object program file. If you omit this attribute, it is set to the default value.

If the ENABLED attribute is set to ON and an attempt is made to set this attribute to null, the command is rejected. The ENABLED attribute must be set to OFF before this field can be set to null.

`LIB [`*`lib-filename`*`]`

specifies the library file to be used with the event-exit process. *`lib-filename`* must be a local file name.

The default value is no library. If you omit this attribute, it is set to the default value.

`SWAP [$`*`vol`*`[`*`.subvol.filename`*`]]`

specifies the name of the volume or file to be used as the swap volume or file for the event-exit process. `$`*`vol`* must be a local volume name. You can optionally supply a local subvolume name and file name.

If you omit this attribute, the value used for SWAP when starting PROG is the same volume that contains the PROG object file.

`PNAME [`*`process-name`*`]`

specifies the process name to be assigned to the event-exit process when it is started. *`process-name`* must be a local process name. Avoid using an existing process name because Safeguard will kill that process before it starts the event-exit process.

The default value is no process name, which indicates that the Safeguard software is to generate a process name. If you omit this attribute, it is set to the default value.

`CPU [`*`cpu-number`*` | ANY]`

specifies the number of the CPU in which the event-exit process is to run. If you specify ANY, any CPU is used.

The default value is ANY CPU. If you omit this attribute, it is set to the default value.

`PRI [`*`priority`*`]`

specifies the priority at which the event-exit process is to run.

The default value is 155. If you omit this attribute, it is set to the default value.

PARAM-TEXT [*startup-param-text*]

specifies up to 255 characters of data to be supplied as the startup message text for the event-exit process. If you specify the PARAM-TEXT attribute, it must be the last attribute in the command string.

The default value is no text. If you omit this attribute, it is set to the default value.

Note. Startup message text is commonly used to specify a backup CPU. You can use startup message text to specify a backup CPU for the event-exit process. To do so, specify the backup CPU number as the first character of *startup-param-text*. You can add any subsequent parameters after a semicolon.

Considerations

- The event-exit process must be multithreaded and must perform NOWAITED I/O.
- Do not specify $SYSTEM.SYSTEM.NULL as *prog-filename*. The process must open its $RECEIVE queue in order to complete the enable.

Examples

1. The following command adds a configuration record for the event-exit process LOGON1, enables the event-exit process for password and logon events, specifies that the program object file named $DEV.SECURE.EVENTS should be started, and specifies that the event-exit process should run in CPU 3, with CPU 4 as the backup CPU. (The startup parameter text 4 specifies the backup CPU.)

```
=ADD EVENT-EXIT-PROCESS logon1 ENABLED on, &
=ENABLE-AUTHENTICATION-EVENT on, &
=ENABLE-PASSWORD-EVENT on, &
=PROG $dev.secure.events, CPU 3, PARAM-TEXT 4
```

ALTER EVENT-EXIT-PROCESS Command

The ALTER EVENT-EXIT-PROCESS command alters one or more of the event-exit configuration attributes.

If you have defined a SECURITY-ADMINISTRATOR security group, only members of that group can use the ALTER EVENT-EXIT-PROCESS command. If you have not defined the SECURITY-ADMINISTRATOR security group, any super-group member can use this command.

```
ALTER EVENT-EXIT-PROCESS  name [ , ] exit-attribute

   [ , exit-attribute ] ...
```

name

specifies the name of the event exit whose *exit-attribute*s are to be changed.

exit-attribute

specifies the name of the event-exit attribute to be changed. The *exit-attributes* are:

```
ENABLED { ON | OFF }
RESPONSE-TIMEOUT [ n [ SECONDS ] ]
TIMEOUT-ALL-AUTHZREQ { ON | OFF }
ENABLE-AUTHENTICATION-EVENT { ON | OFF }
ENABLE-AUTHORIZATION-EVENT { ON | OFF }
ENABLE-PASSWORD-EVENT { ON | OFF }
PROG [ prog-filename ]
LIB [ lib-filename ]
SWAP [ $vol  [ subvol.filename ] ]
PNAME [ process-name ]
CPU [ cpu-number | ANY ]
PRI [ priority ]
PARAM-TEXT [ startup-param-text ]
```

ENABLED { ON | OFF }

defines whether the security event exit is enabled. ON indicates that the event exit is enabled and that the Safeguard software is to start the event-exit process and send designated security event messages to the process. If another process with the same name is running when the event-exit process is enabled, Safeguard kills that process before starting the event-exit process. OFF indicates that the Safeguard software is not to start the exit process. If the event-exit process is running when this attribute is set to OFF, the Safeguard software stops the process.

The PROG object file must be specified before (or in the same command) ENABLED is set to ON.

The default value is OFF.

RESPONSE-TIMEOUT [*n* [SECONDS]]

specifies the number of seconds, from 1 through 900, that the Safeguard software is to wait for the event-exit process to respond to an event.

If a timeout occurs and the request is for a password-quality or authorization event from an undeniable user, the event proceeds with the check performed by the Safeguard software. Locally authenticated super-group members are undeniable users. All other users are considered deniable users.

If a timeout occurs and the request is for a password-quality event from a deniable user, the request is denied.

If a timeout occurs when the attribute TIMEOUT-ALL-AUTHZREQ is set to ON, and the request is for an authorization event from a deniable user, then the request is denied. If the attribute TIMEOUT-ALL-AUTHZREQ is set to OFF, then the deniable user waits indefinitely with neither approval nor denial.

If a timeout occurs and the request is for a authorization event from a deniable user, the user waits indefinitely with no approval or denial.

If a timeout occurs and the request is for authentication by any user, the request is denied.

The default value is five seconds. A null entry resets the value to the default value.

`TIMEOUT-ALL-AUTHZREQ { ON | OFF }`

specifies whether authorization request from a deniable user will be timed out. A time out occurs if the response is not recieved from SEEP within the window indicated by the RESPONSE-TIMEOUT attribute. ON indicates that authorization requests from all users are timed out. OFF indicates that only requests from undeniable users are subject to timeout and the deniable users wait indefinitely with neither approval nor denial to proceed with the request. Locally authenticated super-group members are treated as undeniable and all other users are considered as deniable.

Note. The TIMEOUT-ALL-AUTHZREQ attribute is supported only on systems running H06.26 and later H-series RVUs and J06.15 and later J-series RVUs.

`ENABLE-AUTHENTICATION-EVENT { ON | OFF }`

specifies whether authentication events are sent to the event-exit process.

The default value is OFF.

`ENABLE-AUTHORIZATION-EVENT { ON | OFF }`

specifies whether authorization events are sent to the event-exit process.

The default value is OFF.

`ENABLE-PASSWORD-EVENT { ON | OFF }`

specifies whether password change events are sent to the event-exit process for a password-quality check. ON indicates that the events are sent to the event-exit process when it is enabled.

If ENABLE-PASSWORD-EVENT is ON and ENABLE-AUTHENTICATION-EVENT is also ON, password changes that occur during a logon dialog are not sent to the password-quality exit. In this instance, the password-quality exit is invoked only for password changes from the PASSWORD program and from the Safeguard ADD USER, ALTER USER, ADD ALIAS, and ALTER ALIAS commands.

If ENABLE-PASSWORD-EVENT is ON and ENABLE-AUTHENTICATION-EVENT is OFF, all password change events are sent to the password-quality exit for evaluation.

`PROG [`*`prog-filename`*`]`

specifies the name of the object program file to be run when the ENABLED attribute is set to ON. It must be a local file name. *`prog-filename`* must be specified before the ENABLED attribute can be set to ON.

The default value is no object program file. A null entry resets the value to the default value.

If an attempt is made to set this field to null and the ENABLED attribute is set to ON, the command is rejected. The ENABLED attribute must be set to OFF before this field can be set to null.

`LIB [`*`lib-filename`*`]`

specifies the library file to be used with the event-exit process. *`lib-filename`* must be a local file name.

The default value is no library. A null entry resets the value to the default value.

`SWAP [$`*`vol`*`[`*`.subvol.filename`*`]]`

specifies the name of the volume or file to be used as the swap volume or file for the event-exit process. `$`*`vol`* must be a local volume name. You can optionally supply a subvolume name and file name.

If you omit this attribute, the value used for SWAP when starting PROG is the same volume that contains the PROG object file.

`PNAME [`*`process-name`*`]`

specifies the process name to be assigned to the event-exit process when it is started. *`process-name`* must be a local process name.

The default value is no process name, which indicates that the Safeguard software is to generate a process name. A null entry resets the value to the default value.

`CPU [`*`cpu-number`*` | ANY]`

specifies the number of the CPU in which the event-exit process is to run. If you specify ANY, any CPU is used.

The default value is ANY CPU. A null entry resets the value to the default value.

`PRI [`*`priority`*`]`

specifies the priority at which the event-exit process is to run.

The default value is 155. A null entry resets the value to the default value.

`PARAM-TEXT [`*`startup-param-text`*`]`

specifies up to 255 characters of data to be supplied as the startup message text for the event-exit process. If you specify the PARAM-TEXT attribute, it must be the last attribute in the command string.

The default value is no text. A null entry resets the value to the default value.

Considerations

- The event-exit process must be multithreaded and must perform NOWAITED I/O.
- Do not specify $SYSTEM.SYSTEM.NULL as *`prog-filename`*. The process must open its $RECEIVE queue in order to complete the enable.
- Do not specify an existing process name as *`process-name`* because $ZSMP will kill that process before starting the event-exit process.

Examples

1. To change the response timeout to 15 seconds for the event-exit process LOGON1:

```
=ALTER EVENT-EXIT-PROCESS logon1, RESPONSE-TIMEOUT 15
```

2. To disable the event-exit process:

```
=ALTER EVENT-EXIT-PROCESS logon1 ENABLED off
```

DELETE EVENT-EXIT-PROCESS Command

DELETE EVENT-EXIT-PROCESS removes an event-exit configuration record from the Safeguard database. After the configuration record is deleted, the event-exit process cannot be used unless a new configuration record is created for it.

The event exit must be disabled before its configuration record can be deleted.

If you have defined a SECURITY-ADMINISTRATOR security group, only members of that group can use the DELETE EVENT-EXIT-PROCESS command. If you have not defined the SECURITY-ADMINISTRATOR security group, any super-group member can use this command.

```
DELETE EVENT-EXIT-PROCESS  name
```

`name`

specifies the name of the event-exit process-configuration record to be deleted.

Examples

1. To delete the configuration record for the event-exit process LOGON1:

```
=DELETE EVENT-EXIT-PROC logon1
```

INFO EVENT-EXIT-PROCESS Command

The INFO EVENT-EXIT-PROCESS command shows the event-exit attributes stored in the specified event-exit configuration record. Only one event-exit name can be specified in an INFO EVENT-EXIT-PROCESS command.

Any user can execute the INFO EVENT-EXIT-PROCESS command.

```
INFO EVENT-EXIT-PROCESS  name
```

name

specifies the name of the event-exit process-configuration record for which an INFO report is to be produced.

INFO EVENT-EXIT-PROCESS Report Format

Figure 15-1 on page 15-11shows the format of the INFO EVENT-EXIT-PROCESS report.

Figure 15-1. Detailed INFO EVENT-EXIT-PROCESS Report

```
EVENT-EXIT-PROCESS = name
ENABLED = { ON | OFF }
RESPONSE-TIMEOUT = n SECONDS
TIMEOUT-ALL-AUTHZREQ = { ON | OFF }
ENABLE-AUTHENTICATION-EVENT = { ON | OFF }
ENABLE-AUTHORIZATION-EVENT = { ON | OFF }
ENABLE-PASSWORD-EVENT = { ON | OFF }
PROG  = [prog-filename]
LIB   = [lib-filename]
SWAP  = [$vol[.subvol.filename] ]
PNAME = [process-name]
CPU   = {cpu-number \ ANY}
PRI   = [priority]
PARAM-TEXT = [startup-param-text]
```

The report contains these fields:

`EVENT-EXIT-PROCESS =` *name*

is the name of the event-exit-process configuration record.

`ENABLED = { ON | OFF }`

indicates whether the event-exit process is enabled.

`RESPONSE-TIMEOUT = `*`n`*` SECONDS`

is the maximum number of seconds that the Safeguard software waits for the event-exit process to respond to an event.

`TIMEOUT-ALL-AUTHZREQ = { ON | OFF }`

indicates whether Safeguard will time out while waiting for response from SEEP on authorization events requested by deniable users.

Note. The TIMEOUT-ALL-AUTHZREQ attribute is supported only on systems running H06.26 and later H-series RVUs and J06.15 and later J-series RVUs.

`ENABLE-AUTHENTICATION-EVENT = { ON | OFF }`

indicates whether authentication events are sent to the event-exit process.

`ENABLE-AUTHORIZATION-EVENT = { ON | OFF }`

indicates whether authorization events are sent to the event-exit process.

`ENABLE-PASSWORD-EVENT = { ON | OFF }`

indicates whether password change events are sent to the event-exit process.

`PROG = [ `*`prog-filename`*` ]`

is the name of the object file of the event-exit process started when ENABLED=ON.

`LIB = [ `*`lib-filename`*` ]`

is the name of the library file used with this event-exit process.

`SWAP [ $`*`vol`*`[`*`.subvol.filename`*`] ]`

is the name of the volume or file used as the swap volume or swap file for the event-exit process.

`NAME [ `*`process-name`*` ]`

is the process name assigned to the event-exit process when it is started.

`CPU { `*`cpu-number`*` | ANY }`

is the number of the CPU in which the event-exit process runs.

`PRI [ `*`priority`*` ]`

is the priority at which the event-exit process runs.

`PARAM-TEXT [ `*`startup-param-text`*` ]`

is the data supplied as the startup message text for the event-exit process.

Examples

To display the event-exit attributes for an event-exit process that is enabled:

```
=INFO EVENT-EXIT-PROCESS logon1
```

```
EVENT-EXIT-PROCESS = LOGON1
ENABLED = ON
RESPONSE-TIMEOUT= 15 SECONDS
TIMEOUT-ALL-AUTHZREQ = ON
ENABLE-AUTHENTICATION-EVENT = ON
ENABLE-AUTHORIZATION-EVENT = ON
ENABLE-PASSWORD-EVENT = OFF
PROG  = $DEV.SECURE.EVENTS
LIB   = * NONE *
NAME = $EVENTS2
SWAP  = * NONE *
CPU   = 3
PRI   = 155
PARAM-TEXT = 4
```

Interprocess Communication Messages

The Safeguard software communicates with the event-exit process by sending messages using the standard file system procedure WRITEREAD[X]. The message sent to the event-exit process is divided into three primary sections: Header_Data, Subject_Data, and Message_Data.

Header_Data and Subject_Data have the same format for all message types although the content of some fields can vary based on data availability. (For more information, see Table 15-2 on page 15-15 and Table 15-3 on page 15-17.) The event-exit process provides its response and status information by altering fields in the Header_Data and returning it to the Safeguard subsystem. The event-exit process does not return the Subject_Data. The event-exit process might return the Message_Response_Data, depending on the type of event.

Message_Data sent by the Safeguard software has a different structure for each type of event, as shown in Table 15-4 on page 15-18 through Table 15-6 on page 15-22. The event-exit process replies to logon and password change events by overlaying the Message_Data with message response data as shown in Table 15-7 on page 15-22 and Table 15-8 on page 15-23. It does not provide message response data for access control or logon abort events.

The names of the data items in Table 15-2 on page 15-15 through Table 15-8 on page 15-23 are pseudonames, and do not exactly match the names as specified in the DDL. For variable names, see the DDL output. The DDL is located in the file ZSAFEGRD.SEEPDDL.

Figure 15-2 illustrates the structure of the message buffer between the Safeguard software and the event-exit process.

Figure 15-2. Event-Exit-Process Message Buffer

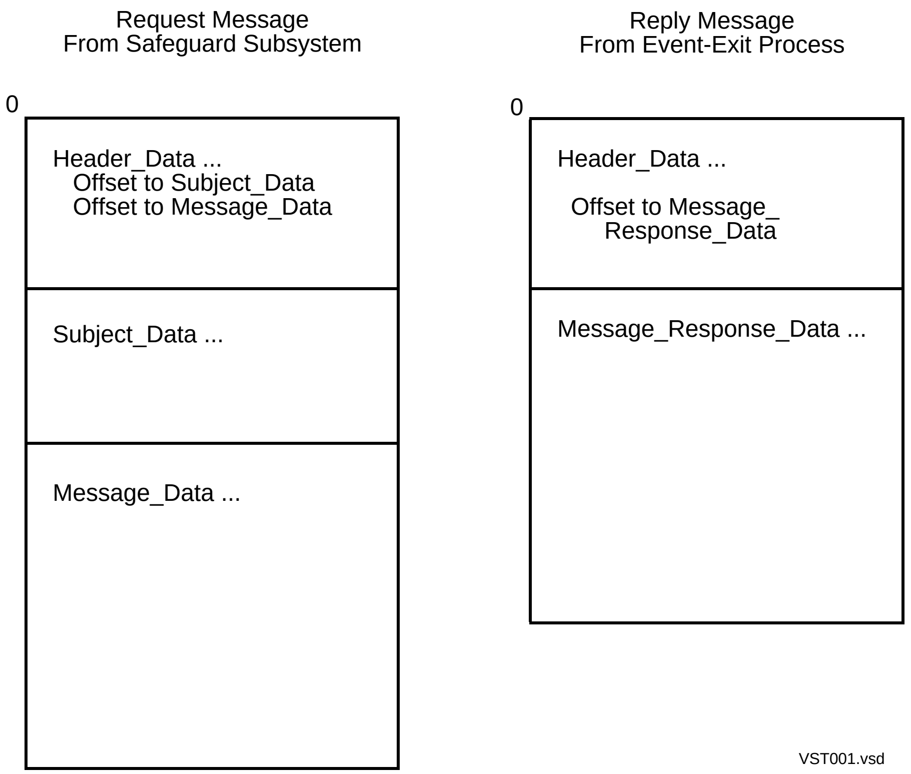

Table 15-2 shows the structure of the header data sent from the Safeguard subsystem to the event-exit process. The header is always present. When the event-exit process responds to an event request, it is expected to alter these fields in the header data:

- Message_Tag
- Error
- Status
- Subject_Data
- Message_Data

Table 15-2. Header_Data (page 1 of 2)

Base	INT[0:-1]	The base from which the offsets to other data areas are calculated. Base indicates the allocation of a placeholder, not data. It is used for reference for all offsets and VAR-STRING fields within the messages.
Event_Type	INT	An enumeration describing the type of Message_Data in the message. Valid message types are Access_Control, Logon, Password Quality, and Logon^Abort.
MSG_Version	INT	The version of this message structure. This value can be either 1 or 2. If 1, the status field is 23. If 2, the status field is as described in Table 15-7. The version is established by Safeguard. The event-exit process is responsible for checking the message version and rejecting any message it does not recognize. The rejection is returned in the Error field of this header.
Min_MSG_Version	INT	The oldest version of the message that must be supported by the event-exit process in order to interpret this message. Currently, this value is 1.
Subject_Data	INT	The byte offset to the subject data associated with this message. It must be an even number. Set to 0 in response from the event exit.
Message_Data	INT	The byte offset to the message data associated with this message. It must be an even number. Set to 0 in response from the event exit if there is no response message data. Set to the data offset by the event exit if there is message response data.
SSID	SSID-STRUCT	Subsystem ID of the sender in standard SSID format. This is the Safeguard SSID (ZSFG).
Timestamp	TIMESTAMP (FIXED)	The time the message is sent, in Greenwich mean time (GMT).
OriginSystemNumber	NT(32)	The system number of the system originating the request.

Table 15-2. Header_Data (page 2 of 2)

Message_Tag	INT(32)	Indicates continuity of ongoing dialog for challenge/response or password dialog interactions. The initial value is 0. This field is filled in by the event-exit process so that it can identify different events when it is handling multiple message dialogs. Safeguard preserves the message tag and returns it to the event exit in subsequent messages during ongoing dialog. For Logon^Abort messages, this tag identifies the particular logon dialog that has just been terminated.
Error	INT	A return value that indicates the event exit's response to the message. The value is always 0 when the message is sent. Valid return values are as follows: 0 = OK (The event exit successfully processed the message and is returning a valid status.) 3501 = Message size exceeded maximum expected message size. 3503 = Event exit process does not support this message data type. 3505 = Event exit's supported message version is lower than Min_MSG_Version in this message.
Status	INT	Status of the reply message. This field is valid only if the Error field contains the value 0. The valid values for this field are: Access Control (result of access attempt) 1 = yes 3 = no 2 = no record Logon (result of logon attempt) 0 = authenticated 20 = denied 70 = continue authentication 500-999 = defined by event exit (A status of 500-999 is recognized by USER_AUTHENTICATE only when Error does not equal 0.)
FEA	22 bytes	A future expansion area; currently filled with zeros.

Table 15-3 on page 15-17shows the structure of the subject data sent from the Safeguard subsystem to the event-exit process. This subject data is always present. It is not returned by the event-exit process.

Table 15-3. Subject_Data (page 1 of 2)

UserName	VAR-STRING	The subject's user name, in external format.
UserID	INT(32)	The user ID associated with the user name. For authentication requests, this is the user ID of the process calling USER_AUTHENTICATE_ or VERIFYUSER.
CAID	INT(32)	CAID of the subject. For authentication requests, this is the CAID of the process calling USER_AUTHENTICATE_ or VERIFYUSER.
SubjectTerm	VAR-STRING	Home terminal of the subject. For some access control messages, SubjectTerm is not always present when the message is sent to the Safeguard subsystem. Therefore it will not be present in the Subject_Data in those instances.
Domain	INT(32)	The security domain from which the request originated. Currently, this is the same as the OriginSystemNumber.
SubjectProcessName/ID	PHANDLE	The process handle of the subject process. Some access control requests do not contain a process handle. For a list of these requests, see Table 15-9 on page 15-23.
AuthType	UINT	Authorization type: 0 = unauthenticated 1 = reserved (not used) 2 = locally authenticated 3 = remotely authenticated
IsUndeniable	BOOLEAN	True if the subject is undeniable. An undeniable user is one who is a locally authenticated member of the super group.
IsAlias	BOOLEAN	True if the UserName is an alias.
RealUserID	INT(32)	Reserved for future use. Zero-filled.
SavedUserID	INT(32)	Reserved for future use. Zero-filled.
RealGroupID	INT(32)	Reserved for future use. Zero-filled.
EffectiveGroupID	INT(32)	The subject's effective group ID. Currently, this is the administrative group number of the subject.
SavedGroupID	INT(32)	Reserved for future use; currently filled with zeros.

Table 15-3. Subject_Data (page 2 of 2)

AuthNode	INT(32)	Last authenticated node number for a remote subject, zero otherwise. Valid only if associated AuthNodeValid is set (True).
GroupList	VAR-STRING	The list of groups of which this subject is a member. Currently, the subject's administrative group is the only group in this list.
AuthNodeValid	BOOLEAN	True indicates the field AuthNode contains a valid remote node value.
FEA	52 bytes	A future expansion area; currently filled with zeros.

Table 15-4 shows the structure of the message data sent from the Safeguard subsystem to the event-exit process for an access control event. The message data is present only for an access control event. The event-exit process does not return this data in its response.

Table 15-4. Access_Data (Access Control Message_Data) (page 1 of 2)

Objecttype	ENUM	A valid Safeguard object type value: disk file, volume, subvolume, device, subdevice, process, or subprocess.
Objectname	VAR-STRING	The name of the object, in external format.
Operation	INT	The action or verb being executed. Examples include Open, Create, Rename, and Alter. For a complete list of operations, see Table 15-10 on page 15-24.
OperationModifier	INT	A modifier to the operation. Examples include Read, Write, Create, and Execute. For a complete list of modifiers, see Table 15-10 on page 15-24.
SQL_Object	BOOLEAN	True indicates that the object is an SQL object.
AccessCheckOnly	BOOLEAN	True indicates that the request is checking for possible access. False indicates that the request is for a real access attempt.
OldFilename (Rename)	VAR-STRING	The file name of the object of a rename request, in external format.
NewOwner (Give)	INT(32)	User ID of the recipient of a SETMODE (function 2) Give request. It is present only for that SETMODE operation. Otherwise, it is null.

Table 15-4. Access_Data (Access Control Message_Data) (page 2 of 2)

Altervalid	INT	Used by requests for ChangeOwner (GIVE), PROGID, and LICENSE. These three requests can be present in one physical request. One bit is set for each of the three requests. The value is 0 if none of the three are present. The bit settings are as follows: <0> - 1 if PROGID <1> - always 0, not used <2> - 1 if LICENSE <3> - always 0, not used <4> - 1 if Change Owner (GIVE) <5> - 1 if TRUST <6:15> - always 0, not used
Alterdata	INT	The values for PROGID and LICENSE. Valid only if the associated Altervalid bit is set. The bit settings are: <0> - 1 = PROGID ON <1> - always 0, not used <2> - always 0, not used <3> - 1 =LICENSE ON <4:5> - TRUST 0=OFF, 1=ME, 2= SHARED <6:15> - always 0, not used
CPUNumber	INT	For Process_Create_ requests, the number of the CPU in which the new process will be created. For all other requests, the value is -1.
ObjFilename	VAR-STRING	The program file name of a process to be started/stopped, in external format.
OssPathname	INT	TRUE indicates that SEEP can convert the extracted ObjFilename to the OSS pathname format by calling guardian procedure call FILENAME_TO_PATHNAME_.
FEA	30 bytes	A future expansion area; currently filled with zeros.

Table 15-5 on page 15-20 shows the structure of the message data sent from the Safeguard subsystem to the event-exit process for a logon event. This message data is present only for a logon event. The event-exit process does not return this data in its response.

Table 15-5. Logon_Data (Logon Message_Data Interactive/Programmatic) (page 1 of 2)

Dialogue_Possible	BOOLEAN	True indicates that the request came from a process that is calling USER_AUTHENTICATE_ and is capable of engaging in dialog with the event-exit process. False indicates that the requestor cannot understand anything except Yes or No (from VERIFYUSER or from callers of USER_AUTHENTICATE_ that cannot handle a dialog).
Logon_UserID_Exists	BOOLEAN	The logon user exists in the Safeguard database. This is always true.
Logon_Name	VAR-STRING	The user name, in external format, of the user attempting to log on. This can be a user name or an alias. Valid only if Logon_UserID_Exists is true.
Logon_UserID	INT(32)	The user ID associated with the Logon_Name. Valid only if Logon_UserID_Exists is true.
IsAlias	BOOLEAN	True indicates that the Logon_Name is a user alias.
Password	VAR-STRING	The password provided by the user at logon time when dialog is not possible and the call is from VERIFYUSER. Because of certain constraints in the Safeguard software, this is a 64-character field in clear text (not encrypted). This field is blank if the caller was USER_AUTHENTICATE_, regardless of the Dialogue_Possible setting.
Logon_Source	VAR-STRING	The subject terminal name (device name for Safeguard terminal). This data is not verified, and the integrity of this field is not guaranteed. A caller of USER_AUTHENTICATE_ can fill in this field with any terminal name.

Table 15-5. Logon_Data (Logon Message_Data Interactive/Programmatic) (page 2 of 2)

Logon_Name_Phrase	VAR-STRING	The user name string typed by the user. If the user entered a password, also includes the password phrase, separated from the name by a comma. Maximum length is 256 bytes. This is the string from which the Logon_Name and Logon_UserID are decomposed. The field is present only when the caller is USER_AUTHENTICATE_. This field is blank if the caller was VERIFYUSER.
Password_Response_Phrase	VAR-STRING	The user's response to a challenge or password request. Maximum length is 80 bytes. This field is blank if the caller is VERIFYUSER.
Logon_Attempts_Num	INT	The current number of unsuccessful logon attempts for this user ID. It is the number of unsuccessful logon attempts since the last successful logon attempt.
Authenticate_Only	BOOLEAN	Indicates a request from VERIFYUSER or USER_AUTHENTICATE_ for information only. True indicates information request. False indicates logon request.
Options_Word	INT	The logon options word that was passed to USER_AUTHENTICATE_, bits 2-15. Bits 0 and 1 of the options word are always zero.
IP_Address	VAR-STRING	The IP address of the client if provided by the caller of the User_Authenticate_ API. The IP address can be either IPv4 or IPv6 with maximum length of 128 bytes. The field is blank if the IP address is not specified by the caller of User_Authenticate_ API.
FEA	24 bytes	A future expansion area; currently populated with zeros.

Note. IP_Address attribute is supported only on systems running on J06.16 and later J-series RVUs, and H06.27 and later H-series RVUs.

Table 15-6 on page 15-22 shows the structure of the message data sent from the Safeguard subsystem to the event-exit process for a password change event. This message data is present only for a password change event resulting from an attempt to change a password with the PASSWORD program or with the Safeguard ADD

USER, ALTER USER, ADD ALIAS, or ALTER ALIAS commands. The event-exit process does not return this data in its response.

Table 15-6. Password_Change_Data (Change Message_Data from PASSWORD Program)

Target_User	VAR-STRING	The user name or alias, in external format, of the user whose password is being changed. This can be a user name or an alias.
Target_UserID	INT(32)	The user ID associated with Target_User.
IsAlias	BOOLEAN	True indicates that Target_User is a user alias.
Password	STRING	The clear text password to be evaluated for password quality. This password comes from the SAFECOM or SPI ADD USER, ALTER USER, ADD ALIAS, or ALTER ALIAS command, or from a password change initiated by the PASSWORD program. Because of certain constraints in the Safeguard software, this is a 64-character field.
FEA	50 bytes	A future expansion area; currently filled with zeros.

Table 15-7 shows the structure of the message reply data sent from the event-exit process to the Safeguard subsystem for a logon event. This reply overlays the message data sent by the Safeguard subsystem for a logon event.

Table 15-7. Logon_Response_Data (Interactive/Programmatic Logon) (page 1 of 2)

Challenge_Phrase	VAR-STRING	Phrase to be returned to the caller for display on the user's terminal.
Logon_Blind_Read	BOOLEAN	True if the response to Challenge_Phrase is to be passed in the blind (character echo disabled).
Delay_Value*	INT	If nonzero, indicates the number of Delay_Units to delay the user before accepting another logon attempt. Delay_Value should be a nonzero positive integer. If it is negative, a delay is not enforced. Delay_Value should be initialized appropriately. Failing to do so can lead to undesired delay.
Delay_Units*	ENUM	The unit of execution for Delay_Value. Valid units are SECONDS, MINUTES, HOURS, DAYS, WEEKS, and MONTHS.
Password_Returned	BOOLEAN	True if the Password field contains a generated password to be filed in the Safeguard database.

Table 15-7. Logon_Response_Data (Interactive/Programmatic Logon) (page 2 of 2)

Password	VAR-STRING	The 64-character password string returned from the event exit to be filled in the Safeguard database. Blanks if the password is not returned. This field is filed without checking by Safeguard.
Status	INT	The response of the SEEP to the message returns a number that indicates an additional status information when the Status field in the Header_Data is 0 (success), 20 (denied), or 70 (continue dialog). Appropriate values are the same as those described for the status parameter of USER_AUTHENTICATE_ in the *Guardian Procedure Calls Reference Manual*. Return zero if none of those status values apply or if the Status field in the Header_Data is not 0, 20, or 70.

*Set Delay_Value and Delay_Units in order to enforce the delay.

Table 15-8 shows the structure of the message reply data sent from the event-exit process to the Safeguard subsystem for a password change event. This reply overlays the message data that was sent by the Safeguard subsystem for a password change event.

Table 15-8. Password_Change_Response

Err_Message_Phrase	VAR-STRING	The text returned to the user to describe the error encountered. Maximum length is 255 characters.

Table 15-9 lists the access control requests that do not contain process handles when they are passed to the Safeguard subsystem.

Table 15-9. Requests Without Process Handles

CREATE	Disk file
PURGE	Disk file
RENAME	Disk file
SETMODE	Disk file
CREATE	SQL Table
CREATE	SQL Pview
CREATE	SQL Label
CREATE	SQL IXview
PURGE	SQL Label
RENAME	SQL Table
RENAME	SQL Pview
RENAME	SQL Label

Table 15-10 lists operations and modifiers for access control events.

Table 15-10. Authorization Operations and Modifiers

CREATE		
OPEN	READ	
OPEN	WRITE	
OPEN	WRITEREAD	
OPEN	EXECUTE	
OPEN	CREATE	For Dialect_Zero compatibility with FileSystem READ request, which is mapped to OPEN in Safeguard SMON.
OPEN	PURGE	For Dialect_Zero compatibility with FileSystem READ request, which is mapped to OPEN in Safeguard SMON.
OPEN	ONLINEDUMP	An HP NonStop Transaction Management Facility (TMF) modifier provided for compatibility with SMON.
OPEN	BACKOUT	A TMF modifier provided for compatibility with SMON.
OPEN	ROLLFORWARD	A TMF modifier provided for compatibility with SMON.
RENAME		
PURGE		
CLOSE		
ALTER	GIVE	ALTER is the mapping of the Setmode operation for Enscribe files and the ALTER operation for SQL files.
ALTER	LICENSE	
ALTER	PROGID	
ALTER	COMPOSITE	Used with ALTER to indicate that one or more of GIVE, PROGID, or LICENSE is present in this request.
NEWPROCESS		Process creation.
STOP	VAR-STRING	HP NonStop process stop.

Design Considerations

This subsection describes the specific events that can be sent to the event-exit process, gives information about the handling of the various events, and provides guidelines for writing an event-exit process.

Security Requests Sent to the Event-Exit Process

Depending on how the event-exit process is configured, the following specific requests are passed to it by the Safeguard subsystem.

If ENABLE-AUTHORIZATION-EVENT is ON, any of the following events that involve objects under Safeguard protection are sent to the event-exit process for a ruling:

- Access attempts to all objects protected by the Safeguard software
- Create
- Give (Setmode function 2)
- License (Setmode function 98)
- Progid (Setmode function 1)
- Process Start
- Open
- Purge
- Rename
- Process Abend
- Process Stop
- Trust (Setmode function 265)

Note. Trust (Setmode function 265) event is sent to the event-exit process only in H06.16 and later H-series RVUs.

If ENABLE-AUTHENTICATE-EVENT is ON, the following events are sent to the event-exit process for a ruling:

- Programmatic logon (USER_AUTHENTICATE_, VERIFYUSER including Verify Only Mode)
- Interactive logon (Safeguard terminals, TACL using USER_AUTHENTICATE_)
- Logon password-related authentication events:
  - Password change at logon
  - Forced password change at logon
  - Generated password at logon

  The event exit process must be able to return an error message describing denial of logon password change.

If ENABLE-PASSWORD-EVENT is ON, the following events are sent to the event-exit process for a ruling:

- Password change with the PASSWORD program
- Password change with the ADD USER, ALTER USER, ADD ALIAS, or ALTER ALIAS commands
- Password change interactive logon if ENABLE-AUTHENTICATE-EVENT is OFF

Processing of Authorization Requests

When ENABLE-AUTHORIZATION-EVENT is ON, authorization requests are routed to the event-exit process. When a subject attempts to access an object, the request flows through the application to the appropriate subsystem software, which calls the privileged library procedure PROTECTION_CHECK_. This request is forwarded by PROTECTION_CHECK_ to the Safeguard SMON, which in turn routes the request to the event-exit process for evaluation. The message links between the SMON and the event-exit process are file-system messages (WRITEREAD[X]) in the format shown in Table 15-2 on page 15-15 through Table 15-8 on page 15-23.

For the event-exit and Safeguard security policies to interact in a meaningful manner, both policies must support the same types of rulings. Safeguard authorization supports rulings of YES, NO, or NORECORD (no opinion). The event-exit process must support these same rulings. If the event-exit process has no opinion on the ruling for a given object, it must respond with NORECORD in the Status field of the Header_Data message. If the event exit responded YES in this instance, a false positive would be passed to the Safeguard software, and Safeguard might grant access to a disk file that should have been controlled by Guardian security.

If the event-exit process rules NO on the access attempt, the SMON returns the denial to PROTECTION_CHECK_ without further processing.

If the event-exit process rules YES or NORECORD on the access attempt, the Safeguard software performs its own access check and returns the combination of the two results to PROTECTION_CHECK_. Therefore, the event-exit process cannot unilaterally grant access to an object if that access is denied by a Safeguard protection record. If the Safeguard access check also results in NORECORD, Guardian security applies.

Table 15-11 shows results of access attempts based on different rulings from the event-exit process and the Safeguard subsystem. The final access control result appears in the PROTECTION_CHECK_ column for all cases except those in which the

PROTECTION_CHECK_ result is NORECORD. When NORECORD is the PROTECTION_CHECK result, the final result appears in the Guardian column.

Table 15-11. Decision Table for Event Exit, Safeguard, and Guardian Results

Event Exit Ruling	Safeguard Ruling	Protection_Check _Result	Guardian Security Ruling
YES	YES	YES	Not consulted
YES	NO	NO	Not consulted
YES	NORECORD	YES	Not consulted
NO	Not consulted	NO	Not consulted
NO	Not consulted	NO	Not consulted
NO	Not consulted	NO	Not consulted
NORECORD	YES	YES	Not consulted
NORECORD	NO	NO	Not consulted
NORECORD	NORECORD	NORECORD	YES or NO
Event exit disabled	YES	YES	Not consulted
Event exit disabled	NO	NO	Not consulted
Event exit disabled	NORECORD	NORECORD	YES or NO
Event exit disabled	Safeguard disabled	NORECORD	YES or NO
Event exit disabled	Safeguard disabled	NORECORD	YES or NO*

* If an object has a Safeguard protection record and the Safeguard subsystem is disabled, access rulings for that object are as described for the STOP SAFEGUARD command in Section 16, Safeguard Subsystem Commands.

Timeout Policy for Authorization

If the event-exit process does not respond to a request within the configured time interval, the SMON assumes that a problem has occurred and continues processing as follows.

If the authorization request is from an undeniable user when a timeout occurs, a response of YES is assumed, and the access attempt is allowed to proceed, subject to a Safeguard access check as described in Processing of Authorization Requests on page 15-26. Locally authenticated super-group members are considered undeniable users. An EMS message is sent to indicate that an undeniable user has timed out, thereby prompting the undeniable user to disable the malfunctioning event-exit process.

If the authorization request is from a deniable user when the time out occurs, and if the attribute TIMEOUT-ALL-AUTHZREQ is enabled, then the SEEP response is treated as NO, and the control returns to the requestor without any further processing by the Safeguard with the status as “security violation”. An EMS message is sent to indicate that a deniable user has timed out. If the attribute TIMEOUT-ALL-AUTHZREQ is disabled, then the deniable user waits indefinitely for a response from the event-exit process. The requestor process (including the entire thread), initiating the authorization check, will hang.

Note. The TIMEOUT-ALL-AUTHZREQ attribute is supported only on systems running H06.26 and later H-series RVUs and J06.15 and later J-series RVUs.

Other Error Handling for Authorization

Other problems that cause timeout behavior are I/O errors, disabling of the event exit, and invalid data received from the event-exit process.

I/O errors can occur when the event-exit process halts before responding to a request, or when it is enabled but is down or restarting and the open is incomplete. In either of these events, the request is resubmitted to the event-exit process once the open is established. However, the timer is still running on these requests, and a timeout is likely to occur. EMS messages will be sent to identify these errors.

If the event-exit process is disabled while a request is pending, the request is allowed to complete, providing it does so within the timeout interval. If a timeout occurs and the request is from a deniable user, a ruling of NORECORD and a status of NOLINK is returned to PROTECTION_CHECK_. If a timeout occurs and the request is from an undeniable user, a ruling of YES is assumed, and the access attempt is allowed to proceed, subject to a Safeguard access check.

If invalid data is returned in a reply from the event-exit process, an EMS message is sent to identify the problem. If the request is from a deniable user, a ruling of NORECORD and a status of NOLINK is returned. If the request is from an undeniable user, a ruling of YES is assumed, and the access attempt is allowed to proceed, subject to a Safeguard access check as described in Processing of Authorization Requests on page 15-26.

Warning Mode Interaction

Safeguard warning mode has no effect on rulings made by the event-exit process. Warning mode rulings are applied only after the event-exit process has ruled and the SMON performs a subsequent access check. For example, if warning mode is in effect and the event-exit process denies access, the access attempt is denied. If warning mode is in effect and the event-exit process grants access, but the SMON denies access, the access is allowed because of warning mode. For more information about warning mode, see the *Safeguard Administrator’s Manual*.

Auditing of Authorization Events

If the Safeguard software is configured for auditing of the object being accessed, audit records are generated for access attempts on that object when Safeguard is involved in the ruling.

If the event-exit process responds NO to an access attempt, the failure is not audited in Safeguard because the event exit and SMON auditing are not integrated. If the event-exit process responds YES or NORECORD, the Safeguard subsystem rules on the request, and auditing is performed as specified for the object.

Therefore, the basic concept in auditing is that if Safeguard is involved in the ruling, auditing is applied as specified. If Safeguard is not involved in the ruling, no auditing is performed.

Processing of Authentication Requests

When ENABLE-AUTHENTICATION-EVENT is ON, authentication requests are routed to the event-exit process. Both interactive and programmatic logon authentication requests are sent to the event-exit process. Unlike authorization events, the rulings on these events are the sole responsibility of the event-exit process. The Safeguard software does not participate in authentication rulings.

However, if the Safeguard subsystem is configured to communicate with the $CMON process, it sends a prelogon message to $CMON and awaits a reply before routing the authentication request to the event-exit process. $CMON has the option of denying the logon attempt prior to authentication by the event-exit process. Similarly, if Safeguard is configured to do so, it sends a logon message to $CMON after authentication occurs. $CMON again has the option of denying the logon attempt even after the user has been authenticated.

Processing of Interactive Authentication

For interactive logon attempts, a process such as TACL provides the logon input and authentication request in a call to USER_AUTHENTICATE_. This input is forwarded by USER_AUTHENTICATE_ to the Safeguard $ZSMP process, which in turn routes it to the event-exit process for evaluation. If the interactive logon attempt occurs at a Safeguard terminal, the Safeguard software captures the input directly, and $ZSMP routes it to the event-exit process. USER_AUTHENTICATE_ is not involved when the logon attempt occurs at a Safeguard terminal.

The event-exit process can approve or deny the logon request, or it can engage in a challenge/response dialog before approving or denying the request. Additionally, the event-exit process can return a generated password as part of a password change dialog. The Safeguard software does not check passwords or otherwise participate in the authentication. It only routes messages between the event-exit process and USER_AUTHENTICATE_. When the authentication is complete, the Safeguard software updates the last logon time and logon failure count in the user’s record in the Safeguard database. It also files the new password if a password change occurred and the event-exit process requested filing of the password.

The password-quality exit is separate from the authentication exit, and it is not invoked by the Safeguard software during an authentication event. For more information, see Processing of Password-Quality Requests on page 15-31.

The event-exit process is responsible for prompting the user for verification of a new password and for storing passwords in its own database. If a new password is collected by the event-exit process, it can inform the Safeguard subsystem of this change after authentication is complete. For more information, see User Database Synchronization on page 15-32.

Processing of Programmatic Authentication

In programmatic logon attempts, a process provides the logon input and authentication request in a call to VERIFYUSER or USER_AUTHENTICATE_. This input is forwarded to the Safeguard $ZSMP process, which in turn routes it to the event-exit process for evaluation.

Programmatic logon attempts handled by VERIFYUSER do not support an authentication dialog or password generation. When the Safeguard software passes this request to the event-exit process, it includes an indicator noting that this attempt is incapable of engaging in a dialog. The event-exit process can only grant or deny the authentication request.

Logon^Abort Processing

A Logon^Abort can occur during the processing of either an interactive or programmatic authentication attempt. The $ZSMP process sends a Logon^Abort message to the event-exit process if either of these events occurs:

- At a Safeguard terminal, the user presses the BREAK key, or an I/O error occurs during the logon dialog.
- During a logon attempt processed by USER_AUTHENTICATE_, the logon dialog times out because it takes longer than two minutes to complete. (The user takes too long to supply input.)

Logon^Abort is indicated by the Event_Type field in the Header_Data message sent from $ZSMP. The event-exit process checks the Message_Tag field in this message to determine which logon session aborted. The purpose of this message is to allow the event-exit process to deallocate the resources it allocated to process the authentication attempt.

Timeout Policy for Authentication

If the event-exit process does not respond to a request within the configured time interval, $ZSMP denies the authentication request. An EMS message indicates a user has timed out, thereby indicating a problem with the event-exit process.

Other Error Handling for Authentication

Other problems that cause timeout behavior are I/O errors, disabling of the event exit, and invalid data received from the event-exit process.

I/O errors can occur when the event-exit process halts before responding to a request, or when it is enabled but down or restarting and the open is incomplete. In these instances, all user requests are denied. EMS messages identify these errors.

If the event-exit process is disabled while an authentication request is pending, the request is allowed to complete, providing it does so within the timeout interval. If a timeout occurs, the request is denied.

If invalid data is returned in a reply from the event-exit process, the request is denied, and an EMS message identifies the problem.

Auditing of Authentication Events

If the Safeguard software is configured for auditing of user authentication attempts, audit records are generated for authentication events.

Processing of Password-Quality Requests

When ENABLE-PASSWORD-EVENT is ON, certain password-change events are routed to the event-exit process. The password-quality exit allows passwords to be subjected to custom validation. Rules that supplement the Safeguard password controls can be applied to password validation. If password rules are disabled in the Safeguard configuration record, validation by the password-quality exit effectively replaces Safeguard password controls.

The password-quality exit is separate from the authentication-exit, and it is not invoked by the Safeguard software during an authentication event. Its sole purpose is password validation. To make use of the password-quality exit during authentication, the authentication process must be written so that it calls or incorporates the logic of the password-quality exit.

The $ZSMP process receives password requests from the PASSWORD program when a password is created or changed. It also receives these requests from the following Safeguard commands: ADD USER, ALTER USER, ADD ALIAS, and ALTER ALIAS. The $ZSMP routes these requests to the event-exit process if ENABLE-PASSWORD-EVENT is ON. If ENABLE-AUTHENTICATION-EVENT is OFF when ENABLE-PASSWORD-EVENT is ON, the $ZSMP also sends password changes that occur during interactive logon dialog.

The event-exit process can only accept or deny the password. It can also send a message to accompany the acceptance or denial. The event-exit process cannot return generated passwords and engage in additional dialog for this event.

Timeout Policy for Password-Quality Requests

If the event-exit process does not respond to a request within the configured time interval, $ZSMP assumes that a problem has occurred and continues processing as follows.

If the password-quality request is from an undeniable user when a timeout occurs, the request is removed from the outstanding queue, and the attempt is allowed to proceed with the Safeguard software performing the password-quality check. Super-group members are considered undeniable users. An EMS message indicates an undeniable user has timed out, thereby prompting the undeniable user to disable the malfunctioning event-exit process.

If the password-quality request is from a deniable user when a timeout occurs, the attempt is denied. An EMS message indicates a deniable user has timed out, thereby indicating a problem with the event-exit process.

User Database Synchronization

The event-exit process is responsible for synchronization between its own user database and the Safeguard user database. To maintain consistency between the two databases, the user files need to be synchronized in these situations:

- During system startup (The event-exit user files must be initialized from the Safeguard user files.)
- When Safeguard user and alias authentication records are added or altered
- When user records are added or altered in the event-exit database
- When passwords are changed during authentication dialog with the event-exit process
- After the event-exit process has been stopped

General Procedure

Except for reading the Safeguard password field, all of these synchronization efforts can be handled with the following Safeguard SPI commands: ADD USER/ALIAS, ALTER USER/ALIAS, and INFO USER/ALIAS. Passwords must be handled in a more complex manner, described in Password Synchronization on page 15-33.

The event-exit process is responsible for propagating to the Safeguard database any changes that occur within its database. This can be accomplished using Safeguard SPI or a SAFECOM script.

To propagate changes from the Safeguard user database to the event-exit user database, the event-exit process must load its database using SPI INFO requests. This provides all information except passwords.

To remain synchronized with the Safeguard database, the event-exit process must poll the Safeguard database at reasonable intervals. The event-exit process must determine if new users have been added since the last polling. It also must check the last modified date in each user record to determine if the record matches that of the corresponding user in its own database. If a user record has changed, the event-exit process must collect the new information and mark the user in its own database if the password has changed.

Similarly, if the event-exit was disabled while the Safeguard subsystem was running, the event-exit process must poll the Safeguard database for changes.

Password Synchronization

The basic premise for database synchronization is that the event-exit process is responsible for keeping passwords synchronized in the two user databases.

Safeguard passwords are stored in an encrypted form, and HP does not export its encryption algorithm. The USER_AUTHENTICATE_ procedure is available to allow the caller to validate a password against the Safeguard user database.

For the Safeguard subsystem to accept passwords from the event exit, it must be configured in a manner consistent with the event exit's password management. For example, if the event-exit process is not applying the Safeguard password rules, the password rules in the Safeguard configuration must be disabled. Otherwise, passwords that are valid for the event exit are rejected when an attempt is made to file them in the Safeguard database.

One way to synchronize passwords is to require that all users change passwords at initial logon. The event-exit process can authenticate these users the first time through the database by calling the USER_AUTHENTICATE_ procedure in Authenticate Only mode at authentication time. Once a user is authenticated, the event-exit process can either store the entered password or force a password change. Turn off the AUTHENTICATE-FAIL-FREEZE and AUTHENTICATE-FAIL-TIMEOUT Safeguard configuration attributes during this authentication. Because USER_AUTHENTICATE_ checks these attributes, they might interfere with updates during password synchronization.

If the event-exit process collects a new password during a logon dialog, it can send the new password in a message response to the Safeguard database when authentication is complete. This allows the user databases to remain synchronized. If the password is not propagated to the Safeguard database, the user cannot log on if the event-exit process becomes disabled and authentication is performed by the Safeguard software.

You can force all password changes to be processed by the event-exit process. To do this, set ENABLE-PASSWORD-EVENT to ON to force all password change attempts through the password-quality exit. Then design the password-quality exit so that it rejects all attempts. This approach forces users to change their passwords during authentication. If the event-exit process handles authentication, it can capture all password changes, assuming the Safeguard software is running.

Event-Exit Design, Management, and Operation

The design of an event-exit process must adhere to these general requirements:

- The event-exit process must be multithreaded (able to handle multiple concurrent requests).

- The event-exit process can be a process pair to ensure its continuous availability to handle authorization requests. If it is not a process pair, the event-exit process is unavailable to handle requests during its initialization interval after a restart.
- Any user file maintained by the event-exit process must support 32-byte user alias names and their passwords.
- To avoid the possibility of deadlocks, the event-exit process must not perform waited operations after initialization.

The Safeguard $ZSMP starts and manages the event-exit process. Each time $ZSMP starts the event-exit process, it first attempts to kill any process with the same process name not started by the $ZSMP. An EMS message notes this action. If the $ZSMP finds that the event-exit process was not started by the $ZSMP, it does not send messages to the event-exit process until it successfully kills the process and restarts it. Be careful to avoid name collisions.

When the ENABLED flag is set to ON, $ZSMP attempts to start the event-exit process. Safeguard tries maximum of three attempts to start this process. For every attempt that fails, an EMS event is logged. If all attempts fail, Safeguard stops the process creation attempt and turns the ENABLED flag to OFF. An event is also logged if Safeguard encounters an error while resetting the ENABLED flag to OFF. If the event-exit process creation request fails, you must verify the ENABLED flag for the process. If the ENABLED flag has not been turned OFF by Safeguard, you must turn the flag OFF manually.

If the event-exit process stops abruptly, $ZSMP attempts to restart the process until it is successfully restarted or disabled. An EMS message is sent to the console informing the operator each time Safeguard attempts a restart. Safeguard tries maximum of three attempts to restart the process. To avoid a negative impact to the performance of the $ZSMP, Safeguard attempts each restart after a specified interval. If the process creation fails at the end of three attempts, the ENABLED flag is reset to OFF. Requests are not sent to the event-exit process until the restart is successful.

Note. HP recommends that when $ZSMP is attempting to start the event-exit process, Safecom must not be used to issue any other commands, until the result of the process creation is displayed at the Safecom prompt.

If the processor in which the event-exit process is running becomes unavailable, the backup CPU is used as the primary CPU. If neither the primary or backup CPU is available, the event-exit process is restarted in the same CPU as the $ZSMP.

$ZSMP opens the event-exit process with the name $*name*.#ZSEEP, where *name* is the name assigned in the event-exit configuration record. Because the event-exit process can be opened by processes other than Safeguard security processes, the subdevice name #ZSEEP allows the event-exit process to determine the intention of the opener. It also allows the event-exit process to determine which message protocol is being used.

To avoid deadlocks, the event-exit process is a security process. Messages received from the event-exit process are not be sent to the event-exit process. Child processes of the event-exit process are not security processes. To avoid deadlocks, the event-exit process must maintain an internal list of its child processes and not forward their own requests to them.

Once the event-exit process has responded to the open request of an SMON, it must not perform waited I/O.

16

Safeguard Subsystem Commands

This section describes the commands that affect the Safeguard subsystem itself.

Table 16-1 gives a brief summary of these Safeguard subsystem commands.

Table 16-1. Safeguard Subsystem Command Summary

Command	Description
STOP SAFEGUARD	Disables Safeguard authorization checks and access auditing for all local protected objects. The security string for all protected disk files is ****, and those files are accessible only to the primary owner, the owner's group manager, and the super ID while Safeguard is stopped. Other system objects are unprotected. Only members of the SECURITY-ADMINISTRATOR security group can issue this command. If the SECURITY-ADMINISTRATOR group does not exist, only by the super ID the STOP SAFEGUARD command can be issued. If the Safeguard software is included with system generation, it cannot be stopped with the STOP command.
INFO SAFEGUARD	Displays the current global configuration attributes. Any user (local or remote) can issue this command.
ALTER SAFEGUARD	Changes the current global configuration attributes. Only members of the SECURITY-ADMINISTRATOR security group can issue this command. If the SECURITY-ADMINISTRATOR group does not exist, only local super-group members can issue the STOP SAFEGUARD command. If that security group has not been defined, only local super-group members can use the ALTER SAFEGUARD command.

Note. For backward compatibility, the Safeguard software accepts but ignores a START SAFEGUARD command. In some previous Safeguard product versions, this command was necessary to fully enable the Safeguard software.

Command Syntax

This section contains syntax descriptions of the Safeguard subsystem commands. Each command description contains these elements:

- A description of the function performed by the command, including any restrictions on who can use the command
- The syntax of the command, including descriptions of the command parameters and variables
- Considerations for the use of the command

STOP SAFEGUARD Command

STOP SAFEGUARD stops each SMON process, each SHP and the SMP pair. The command also stops an event-exit process if one is running. After these processes are stopped, disk files that have Safeguard protection can be accessed only by the primary owner, the owner’s group manager, and the super ID. Attempts to access other system objects are subject only to access controls provided by the standard Guardian security system.

Only members of the SECURITY-ADMINISTRATOR security group can use the STOP SAFEGUARD command. If that group has not been defined, only the local super ID can use the STOP SAFEGUARD command.

To restart the Safeguard software, you must start the SMP as described the *Safeguard Administrator’s Manual*.

Note. If the Safeguard software has been included as part of the operating system during system generation, the STOP SAFEGUARD command is accepted but ignored. In this case, the Safeguard software cannot be disabled while the system is operational.

```
STOP [ SAFEGUARD ]
```

disables Safeguard authorization checks and access auditing for all local protected objects and stops all SMON, all SHP, and SMP processes in the local system.

Note. A global attribute PROMPT-BEFORE-STOP on page 16-29, is supported on systems running on J06.16 and later J-series RVUs, and H06.27 and later H-series RVUs. If the attribute value is ON, a confirmation message is displayed to the user when the STOP command is issued at the SAFECOM prompt. This attribute is part of the global Safeguard configuration which can be set from SAFECOM, and has default value as OFF.

Considerations

- Disk file security following a STOP SAFEGUARD command

  After the execution of the STOP SAFEGUARD command, each disk file that was under Safeguard control becomes accessible only to the primary owner, the owner’s group manager, and the super ID. If it is necessary to reestablish Guardian security for these files, the super ID can do so by using FUP SECURE.

- Volume and subvolume security following a STOP SAFEGUARD command

  After the execution of the STOP SAFEGUARD command, any user can create a disk file on any subvolume in the system.

- Device security following a STOP SAFEGUARD command

  Following the execution of the STOP SAFEGUARD command, any user can access any device attached to the system.

- Named-process security following a STOP SAFEGUARD command

Following the execution of the STOP SAFEGUARD command:

- Any user can create a process with any legal process name.
- Any user can access any named process.
- Only the user identified by a named process’s creator accessor ID (CAID), that user’s group manager, and the local super ID can stop a named process.

- Effect of the STOP SAFEGUARD command on user authentication

  Following the execution of the STOP SAFEGUARD command, logon attempts are subject only to the standard Guardian security authentication. User IDs that are frozen, expired, or have expired passwords are no longer prevented from logging on. That is, if a user knows the proper password (if any), those IDs can be used to access the system.

- Effect of the STOP SAFEGUARD command on the SAFECOM command language

  Following the execution of the STOP SAFEGUARD command, any attempt to use SAFECOM to access the Safeguard database results in an error. The security database is inaccessible when the Safeguard software is stopped.

  When the Safeguard software is restarted, it enforces all access controls that were defined for system objects in the Safeguard object database at the time that it was stopped.

INFO SAFEGUARD Command

The INFO SAFEGUARD command displays the current global configuration attributes. Any user (local or remote) can execute this command.

```
INFO SAFEGUARD [ [ , ] option ] [ , option  ] ...
```

displays most of the current global configuration attributes. If the appropriate `option` is not specified, INFO SAFEGUARD does not display attributes that relate to auditing, the default command interpreter, communication with $CMON, and logon dialog. All attributes are described later in this section.

INFO SAFEGUARD also displays the following header line if an undeniable super ID has been specified during system generation:

```
SAFEGUARD IS CONFIGURED WITH SUPER.SUPER UNDENIABLE
```

This header line means that you cannot deny the super ID any Safeguard access authorities.

option

is one of:

```
GENERAL
DETAIL
AUDIT
CI
COMPARE
```

`GENERAL`

displays the same global configuration attributes as INFO SAFEGUARD with no *option* specified.

`DETAIL`

displays all of the global configuration attributes including those for auditing, the default command interpreter, communication with $CMON, and logon dialog.

`AUDIT`

displays only global configuration attributes that relate to auditing.

`CI`

displays only global configuration attributes that relate to the default command interpreter.

`COMPARE`

displays all those attributes that are already selected as a result of the other options namely including the GENERAL, AUDIT, DETAIL, CI attributes, and it displays along with the default values of each of those attributes.

Note. The COMPARE option is supported only on systems running J06.14 and later J-series RVUs and H06.25 and later H-series RVUs.

ALTER SAFEGUARD Command

The ALTER SAFEGUARD command alters the current global configuration attributes. Only members of the SECURITY-ADMINISTRATOR security group can execute the command. For more information regarding the SECURITY-ADMINISTRATOR security group, see Section 13, Security Group Commands. If that security group has not been defined, only local super-group members can use the ALTER SAFEGUARD command.

Attempts to change the Safeguard configuration are always audited.

```
ALTER SAFEGUARD [ , ] attribute [ , attribute ] ...
```

attribute

is one of:

```
AUTHENTICATE-MAXIMUM-ATTEMPTS [ n ]

AUTHENTICATE-FAIL-TIMEOUT [ n [ SECONDS ] ]
                              [ MINUTES ]
                              [ HOURS   ]
                              [ DAYS    ]
                              [ WEEKS   ]
                              [ MONTHS  ]

AUTHENTICATE-FAIL-FREEZE { ON | OFF }

PASSWORD-HISTORY  n

PASSWORD-MINIMUM-LENGTH  n

PASSWORD-MAY-CHANGE [ n [ DAYS [ BEFORE-EXPIRATION ] ] ]

PASSWORD-REQUIRED { ON | OFF }

PASSWORD-EXPIRY-GRACE [ n [ DAYS ] ]

PASSWORD-ENCRYPT { ON | OFF }

CHECK-DEVICE { ON | OFF }

CHECK-SUBDEVICE { ON | OFF }

DIRECTION-DEVICE { DEVICE-FIRST    }
                 { SUBDEVICE-FIRST }

COMBINATION-DEVICE { FIRST-RULE }
                   { FIRST-ACL  }
                   { ALL        }

ACL-REQUIRED-DEVICE { ON | OFF }

CHECK-PROCESS { ON | OFF }

CHECK-SUBPROCESS { ON | OFF }

DIRECTION-PROCESS { PROCESS-FIRST    }
                  { SUBPROCESS-FIRST }

COMBINATION-PROCESS { FIRST-RULE }
                    { FIRST-ACL  }
                    { ALL        }

ACL-REQUIRED-PROCESS { ON | OFF }

CHECK-VOLUME { ON | OFF }

CHECK-SUBVOLUME { ON | OFF }
```

```
CHECK-FILENAME { ON | OFF }

DIRECTION-DISKFILE { VOLUME-FIRST   }
                   { FILENAME-FIRST }

COMBINATION-DISKFILE { FIRST-RULE }
                     { FIRST-ACL  }
                     { ALL        }

ACL-REQUIRED-DISKFILE { ON | OFF }

ALLOW-DISKFILE-PERSISTENT { NORMAL | ALWAYS }

CLEARONPURGE-DISKFILE { ON | OFF }

{ AUDIT-AUTHENTICATE-PASS    } [ ALL    ]
{ AUDIT-AUTHENTICATE-FAIL    } [ NONE   ]
{ AUDIT-SUBJECT-MANAGE-PASS  } [ LOCAL  ]
{ AUDIT-SUBJECT-MANAGE-FAIL  } [ REMOTE ]
{ AUDIT-OBJECT-ACCESS-PASS   }
{ AUDIT-OBJECT-ACCESS-FAIL   }
{ AUDIT-OBJECT-MANAGE-PASS   }
{ AUDIT-OBJECT-MANAGE-FAIL   }
{ AUDIT-DEVICE-ACCESS-PASS   }
{ AUDIT-DEVICE-ACCESS-FAIL   }
{ AUDIT-DEVICE-MANAGE-PASS   }
{ AUDIT-DEVICE-MANAGE-FAIL   }
{ AUDIT-PROCESS-ACCESS-PASS  }
{ AUDIT-PROCESS-ACCESS-FAIL  }
{ AUDIT-PROCESS-MANAGE-PASS  }
{ AUDIT-PROCESS-MANAGE-FAIL  }
{ AUDIT-DISKFILE-ACCESS-PASS }
{ AUDIT-DISKFILE-ACCESS-FAIL }
{ AUDIT-DISKFILE-MANAGE-PASS }
{ AUDIT-DISKFILE-MANAGE-FAIL }
{ AUDIT-EXCLUDE-FIELD}{field-name} (only for systems running
J06.03 and later J-series RVUs,H06.14 and later H-series
RVUs,and G06.32 and later G-series RVUs.)

{ AUDIT-EXCLUDE-VALUE}{value}
                     {(value[:...])}(only for systems running
J06.03 and later J-series RVUs, H06.14 and later H-series
RVUs,and G06.32 and later G-series RVUs)

AUDIT-CLIENT-GUARDIAN { ON | OFF }(AUDIT-CLIENT-GUARDIAN is a
synonym for AUDIT-CLIENT-SERVICE)

AUDIT-OSS-FILTER { ON | OFF }(only for systems running J06.04
and later J-series RVUs, H06.15 and later H-series RVUs, and
G06.32 and later G-series RVUs)

AUDIT-TACL-LOGOFF { ON | OFF }(only for systems running
J06.08 and later J-series RVUs, H06.19 and later H-series
RVUs, and G06.32 and later G-series RVUs)
```

DYNAMIC-PROC-UPDATE { ON | OFF } (only for systems running J06.10 and later J-series RVUs and H06.21 and later H-series RVUs.)

CI-PROG [*prog-filename*]

CI-LIB [*lib-filename*]

CI-SWAP [$*vol*[*.subvol.filename*]]

CI-CPU [*n* | ANY]

CI-PRI [*n*]

CI-PARAM-TEXT [*text*]

CMON { ON | OFF }

CMONTIMEOUT [*n* [SECONDS]]

CMONERROR [ACCEPT | DENY]

BLINDLOGON { ON | OFF }

NAMELOGON { ON | OFF }

TERMINAL-EXCLUSIVE-ACCESS { ON | OFF }

WARNING-MODE { ON | OFF }

WARNING-FALLBACK-SECURITY { GUARDIAN | GRANT }

SYSTEM-WARNING-MODE { ON | OFF }

ALLOW-NODE-ID-ACL { ON | OFF }

CHECK-DISKFILE-PATTERN { OFF | ONLY | FIRST | LAST | MID}(only for systems running J06.08 and later J-series RVUs and H-06.18 and later H-series RVUs)

AUDIT-CLIENT-OSS { ON | OFF } (only for systems running G06.29 and later G-series RVUs and H06.08 and later H-series RVUs)

PASSWORD-ALGORITHM { DES | HMAC256 } (only for systems running G06.29 and later G-series RVUs and H06.06 and later H-series RVUs)

PASSWORD-MAXIMUM-LENGTH *n* (only for systems running G06.31 and later G-series RVUs and H06.08 and later H-series RVUs)

PASSWORD-COMPATIBILITY-MODE { ON | OFF } (only for systems running G06.31 and later G-series RVUs and H06.08 and later H-series RVUs)

PASSWORD-UPPERCASE-REQUIRED {ON / OFF} (only for systems running G06.31 and later G-series RVUs and H06.09 and later H-series RVUs)

PASSWORD-LOWERCASE-REQ {ON / OFF} (only for systems running G06.31 and later G-series RVUs and H06.09 and later H-series RVUs)

PASSWORD-NUMERIC-REQUIRED {ON / OFF}(only for systems running G06.31 and later G-series RVUs and H06.09 and later H-series RVUs)

PASSWORD-SPECIALCHAR-REQUIRED {ON / OFF}(only for systems running G06.31 and later G-series RVUs and H06.09 and later H-series RVUs)

PASSWORD-SPACES-ALLOWED {ON / OFF}(only for systems running G06.31 and later G-series RVUs and H06.09 and later H-series RVUs)

PASSWORD-MIN-QUALITY-REQUIRED {0 - 4}(only for systems running G06.31 and later G-series RVUs and H06.09 and later H-series RVUs)

PASSWORD-MIN-UPPERCASE-REQ *n* (only for systems running J06.11 and later J-series RVUs and H06.22 and later H-series RVUs)

PASSWORD-MIN-LOWERCASE-REQ *n* (only for systems running J06.11 and later J-series RVUs and H06.22 and later H-series RVUs)

PASSWORD-MIN-NUMERIC-REQ *n* (only for systems running J06.11 and later J-series RVUs and H06.22 and later H-series RVUs)

PASSWORD-MIN-SPECIALCHAR-REQ *n* (only for systems running J06.11 and later J-series RVUs and H06.22 and later H-series RVUs)

PASSWORD-APLHA-REQUIRED {ON / OFF} (only for systems running J06.11 and later J-series RVUs and H06.22 and later H-series RVUs)

PASSWORD-MIN-APLHA-REQ *n* (only for systems running J06.11 and later J-series RVUs and H06.22 and later H-series RVUs)

PASSWORD-ERROR-DETAIL {ON / OFF} (only for systems running J06.14 and later J-series RVUs and H06.25 and later H-series RVUs)|

PROMPT-BEFORE-STOP {ON / OFF } (only for systems running J06.16 and later J-series RVUs and H06.27 and later H-series RVUs)|

AUTHENTICATE-MAXIMUM-ATTEMPTS [*n*]

n defines the maximum number of failed authentication attempts allowed before the defined actions take place. The default value is 3. (Action is not taken until after three consecutive invalid attempts have been made.) A value of 0 specifies no limit to the number of failed logon attempts. A null entry for this attribute resets the value to the default value.

`AUTHENTICATE-FAIL-TIMEOUT [ n [ units ] ]`

n defines the length of time to suspend the logon process at the terminal when AUTHENTICATE-MAXIMUM-ATTEMPTS has been exceeded. The default value is 60 seconds. (A process is suspended for one minute after exceeding AUTHENTICATE-MAXIMUM-ATTEMPTS invalid attempts to log on.) *`units`* can be one of the following: SECONDS, MINUTES, HOURS, DAYS, WEEKS, MONTHS. A null entry for this attribute resets the value to the default value.

△ **Caution.** Because the command interpreter process at the terminal remains locked for the duration of the AUTHENTICATE-FAIL-TIMEOUT period, avoid specifying an unreasonably long period. The terminal is effectively not usable during this period. The only recovery is to start a new process at the terminal.

`AUTHENTICATE-FAIL-FREEZE { ON | OFF }`

defines whether to freeze a user ID automatically (as if FREEZE USER had been invoked) if AUTHENTICATE-MAXIMUM-ATTEMPTS has been exceeded against that user ID. The initial value is OFF. (User IDs are not automatically frozen.)

△ **Caution.** If you set AUTHENTICATE-FAIL-FREEZE ON, a user can freeze the user IDs of others by attempting to log on with those other user names or user IDs.

`PASSWORD-HISTORY n`

n defines the number of previous passwords to retain in a per-user-ID password database. Any new password must be different from all the previously retained passwords to be acceptable. The initial value is 0. (Passwords are not subject to a history.)

`PASSWORD-MINIMUM-LENGTH n`

n defines the minimum character length of a new password. (Present passwords are not affected.) The initial value is 0 and the maximum value is 8 for DES algorithm and 64 for HMAC256 algorithm.

Note. A password can be any length, including a null password. The initial value of PASSWORD-MINIMUM-LENGTH is six only on systems running G06.29 and later G-series RVUs and H06.06 and later H-series RVUs.

`PASSWORD-MAY-CHANGE [ n [ DAYS [ BEFORE-EXPIRATION ] ] ]`

n defines the number of days before the password expiration date in which the users can change their own password. If no password expiration date is in effect, users can change their own password at any time. A value of 0 also allows the password to be changed at any time. The default value is 0 (no restrictions on password change date). A null entry for this attribute resets the value to the default value.

If the PASSWORD-MAY-CHANGE period is greater than the PASSWORD-MUST-CHANGE period in a user authentication record, that user's password can be changed at any time.

Note. The owner of a user authentication record can always change the password. After the owner changes the password, the users can change their own password once before the PASSWORD-MAY-CHANGE setting is effective.

`PASSWORD-REQUIRED { ON | OFF }`

defines whether a password is required for a super ID or group manager ID to log on as another user. The initial value is OFF. (No password is required.)

`PASSWORD-EXPIRY-GRACE [` *n* `[ DAYS ] ]`

n defines the number of days after password expiration during which users can change their expired passwords during logon. The default value is 0 (no extension period). A null entry for this attribute resets the value to the default value.

PASSWORD-EXPIRY-GRACE can also be specified in individual user authentication records. If the value of this attribute is not specified in a user authentication record, the Safeguard software uses the value specified in the Safeguard configuration record.

`PASSWORD-ENCRYPT { ON | OFF }`

defines whether new passwords are stored in an encrypted form. Changing this setting does not affect current passwords. The initial value is ON.

Note. Passwords are stored unencrypted. Any process with access to the $SYSTEM.SYSTEM.USERID file can identify the current passwords. The initial value for PASSWORD-ENCRYPT is ON only on systems running G06.29 and later G-series RVUs and H06.06 and later H-series RVUs.

`CHECK-DEVICE { ON | OFF }`

defines whether the device ACL is consulted to determine access to devices and subdevices. The initial value is ON. (Device ACLs are consulted.)

`CHECK-SUBDEVICE { ON | OFF }`

defines whether the subdevice ACL is consulted to determine access to subdevices. The initial value is OFF. (Subdevice ACLs are not consulted.)

`DIRECTION-DEVICE { DEVICE-FIRST | SUBDEVICE-FIRST }`

defines the direction in which device and subdevice ACLs are consulted to determine access to devices and subdevices when both CHECK-DEVICE and CHECK-SUBDEVICE are ON. The initial value is DEVICE-FIRST.

`DEVICE-FIRST`

specifies that device ACLs are to be consulted before subdevice ACLs.

`SUBDEVICE-FIRST`

specifies that subdevice ACLs are to be consulted before device ACLs.

`COMBINATION-DEVICE { FIRST-RULE | FIRST-ACL | ALL }`

defines the method by which overlapping ACLs are resolved for access to devices and subdevices. COMBINATION-DEVICE is used in conjunction with DIRECTION-DEVICE to resolve access conflicts. The initial value is FIRST-ACL

`FIRST-RULE`

specifies that the Safeguard software is to determine access by searching the ACLs until it finds the user ID mentioned.

`FIRST-ACL`

specifies that the Safeguard software is to determine access based on the first ACL it finds.

`ALL`

specifies that all consulted ACLs must grant the requested access for the success of the operation.

`ACL-REQUIRED-DEVICE { ON | OFF }`

defines whether the absence of an ACL for a device or subdevice causes the denial of access to that device or subdevice. The initial value is OFF. (The absence of ACLs causes operation to revert to Guardian rules.)

`CHECK-PROCESS { ON | OFF }`

defines whether the process ACL is consulted to determine access to processes and subprocesses. The initial value is ON. (Process ACLs are consulted.)

`CHECK-SUBPROCESS { ON | OFF }`

defines whether the subprocess ACL is consulted to determine access to subprocesses. The initial value is OFF. (Subprocess ACLs are not consulted.)

`DIRECTION-PROCESS { PROCESS-FIRST | SUBPROCESS-FIRST }`

defines the direction in which process and subprocess ACLs are consulted to determine access to processes and subprocesses when both CHECK-PROCESS and CHECK-SUBPROCESS are ON. The initial value is PROCESS-FIRST.

`PROCESS-FIRST`

specifies that process ACLs are to be consulted before subprocess ACLs.

`SUBPROCESS-FIRST`

specifies that subprocess ACLs are to be consulted before process ACLs.

`COMBINATION-PROCESS { FIRST-RULE | FIRST-ACL | ALL }`

defines the method by which overlapping ACLs are resolved for access to processes and subprocesses. COMBINATION-PROCESS is used in conjunction with DIRECTION-PROCESS to resolve access conflicts. The initial value is FIRST-ACL.

`FIRST-RULE`

specifies that the Safeguard software is to determine access by searching the ACLs until it finds the user ID mentioned.

`FIRST-ACL`

specifies that the Safeguard software is to determine access based on the first ACL it finds.

`ALL`

specifies that all consulted ACLs must grant the requested access for the success of the operation.

`ACL-REQUIRED-PROCESS { ON | OFF }`

defines whether the absence of an ACL for a process or subprocess causes the denial of access to that process or subprocess. The initial value is OFF. (The absence of ACLs reverts operation to Guardian rules.)

`CHECK-VOLUME { ON | OFF }`

defines whether the volume ACL is consulted to determine access to volumes, subvolumes, and disk files. The initial value is OFF. (Volume ACLs are not consulted.)

`CHECK-SUBVOLUME { ON | OFF }`

defines whether the subvolume ACL is consulted to determine access to subvolumes and disk files. The initial value is OFF. (Subvolume ACLs are not consulted.)

`CHECK-FILENAME { ON | OFF }`

defines whether the disk file ACL is consulted to determine access to disk files. The initial value is ON. (Disk-file ACLs are consulted.)

`DIRECTION-DISKFILE { VOLUME-FIRST | FILENAME-FIRST }`

defines the direction in which the Safeguard software searches for an ACL to determine access to volumes, subvolumes, and disk files. DIRECTION-DISKFILE applies only when two or more of the following attributes are ON: CHECK-VOLUME, CHECK-SUBVOLUME, and CHECK-FILENAME. The initial value for DIRECTION-DISKFILE is VOLUME-FIRST.

`VOLUME-FIRST`

specifies that volume, subvolume, and disk file ACLs are consulted, in that order.

`FILENAME-FIRST`

specifies that disk file, subvolume, and volume ACLs are consulted, in that order.

`COMBINATION-DISKFILE { FIRST-RULE | FIRST-ACL | ALL }`

defines the method by which overlapping ACLs are resolved for access to volumes, subvolumes, and disk files. COMBINATION-DISKFILE is used in conjunction with DIRECTION-DISKFILE to resolve access conflicts. The initial value is ALL. (For more information about the evaluation of overlapping ACLs, see Appendix B, Disk-File Access Rules.)

`FIRST-RULE`

specifies that the Safeguard software is to determine access by searching the ACLs until it finds the user ID mentioned.

`FIRST-ACL`

specifies that the Safeguard software is to determine access based on the first ACL it finds.

`ALL`

specifies that all consulted ACLs must grant the requested access for the ultimate success of the operation.

`ACL-REQUIRED-DISKFILE { ON | OFF }`

defines whether the absence of an ACL for a volume, subvolume, or disk file causes the denial of access to that volume, subvolume, or disk file. The initial value is OFF. (The absence of a Safeguard protection record reverts operation to Guardian rules.)

`CLEARONPURGE-DISKFILE { ON | OFF }`

defines whether all disk files act as if the CLEARONPURGE file attribute had been set. The initial value is OFF. (Disk files are purged according to their individual CLEARONPURGE file attributes.) When `CLEARONPURGE-DISKFILE` is set to ON, all disk files act as if the CLEARONPURGE file attribute had been set.

The following considerations apply to the `CLEARONPURGE-DISKFILE` attribute:

- CLEARONPURGE is a waited operation because the DiscProcess (DP2) writes zeros to all allocated extents before returning to the purge application. DP2 might take up to a minute to execute CLEARONPURGE for a 30 megabyte file.
- Disk files include Safeguard Audit files, TMF Audit files, Event log files, Temporary disc files, and so on.
- $TMP creates and purges Audit trails.
- $TMP stalls if the CLEARONPURGE-DISKFILE is ON while purging an audit trail, because $TMP waits until DP2 finishes CLEARONPURGE operations. Begin, Abort, and End transactions also stall while $TMP is stalled.

`AUDIT-AUTHENTICATE-PASS [ LOCAL | REMOTE | ALL | NONE ]`

defines additional auditing for successful authentication attempts for users and aliases. This setting supplements the audit settings in user or alias authentication records. The default value is NONE. (Auditing is selected by the individual audit settings.)

The conditions specified for this attribute also apply to the systemwide auditing of automatic logoffs described in the *Safeguard Audit Service Manual*.

`AUDIT-AUTHENTICATE-FAIL [ LOCAL | REMOTE | ALL | NONE ]`

defines additional auditing for unsuccessful authentication attempts for users and aliases. This setting supplements the audit settings in user or alias authentication records. The default value is NONE. (Auditing is selected by the individual audit settings.)

`AUDIT-SUBJECT-MANAGE-PASS [ LOCAL | REMOTE | ALL | NONE ]`

defines additional auditing for successful attempts to manage user and alias authentication records. This setting supplements the audit settings in user or alias authentication records. The default value is NONE. (Auditing is selected by the individual audit settings.)

`AUDIT-SUBJECT-MANAGE-FAIL [ LOCAL | REMOTE | ALL | NONE ]`

defines additional auditing for unsuccessful attempts to manage user and alias authentication records and group definition records. This setting supplements the audit settings in user or alias authentication records. The default value is NONE. (Auditing is selected by the individual audit settings.)

`AUDIT-OBJECT-ACCESS-PASS [ LOCAL | REMOTE | ALL | NONE ]`

defines additional auditing for successful object accesses. This setting supplements the audit settings in all object protection records. The default value is NONE. (Auditing is selected by the individual audit settings)

This attribute can also affect auditing of some HP client subsystems. For more information, see the *Safeguard Audit Service Manual*.

`AUDIT-OBJECT-ACCESS-FAIL [ LOCAL | REMOTE | ALL | NONE ]`

defines additional auditing for unsuccessful object accesses. This setting supplements the audit settings in all object protection records. The default value is NONE. (Auditing is selected by the individual audit settings.)

This attribute can also affect auditing of some HP client subsystems. For more information, see the *Safeguard Audit Service Manual*.

`AUDIT-OBJECT-MANAGE-PASS [ LOCAL | REMOTE | ALL | NONE ]`

defines additional auditing for successful object authorization record accesses. This setting supplements the audit settings in all object protection records. The default value is NONE. (Auditing is selected by the individual audit settings.)

`AUDIT-OBJECT-MANAGE-FAIL [ LOCAL | REMOTE | ALL | NONE ]`

defines additional auditing for unsuccessful object authorization record accesses. This setting supplements the audit settings in all object protection records. The default value is NONE. (Auditing is selected by the individual audit settings.)

`AUDIT-DEVICE-ACCESS-PASS [ LOCAL | REMOTE | ALL | NONE ]`

defines additional auditing for successful device or subdevice accesses. This setting supplements the audit settings in all device and subdevice protection records. The default value is NONE. (Auditing is selected by the individual audit settings.)

This attribute can also affect auditing of some HP client subsystems. For more information, see the *Safeguard Audit Service Manual*.

```
AUDIT-DEVICE-ACCESS-FAIL [ LOCAL | REMOTE | ALL | NONE ]
```

defines additional auditing for unsuccessful device or subdevice accesses. This setting supplements the audit settings in all device and subdevice protection records. The default value is NONE. (Auditing is selected by the individual audit settings.)

This attribute can also affect auditing of some HP client subsystems. For more information, see the *Safeguard Audit Service Manual*.

```
AUDIT-DEVICE-MANAGE-PASS [ LOCAL | REMOTE | ALL | NONE ]
```

defines additional auditing for successful device or subdevice authorization record accesses. This setting supplements the audit settings in all device and subdevice protection records. The default value is NONE. (Auditing is selected by the individual audit settings.)

```
AUDIT-DEVICE-MANAGE-FAIL [ LOCAL | REMOTE | ALL | NONE ]
```

defines additional auditing for unsuccessful device or subdevice authorization record accesses. This setting supplements the audit settings in all device and subdevice protection records. The default value is NONE. (Auditing is selected by the individual audit settings.)

```
AUDIT-PROCESS-ACCESS-PASS [ LOCAL | REMOTE | ALL | NONE ]
```

defines additional auditing for successful process or subprocess accesses. This setting supplements the audit settings in all process and subprocess protection records. The default value is NONE. (Auditing is selected by the individual audit settings.)

This attribute can also affect auditing of some HP client subsystems. For more information, see the *Safeguard Audit Service Manual*.

```
AUDIT-PROCESS-ACCESS-FAIL [ LOCAL | REMOTE | ALL | NONE ]
```

defines additional auditing for unsuccessful process or subprocess accesses. This setting supplements the audit settings in all process and subprocess protection records. The default value is NONE. (Auditing is selected by the individual audit settings.)

This attribute can also affect auditing of some HP client subsystems. For more information, see the *Safeguard Audit Service Manual*.

```
AUDIT-PROCESS-MANAGE-PASS [ LOCAL | REMOTE | ALL | NONE ]
```

defines additional auditing for successful process or subprocess authorization record accesses. This setting supplements the audit settings in all process and subprocess protection records. The default value is NONE. (Auditing is selected by the individual audit settings.)

`AUDIT-PROCESS-MANAGE-FAIL [ LOCAL | REMOTE | ALL | NONE]`

defines additional auditing for unsuccessful process or subprocess authorization record accesses. This setting supplements the audit settings in all process and subprocess protection records. The default value is NONE. (Auditing is selected by the individual audit settings.)

`AUDIT-DISKFILE-ACCESS-PASS [ LOCAL | REMOTE | ALL | NONE ]`

defines additional auditing for successful disk file, volume, and subvolume accesses. This setting supplements the audit settings in the individual protection records. The default value is NONE. (Auditing is selected by the individual audit settings.)

This attribute can also affect auditing of some HP client subsystems. For more information, see the *Safeguard Audit Service Manual*.

`AUDIT-DISKFILE-ACCESS-FAIL [ LOCAL | REMOTE | ALL | NONE ]`

defines additional auditing for unsuccessful disk file, volume, and subvolume accesses. This setting supplements the audit settings in the individual protection records. The default value is NONE. (Auditing is selected by the individual audit settings.)

This attribute can also affect auditing of some HP client subsystems. For more information, see the *Safeguard Audit Service Manual*.

`AUDIT-DISKFILE-MANAGE-PASS [ LOCAL | REMOTE | ALL | NONE ]`

defines additional auditing for successful disk file, volume, and subvolume authorization record accesses. This setting supplements the audit settings in the individual protection records. The default value is NONE. (Auditing is selected by the individual audit settings.)

`AUDIT-DISKFILE-MANAGE-FAIL [ LOCAL | REMOTE | ALL | NONE ]`

defines additional auditing for unsuccessful disk file, volume, and subvolume authorization record accesses. This setting supplements the audit settings in the individual protection records. The default value is NONE. (Auditing is selected by the individual audit settings.)

`AUDIT-DISKFILE-PRIV-LOGON`

specifies conditions for auditing attempts to perform a priv logon on the system. This setting supplements the individual audit settings. The conditions can be ON or OFF. The default is OFF.

Note. This attribute is supported only on systems running H06.11 and later H-series RVUs.

`AUDIT-CLIENT-GUARDIAN { ON | OFF }`

defines whether the Safeguard software accepts Guardian-related audit records from HP privileged subsystems. These subsystems are known as clients. ON indicates that the Safeguard software accepts audit records from Guardian clients. OFF indicates that it does not accept the audit records from Guardian clients. The initial value is ON. For more information about client subsystem auditing, see the *Safeguard Audit Client Service Manual*.

Note. The AUDIT-CLIENT-GUARDIAN attribute is a synonym for AUDIT-CLIENT-SERVICE attribute.

`AUDIT-EXCLUDE-FIELD`

specifies the field name of an audit record. All NonStop client audit events containing the specified field name are not generated by the Safeguard subsystem. The default value is NONE.

Table 16-2 lists the different AUDIT-EXCLUDE-VALUES each AUDIT-EXCLUDE-FIELD can take.

Table 16-2. AUDIT-EXCLUDE-FIELD values (page 1 of 4)

AUDIT-EXCLUDE-FIELD	Values for AUDIT-EXCLUDE-VALUE
Operation	READ
	WRITE
	EXECUTE
	DELETE
	CREATE
	UPDATE
	CHANGE
	RENAME
	START
	STOP
	SUSPEND
	REVIVE
	OPEN
	CLOSE
	GRANT
	REVOKE
	PURGE
	SELECT
	INSERT
	REFERENCE
	CHANGE_OWNER
	BACK_OUT
	ONLINE_DUMP
	OTHER
	ABORT
	ADD
	ENABLE
	EXCLUDE
	INITIALIZE
	NEXT
	RESOLVE

Table 16-2. AUDIT-EXCLUDE-FIELD values (page 2 of 4)

AUDIT-EXCLUDE-FIELD	Values for AUDIT-EXCLUDE-VALUE
	● ACCEPT ● NEXTTAPE ● REJECT ● RESET ● SCRATCH ● SET ● USETAPE ● DEBUG ● CHANGEPRIORITY ● CHANGESTEPMOM ● ALTER ● GIVE ● LICENSE ● PROGID ● NEWPROCESS ● SECURITY ● COMPOSITE ● ACCESS ● DIRSEARCH ● KILL ● LINK ● OSSRESOLVE ● TRUST ● TACLLOGOFF
OUTCOME	● GRANTED ● DENIED ● MAYBE ● PASSED ● FAILED ● NORECORD

Table 16-2. AUDIT-EXCLUDE-FIELD values (page 3 of 4)

AUDIT-EXCLUDE-FIELD	Values for AUDIT-EXCLUDE-VALUE
	● OTHER ● PARTIAL_SUCCE ● PENDING ● WARNING
OBJECTTYPE	● DISKFILE ● SUBVOLUME ● VOLUME ● DEVICE ● SUBDEVICE ● PROCESS ● SUBPROCESS ● SUBSYSTEM ● COMMAND ● USER ● GUARDIAN_USER ● SQL_TABLE ● SQL_VIEW ● SQL_INDEX ● SQL_CATALOG ● USER_RECORD ● PROT_RECORD ● CONTROLLER ● PATH ● CONFIG_RECORD ● SFG_CONFIG_REC ● AUD_TR_CONFIG_REC ● TMF_TRANSACTION ● TMF_AUDITTRAIL ● TMF_TAPEMEDIA ● TMF_AUDITDUMP

Table 16-2. AUDIT-EXCLUDE-FIELD values (page 4 of 4)

AUDIT-EXCLUDE-FIELD	Values for AUDIT-EXCLUDE-VALUE
	● TMF_BACKOUT ● TMF_CATALOG ● TMF_DUMPS ● SFG_LU_RECORD ● USER_REMPASS ● TAPEMOUNT ● TAPEVOLUME ● SYSTEMDEVICE ● SHAREDSEGMNT ● GROUP ● SFG_PROC_RECORD ● DIRECTORY ● FIFO ● OSSDISKFILE ● OSSFILESET ● SOCKET ● SYMLINK ● TTY ● PROCESSGROUP ● OSSPROCESS
OWNERISREMOTE	● REMOTE ● LOCAL ● NONE ● UNKNOWN

The following AUDIT-EXCLUDE-FIELD values have dynamic variable names, therefore no enums are defined.

- OWNERUSERNAME
- OWNERUSERNUMBER
- SUBJECTUSERNAME
- SUBJECTUSERNUMBER

- SUBJECTSYSTEMNAME
- SUBJECTCREATORNAME
- SUBJECTCREATORNUMBER
- SUBJECTSYSTEMNUMBER
- SUBJECTPROCESSNAME
- SUBJECTAUTHLOCNAME
- SUBJECTTERMINALNAME
- SUBJECTAUTHLOCNUMBER
- CREATORUSERNAME
- CREATORUSERNUMBER
- CREATORSYSTEMNAME
- CREATORCREATORNAME
- CREATORCREATORNUMBER
- CREATORSYSTEMNUMBER
- CREATORPROCESSNAME
- CREATORAUTHLOCNAME
- CREATORTERMINALNAME
- CREATORAUTHLOCNUMBER
- OBJECTNAME

`AUDIT-EXCLUDE-VALUE`

specifies a set of values (up to five) for the respective field names in the AUDIT-EXCLUDE-FIELD. Combination of field names and the values determine the exclusion of NonStop client audit events. The default value is NONE.

Note. The attributes, AUDIT-EXCLUDE-FIELD and AUDIT-EXCLUDE-VALUE, are supported only on systems running J06.03 and later J-series RVUs, H06.14 and later H-series RVUs, and G06.32 and later G-series RVUs. Also, both the attributes must be used in a single command line while filtering the audit records.

`AUDIT-OSS-FILTER`

indicates if user level attributes, AUDIT-USER-ACTION-PASS and AUDIT-USER-ACTION-FAIL, enable or disable OSS auditing. The AUDIT-OSS-FILTER attribute takes effect only if the Safeguard global configuration attribute AUDIT-CLIENT-OSS is enabled.

The default value is OFF.

Note. The attribute AUDIT-OSS-FILTER is supported only on systems running J06.04 and later J-series RVUs, H06.15 and later H-series RVUs, and G06.32 and later G-series RVUs.

`AUDIT-TACL-LOGOFF`

controls generation of audits for the TACL LOGOFF or TACL EXIT operations. When set to ON, audits for the TACL LOGOFF or TACL EXIT operations are generated based on the value of the AUDIT-AUTHENTICATE-PASS and AUDIT-AUTHENTICATE-FAIL attributes.

When set to OFF, audits for the TACL LOGOFF or TACL EXIT operations are generated based on the value of the AUDIT-CLIENT-GUARDIAN, AUDIT-PROCESS-ACCESS-PASS, and AUDIT-PROCESS-ACCESS-FAIL attributes.

The default value is OFF.

Note. The attribute AUDIT-TACL-LOGOFF is supported only on systems running J06.08 and later J-series RVUs, H06.19 and later H-series RVUs, and G06.32 and later G-series RVUs.

`DYNAMIC-PROC-UPDATE`

ON specifies that the process identity attributes (AUDIT-USER-ACTION-PASS, AUDIT-USERACTION-FAIL, primary group, supplementary group list, and group count) are updated dynamically when the audit and group attributes of the corresponding user are modified.

The default value is OFF.

Note. The attribute DYNAMIC-PROC-UPDATE is supported only on systems running J06.10 and later J-series RVUs and H06.21 and later H-series RVUs.

`CI-PROG [` *`prog-filename`* `]`

`prog-filename` defines the command interpreter started after user authentication at a Safeguard terminal if no command interpreter is defined for the user or the terminal. The initial value is $SYSTEM.SYSTEM.TACL. A null entry for this attribute sets the value to NONE. *`prog-filename`* must be a local file name.

If CI-PROG is set to NONE, and no value is specified for CI-PROG for the terminal or user, no command interpreter is started at the Safeguard terminal.

`CI-LIB [` *`lib-filename`* `]`

`lib-filename` defines the library file to be used with the CI-PROG command interpreter specified in the Safeguard configuration record. The default value is no library. A null entry for this attribute resets the value to the default value. *`lib-filename`* must be a local file name.

`CI-SWAP [ $`*`vol`*` [ `*`subvol-filename`*` ] ]`

`$`*`vol`*`[ `*`subvol-filename`*`]` defines the swap volume or file to be used with the CI-PROG command interpreter specified in the Safeguard configuration record. `$`*`vol`* must be a local volume name. The default value is *NONE*. A null entry for this attribute resets the value to the default value. If no swap volume is specified, the volume that contains the CI-PROG object file is used as the swap volume when CI-PROG is started.

`CI-CPU [ `*`n`*` | ANY]`

n defines the number of the CPU in which the CI-PROG command interpreter runs. The default value is any CPU. A null entry for this attribute resets the value to the default value.

`CI-PRI [ `*`n`*` ]`

n defines the priority at which the CI-PROG command interpreter runs. The initial value is 149. A null entry for this attribute sets the value to NONE, and CI-PROG is started with the same priority as the $ZSMP process.

`CI-PARAM-TEXT [ `*`text`*` ]`

`text` specifies startup parameter text supplied to the CI-PROG command interpreter specified in the Safeguard configuration record. The default value is no startup parameter text. If you specify the CI-PARAM-TEXT attribute, it must be the last attribute in the command string. A null entry for this attribute resets the value to the default value.

`CMON { ON | OFF }`

defines whether the Safeguard software is to communicate with the $CMON process during the following events: logon, illegal logon attempts, logoff, and newprocess of the command interpreter. The initial value is OFF. (The Safeguard software does not communicate during these events.)

`CMONTIMEOUT [ `*`n`*` SECONDS ] ]`

n defines the number of seconds that the Safeguard software is to wait for any $CMON operation. The default value is 30 seconds. A null entry for this attribute resets the value to the default value.

`CMONERROR [ ACCEPT | DENY ]`

defines whether failures to communicate with $CMON should be ignored or should result in the authentication being denied. ACCEPT means to ignore these failures. DENY means to deny the authentication. The default is ACCEPT. A null entry for this attribute resets the value to the default value.

`BLINDLOGON { ON | OFF }`

defines whether passwords are accepted if they are typed on the same line as the user name during logon. ON specifies that passwords are not accepted if they are typed on the same line as the user name and that they must be entered on a separate line following the password prompt. OFF specifies that passwords can be entered on the same line as the user name during logon. The initial value is ON.

`NAMELOGON { ON | OFF }`

defines whether the Safeguard software accepts only a user name (*group name.member name*) during logon. OFF means that the Safeguard software accepts either a user name or a user ID (*group number,member number*). The initial value is ON.

`TERMINAL-EXCLUSIVE-ACCESS { ON | OFF }`

defines whether a user logged on at a Safeguard terminal has exclusive access to that terminal until the user logs off. ON means that the terminal is exclusively reserved for the user currently logged on. No other user can have access to the terminal during the authenticated user’s session. OFF means that exclusive access is not guaranteed to the user who is logged on. The initial value is OFF.

`SYSTEM-WARNING-MODE { ON | OFF }`

defines whether warning mode for individual objects is to be enabled. (For more information about warning mode, see the *Safeguard Administrator’s Manual*.) ON enables warning mode for individual objects. The initial value is OFF.

`WARNING-FALLBACK-SECURITY { GUARDIAN | GRANT }`

defines whether Guardian security settings are to be enforced when warning mode is enabled and the ACL evaluation denies access. GUARDIAN specifies that the rules are to be enforced. GRANT specifies that the rules are not to be enforced. The initial value is GUARDIAN.

`ALLOW-DISKFILE-PERSISTENT { NORMAL | ALWAYS }`

controls creation of persistent protection records for disk files with the ADD command. The NORMAL value of this attribute is designed to preserve backward compatibility. The ALWAYS value provides access to the persistence feature.

`ALWAYS`

allows the creation of persistent protection records for files that might not exist. Use of this value enforces some restrictions on the ADD DISKFILE *fn*, … PERSISTENT ON command in cases where the disk file does not exist.

- The OWNER attribute must be specified.
- Only these users are allowed to create PERSISTENT protection records for disk files that do not exist:
    - Users that have CREATE authority for OBJECTTYPE DISKFILEs
    - The manager (*,255) of the group of the specified owner
    - The super ID (255,255)

`NORMAL`

restricts creation of disk-file protection records to files that exist. NORMAL is the initial, default value.

`OBJECT-WARNING-MODE { ON | OFF }`

defines whether warning mode is enabled for the specified object. The value is required. For more information on warning mode, see the *Safeguard Administrator's Manual*.

ON enables warning mode for the specified object. The initial value is OFF, which disables warning mode for the specified object.

`ALLOW-NODE-ID-ACL { ON | OFF }`

defines whether ACL entries containing explicit node identifiers for subjects are consulted to determine remote access. The initial value is OFF, ignoring ACL entries containing explicit node identifiers.

`CHECK-DISKFILE-PATTERN { OFF | ONLY | FIRST | LAST | MID }`

defines diskfile-patterns operations.

`OFF`

specifies that no pattern searches will occur.

`FIRST`

specifies that pattern searching will occur first, and if and only if the result is NORECORD then the normal search for a protection record will occur.

`LAST`

specifies that pattern searching will occur after the normal search if and only if the normal search result is NORECORD.

`ONLY`

specifies that only pattern searching will occur. That is, normal non-pattern searching will not be performed even if the pattern search returns NORECORD.

`MID`

specifies that pattern based protection records will be searched:

- After the diskfile protection record search returns NORECORD when Direction-Diskfile is set to Filename-First.
- Before the diskfile protection record search, when the Direction-Diskfile is set to VOLUME-FIRST, and the VOLUME and SUBVOLUME protection record search returns NORECORD.

Note. The MID option is supported only on systems running J06.08 and later J-series RVUs and H06.19 and later H-series RVUs.

`AUDIT-CLIENT-OSS { ON | OFF }`

specifies whether the OSS-related audits are written to the Safeguard audit trial. It allows the auditing of OSS-related operations to be configured independently of the existing AUDIT-CLIENT-GUARDIAN attribute, and all other subsystem client auditing. The initial value is ON.

Note. The AUDIT-CLIENT-OSS attribute is supported only on systems running H06.08 and later H-series RVUs and G06.29 and later G-series RVUs.

`PASSWORD-ALGORITHM { DES | HMAC256 }`

indicates the algorithm to encrypt passwords when they are changed. The initial value is DES.

Note. The PASSWORD-ALGORITHM attribute is supported only on systems running H06.06 and later H-series RVUs and G06.29 and later G-series RVUs.

`DES`

indicates to use the DES algorithm to encrypt passwords. This is the initial value. Encrypted passwords are stored in the L/USERID and L/USERAX files.

`HMAC256`

indicates to use the HMAC with SHA-256 algorithm to encrypt passwords, when PASSWORD-ENCRYPT is ON. Encrypted passwords are stored in the L/USERAX files.

`PASSWORD-MAXIMUM-LENGTH {n}`

specifies the maximum acceptable length of a password. The initial value is 8 and the maximum value is 8 for DES algorithm and 64 for HMAC256 algorithm.

Note. This attribute is supported only on systems running H06.08 and later H-series RVUs and G06.31 and later G-series RVUs.

`PASSWORD-COMPATIBILITY-MODE {ON | OFF}`

specifies that only first eight characters of the password will be considered during password change. This attribute can take effect only when PASSWORD-ENCRYPT is ON, and PASSWORD-ALGORITHM is HMAC256. The initial value is ON.

Note. This attribute is supported only on systems running H06.08 and later H-series RVUs and G06.31 and later G-series RVUs.

`PASSWORD-ERROR-DETAIL {ON | OFF}`

determines whether a detailed error message is displayed to the user when the password supplied does not meet the minimum complexity as defined.

Detailed error message is displayed when PASSWORD-ERROR-DETAIL is ON as per the minimum required complexity for the password. A default error message is displayed when PASSWORD-ERROR-DETAIL is OFF.

However, the default value is OFF. This attribute defines part of the SAFEGUARD global configuration.

For an example of the detailed error message, see the *Safeguard User's Guide*.

Note. This attribute is supported only on systems running J06.14 and later J-series RVUs and H06.25 and later H-series RVUs.

`PROMPT-BEFORE-STOP {ON | OFF}`

determines whether a confirmation message is displayed to the user when the STOP command is issued at the SAFECOM prompt.

If PROMPT-BEFORE-STOP is set to ON, following confirmation message is displayed:

```
Are you sure you want to stop Safeguard? Y[ES] / N[O] :
```

You can choose to stop Safeguard by entering Y or YES, or continue by entering N or NO.

The default value for PROMPT-BEFORE-STOP attribute is OFF. This attribute is part of the SAFEGUARD global configuration.

Note. This attribute is supported only on systems running J06.16 and later J-series RVUs, and H06.27 and later H-series RVUs.

`PASSWORD-UPPERCASE-REQUIRED { ON | OFF }`

defines whether the user password will be enforced to have at least one uppercase character. The initial value is OFF.

The PASSWORD-UPPERCASE-REQUIRED attribute can be set to ON when PASSWORD-ALGORITHM is HMAC256 and PASSWORD-ENCRYPT is ON.

Note.

- On systems running J06.11 and later J-series RVUs and H06.22 and later H-series RVUs, the PASSWORD-UPPERCASE-REQUIRED attribute supports the DES and HMAC256 password algorithms. Therefore, the PASSWORD-UPPERCASE-REQUIRED attribute can be set to ON when PASSWORD-ENCRYPT is ON.
- The PASSWORD-UPPERCASE-REQUIRED attribute will take effect only when the PASSWORD-MIN-QUALITY-REQUIRED attribute name is set to value greater than 0.
- The PASSWORD-UPPERCASE-REQUIRED attribute is supported only on systems running H06.09 and later H-series RVUs and G06.31 and later G-series RVUs.

`PASSWORD-LOWERCASE-REQUIRED { ON | OFF }`

defines whether the user password will be enforced to have at least one lowercase character. The initial value is OFF.

The PASSWORD-LOWERCASE-REQUIRED attribute can be set to ON when PASSWORD-ALGORITHM is HMAC256 and PASSWORD-ENCRYPT is ON.

Note.

- On systems running J06.11 and later J-series RVUs and H06.22 and later H-series RVUs, the PASSWORD-LOWERCASE-REQUIRED attribute supports the DES and HMAC256 password algorithms. Therefore, the PASSWORD-LOWERCASE-REQUIRED attribute can be set to ON when PASSWORD-ENCRYPT is ON.
- The PASSWORD-LOWERCASE-REQUIRED attribute will take effect only when the PASSWORD-MIN-QUALITY-REQUIRED attribute name is set to value greater than 0.
- The PASSWORD-LOWERCASE-REQUIRED attribute is supported only on systems running H06.09 and later H-series RVUs and G06.31 and later G-series RVUs.

`PASSWORD-NUMERIC-REQUIRED {ON / OFF}`

defines whether the user password will be enforced to have at least one numeric character. The initial value is OFF.

The PASSWORD-NUMERIC-REQUIRED attribute can be set to ON when PASSWORD-ALGORITHM is HMAC256 and PASSWORD-ENCRYPT is ON.

Note.

- On systems running J06.11 and later J-series RVUs and H06.22 and later H-series RVUs, the PASSWORD-NUMERIC-REQUIRED attribute supports the DES and HMAC256 password algorithms. Therefore, the PASSWORD-NUMERIC-REQUIRED attribute can be set to ON when PASSWORD-ENCRYPT is ON.
- The PASSWORD-NUMERIC-REQUIRED attribute will take effect only when the PASSWORD-MIN-QUALITY-REQUIRED attribute is set to value greater than 0.
- The PASSWORD-NUMERIC-REQUIRED attribute is supported only on systems running H06.09 and later H-series RVUs and G06.31 and later G-series RVUs.

`PASSWORD-SPECIALCHAR-REQUIRED {ON / OFF}`

defines whether the user password will be enforced to have at least one special character. The initial value is OFF.

The PASSWORD-SPECIALCHAR-REQUIRED attribute can be set to ON when PASSWORD-ALGORITHM is HMAC256 and PASSWORD-ENCRYPT is ON.

Note.

- On systems running J06.11 and later J-series RVUs and H06.22 and later H-series RVUs, the PASSWORD-SPECIALCHAR-REQUIRED attribute supports the DES and HMAC256 password algorithms. Therefore, the PASSWORD-SPECIALCHAR-REQUIRED attribute can be set to ON when PASSWORD-ENCRYPT is ON.
- The PASSWORD-SPECIALCHAR-REQUIRED attribute will take effect only when the PASSWORD-MIN-QUALITY-REQUIRED attribute is set to value greater than 0.
- The PASSWORD-SPECIALCHAR-REQUIRED attribute is supported only on systems running H06.09 and later H-series RVUS and G06.31 and later G-series RVUs.

`PASSWORD-SPACES-ALLOWED {ON / OFF}`

defines whether a user password will be allowed to have embedded spaces. The initial value is OFF.

Note. This attribute is supported only on systems running H06.09 and later H-series RVUs and G06.31 and later G-series RVUs.

When PASSWORD-ENCRYPT is OFF or PASSWORD-ALGORITHM is DES or PASSWORD-COMPATIBILITY-MODE is ON, an attempt to alter PASSWORD-SPACES-ALLOWED to ON shall result in an error. The error messages displayed are:

```
THIS ATTRIBUTE CANNOT BE MODIFIED UNLESS PASSWORD-ALGORITHM = HMAC256,
PASSWORD-ENCRYPT = ON, and PASSWORD-COMPATIBILITY-MODE = OFF; COMMAND NOT
EXECUTED.
```

```
ZSFG^ERR^PSWD^SPACE^NEED^CMOFF
```

`PASSWORD-MIN-QUALITY-REQUIRED`

specifies the minimum quality criteria that must be met when a password is set or changed. The valid values of PASSWORD-MIN-QUALITY-REQUIRED range from 0 to 5. The initial value is 0.

Note. When any one of the following password quality attributes is enabled,

PASSWORD-MIN-QUALITY-REQUIRED will be automatically set from 0 to 1:

- PASSWORD-UPPERCASE-REQUIRED
- PASSWORD-LOWERCASE-REQUIRED
- PASSWORD-NUMERIC-REQUIRED
- PASSWORD-SPECIALCHAR-REQUIRED
- PASSWORD-ALPHA-REQUIRED

.

Note. This attribute is supported only on systems running H06.09 and later H-series RVUs and G06.31 and later G-series RVUs.

The following considerations apply to the PASSWORD-MIN-QUALITY-REQUIRED attribute:

- PASSWORD-MIN-QUALITY-REQUIRED can be modified only when PASSWORD-ENCRYPT is ON.

- When PASSWORD-ENCRYPT is OFF, an attempt to alter the quality attributes results in an error. The error messages displayed are:

```
THIS ATTRIBUTE CANNOT BE MODIFIED UNLESS PASSWORD-ENCRYPT = ON; COMMAND
NOT EXECUTED.
```

```
ZSFG^ERR^PSWD^QUAL^NEED^ENC
```

- PASSWORD-MIN-QUALITY-REQUIRED set to a value greater than 0, indicates that the PASSWORD-UPPERCASE-REQUIRED, PASSWORD-LOWERCASE-REQUIRED, PASSWORD-NUMERIC-REQUIRED, and PASSWORD-SPECIALCHAR-REQUIRED attributes, if enabled, meet the password quality criteria.
- When PASSWORD-MIN-QUALITY-REQUIRED is set to a value greater than 0, and if PASSWORD-ENCRYPT is changed from ON to OFF, the PASSWORD-MIN-QUALITY-REQUIRED attribute is reset to 0.

`PASSWORD-MIN-UPPERCASE-REQ [` *n* `]`

n specifies the minimum number of uppercase characters required in a user password when it is set or changed.

The valid values of PASSWORD-MIN-UPPERCASE-REQ range from 0 to 8. The initial value is 0.

Note. The PASSWORD-MIN-UPPERCASE-REQ attribute is supported only on systems running J06.11 and later J-series RVUs and H06.22 and later H-series RVUs.

The following considerations apply to the PASSWORD-UPPERCASE-REQ attribute:

- The PASSWORD-MIN-UPPERCASE-REQ attribute will take effect only when the PASSWORD-UPPERCASE-REQUIRED attribute is enabled.
- When the PASSWORD-UPPERCASE-REQUIRED attribute is changed from OFF to ON, Safeguard sets the value of the PASSWORD-MIN-UPPERCASE-REQ attribute to 1.
- When the PASSWORD-UPPERCASE-REQUIRED attribute is changed from ON to OFF, Safeguard sets the value of the PASSWORD-MIN-UPPERCASE-REQ attribute to 0.
- The sum of the values of the effective password quality attributes (PASSWORD-MIN-UPPERCASE-REQ, PASSWORD-MIN-LOWERCASE-REQ, PASSWORD-MIN-NUMERIC-REQ, PASSWORD-MIN-SPECIALCHAR-REQ or PASSWORD-MIN-ALPHA-REQ) must not be greater than the value of the PASSWORD-MAXIMUM-LEN attribute.

`PASSWORD-MIN-LOWERCASE-REQ [ *n* ]`

n specifies the minimum number of lowercase characters required in a user password when it is set or changed.

The valid values of PASSWORD-MIN-LOWERCASE-REQ range from 0 to 8. The initial value is 0.

Note. The PASSWORD-MIN-LOWERCASE-REQ attribute is supported only on systems running J06.11 and later J-series RVUs and H06.22 and later H-series RVUs.

The following considerations apply to the PASSWORD-MIN-LOWERCASE-REQ attribute:

- The PASSWORD-MIN-LOWERCASE-REQ attribute will take effect only when the PASSWORD-LOWERCASE-REQUIRED attribute is enabled.
- When the PASSWORD-LOWERCASE-REQUIRED attribute is changed from OFF to ON, Safeguard sets the value of the PASSWORD-MIN-LOWERCASE-REQ attribute to 1.
- When the PASSWORD-LOWERCASE-REQUIRED attribute is changed from ON to OFF, Safeguard sets the value of the PASSWORD-MIN-LOWERCASE-REQ attribute to 0.
- The sum of the values of the effective password quality attributes (PASSWORD-MIN-UPPERCASE-REQ, PASSWORD-MIN-LOWERCASE-REQ, PASSWORD-MIN-NUMERIC-REQ, PASSWORD-MIN-SPECIALCHAR-REQ or PASSWORD-MIN-ALPHA-REQ) must not be greater than the value of the PASSWORD-MAXIMUM-LEN attribute.

`PASSWORD-MIN-NUMERIC-REQ [ *n* ]`

n specifies the minimum number of numeric characters required in a user password when it is set or changed.

The valid values of PASSWORD-MIN-NUMERIC-REQ range from 0 to 8. The initial value is 0.

Note. The PASSWORD-MIN-NUMERIC-REQ attribute is supported only on systems running J06.11 and later J-series RVUs and H06.22 and later H-series RVUs**.**

The following considerations apply to the PASSWORD-MIN-NUMERIC-REQ attribute:

- The PASSWORD-MIN-NUMERIC-REQ attribute will take effect only when the PASSWORD-NUMERIC-REQUIRED attribute is enabled.
- When the PASSWORD-NUMERIC-REQUIRED attribute is changed from OFF to ON, Safeguard sets the value of the PASSWORD-MIN-NUMERIC-REQ attribute to 1.

- When the PASSWORD-NUMERIC-REQUIRED attribute is changed from ON to OFF, Safeguard sets the value of the PASSWORD-MIN-NUMERIC-REQ attribute to 0.
- The sum of the values of the effective password quality attributes (PASSWORD-MIN-UPPERCASE-REQ, PASSWORD-MIN-LOWERCASE-REQ, PASSWORD-MIN-NUMERIC-REQ, PASSWORD-MIN-SPECIALCHAR-REQ or PASSWORD-MIN-ALPHA-REQ) must not be greater than the value of the PASSWORD-MAXIMUM-LEN attribute.

`PASSWORD-MIN-SPECIALCHAR-REQ [ n ]`

specifies the minimum number of special characters required in a user password when it is set or changed.

The valid values of PASSWORD-MIN-SPECIALCHAR-REQ range from 0 to 8. The initial value is 0.

Note. The PASSWORD-MIN-SPECIALCHAR-REQ attribute is supported only on systems running J06.11 and later J-series RVUs and H06.22 and later H-series RVUs. **.**

The following considerations apply to the PASSWORD-MIN-SPECIALCHAR-REQ attribute:

- The PASSWORD-MIN-SPECIALCHAR-REQ attribute will take effect only when the PASSWORD-NUMERIC-REQUIRED attribute is enabled.
- When the PASSWORD-SPECIALCHAR-REQUIRED attribute is changed from OFF to ON, Safeguard sets the value of the PASSWORD-MIN-NUMERIC-REQ attribute to 1.
- When the PASSWORD-SPECIALCHAR-REQUIRED attribute is changed from ON to OFF, Safeguard sets the value of the PASSWORD-MIN-SPECIALCHAR-REQ attribute to 0.
- The sum of the values of the effective password quality attributes (PASSWORD-MIN-UPPERCASE-REQ, PASSWORD-MIN-LOWERCASE-REQ, PASSWORD-MIN-NUMERIC-REQ, PASSWORD-MIN-SPECIALCHAR-REQ or PASSWORD-MIN-ALPHA-REQ) must not be greater than the value of the PASSWORD-MAXIMUM-LEN attribute.

`PASSWORD-ALPHA-REQUIRED {ON / OFF}`

specifies whether the user password will be enforced to have at least one alphabetical character. The initial value is OFF.

The PASSWORD-ALPHA-REQUIRED attribute can be set to ON when PASSWORD-ENCRYPT is ON.

Note.

- The PASSWORD-ALPHA-REQUIRED attribute will take effect only when the PASSWORD-MIN-QUALITY-REQUIRED attribute is set to value greater than 0.
- The PASSWORD-ALPHA-REQUIRED attribute is supported only on systems running J06.11 and later J-series RVUs and H06.22 and later H-series RVUs.

`PASSWORD-MIN-ALPHA-REQ [` *`n`* `]`

specifies the minimum number of alphabetical characters required in a user password when it is set or changed.

The valid values of PASSWORD-MIN-ALPHA-REQ range from 0 to 8. The initial value is 0.

Note. The PASSWORD-MIN-ALPHA-REQ attribute is supported only on systems running J06.11 and later J-series RVUs and H06.22 and later H-series RVUs. **.**

The following considerations apply to the PASSWORD-MIN-ALPHA-REQ attribute:

- The PASSWORD-MIN-ALPHA-REQ attribute will take effect only when the PASSWORD-ALPHA-REQUIRED attribute is enabled.
- When the PASSWORD-ALPHA-REQUIRED attribute is changed from OFF to ON, Safeguard sets the value of the PASSWORD-MIN-ALPHA-REQUIRED attribute to 1.
- When the PASSWORD-ALPHA-REQUIRED attribute is changed from ON to OFF, Safeguard sets the value of the PASSWORD-MIN-ALPHA-REQ attribute to 0.
- The sum of the values of the effective password quality attributes (PASSWORD-MIN-UPPERCASE-REQ, PASSWORD-MIN-LOWERCASE-REQ, PASSWORD-MIN-NUMERIC-REQ, PASSWORD-MIN-SPECIALCHAR-REQ or PASSWORD-MIN-ALPHA-REQ) must not be greater than the value of the PASSWORD-MAXIMUM-LEN attribute.

For more diskfile-pattern information, see the *Safeguard User’s Guide*.

For examples of how to use these parameters, see the *Safeguard Administrator’s Manual*.

17

Running Other Programs From SAFECOM

You can execute the RUN command directly from SAFECOM. This feature allows a security administrator to run programs without having to leave SAFECOM.

The SAFECOM RUN command is a modified form of the TACL RUN command. It differs from the TACL RUN command in these ways:

- An implicit RUN command is not supported.
- The RUND command is not supported.
- Several run options are not supported.

The SAFECOM RUN command does not interact with a $CMON process, nor does it pass ASSIGNs, PARAMs, or DEFINEs. These actions are not acceptable in a security subsystem because they allow substitutions for file names.

Run Command

For more information on the RUN command, see the *TACL Reference Manual*. The syntax of the SAFECOM RUN command is:

```
RUN  program-file [ [ / run-option [ , run-option ] ... / ]
                    [ param-set ] ]
```

program-file

is the name of the file containing the object program to be run.

run-option

is any of the following run options, which are described in the *TACL Reference Manual*:

CPU *cpu-number*

INSPECT { OFF | ON | SAVEABEND }

IN [*file-name*]

LIB [*file-name*]

MEM *num-pages*

NAME [$*process-name*]

NOWAIT

OUT [*list-file*]

PRI *priority*

TERM [\system-name.]$*terminal-name*

param-set

is a program parameter or series of parameters sent to the new process in the startup message.

Consideration

If the RUN command is used in a command line that contains multiple commands (separated by semicolons), it must be the last command in that line.

Example

This RUN command runs the program MYEDIT, which resides in the current default subvolume. The program uses the library file EDITLIB.

```
=RUN myedit / LIB editlib /
```

A

SAFECOM Error and Warning Messages

If SAFECOM encounters a condition that prohibits it from successfully executing a command, SAFECOM displays an error or warning message. The error or warning message gives a brief description of the condition that prohibited SAFECOM from executing the command.

This appendix describes the SAFECOM error and warning messages. The messages are listed in alphabetical order. For each message, the cause of the error, the action taken by SAFECOM when the error is encountered, and the steps you can take to recover from the error are described.

In addition to the messages described in this appendix, the Safeguard software also issues operator messages. For more information, see the *Operator Messages Manual*.

For some operations, SAFECOM indicates the nature of the problem with either of these messages:

FILE ERROR = *nnn* indicates that either SAFECOM or the SMP encountered a file-system error attempting to perform some operation. (Typically, this error condition results from an attempt by the SMP to access either the subject or object database.)

SMP ERROR = *nnnn* indicates that the SMP encountered an internal error attempting to perform the requested operation.

Report file-system errors to your system manager. The problem might be the result of some improper system management procedure.

SMP errors indicate a problem with the Safeguard product. Report them to your HP representative.

If you report an error to your HP representative, be prepared to provide the following information to aid in diagnosing the problem:

1. Provide the full version procedures of OSMON, OSMP, and SAFECOM. To obtain this information, use the VPROC utility:

   ```
   > VPROC $SYSTEM.SYSnn.SAFECOM
   > VPROC $SYSTEM.SYSnn.OSMON
   > VPROC $SYSTEM.SYSnn.OSMP
   ```

   Also provide the full version procedures of other products, such as TACL or FUP, that interact with the Safeguard subsystem to produce this problem.

2. Provide the Safeguard configuration in effect when the error occurred. Also include information on a user who is experiencing the problem. To obtain this information, execute the following SAFECOM commands:

   ```
   > SAFECOM INFO SAFEGUARD, DETAIL
   > SAFECOM INFO USER user-ID, DETAIL
   ```

3. If the problem is reproducible, list in detail the steps required to reproduce the problem. If the problem is not reproducible, provide the EMSLOG that was active when the problem occurred.

4. Provide a synopsis of the series of SAFECOM commands that caused the problem.

5. If the problem started occurring only recently, list any recent changes made to the system.

The SAFECOM error and warning messages follow in alphabetical order.

```
ACL AUTHORITY NOT VALID FOR OBJECT TYPE; COMMAND NOT EXECUTED.
```

Cause. An access authority (READ, WRITE, EXECUTE, PURGE, or CREATE) specified in an *access-spec* is not valid for the type of object being added or altered.

Effect. The command is not executed.

Recovery. Reenter the command with a valid access authority for the specified object type.

```
ATTRIBUTE IS NOT COMPATIBLE WITH OBJECT TYPE; COMMAND NOT EXECUTED.
```

Cause. An attribute was specified that is not compatible with the type of object named in the command.

Effect. The command is not executed.

Recovery. Reenter the command using only attributes compatible with the specified object type.

```
ATTRIBUTE IS NOT COMPATIBLE WITH SUBJECT TYPE; COMMAND NOT EXECUTED.
```

Cause. An attribute was specified that is not compatible with the subject type named in the command.

Effect. The command is not executed.

Recovery. Reenter the command using only attributes compatible with the subject type.

```
CPU OR SYSTEM UNAVAILABLE
```

Cause. The CPU option in a RUN command specified a CPU that is unavailable, or the program was to be run on a system that is unavailable.

Effect. The command is rejected.

Recovery. Specify a different CPU or system, or retry when the CPU or system is available.

```
DIFFERENT LIBRARY CURRENTLY IN USE
```

Cause. The LIB option in a RUN command specified a library other than the one the program is currently using.

Effect. The command is rejected.

Recovery. Change the LIB option and retry the command.

```
* ERROR *   Attempt to enable event-exit with no program file name.
```

Cause. An attempt to enable the event-exit process failed because no program file name was specified.

Effect. The command is rejected.

Recovery. Specify a program file name and then retry the command.

```
* ERROR *   Attempt to remove program file name with event-exit enabled.
```

Cause. An attempt to remove the program file name failed because the event-exit process was enabled.

Effect. The command is rejected.

Recovery. Set ENABLED OFF and then remove the file name.

```
* ERROR *   Audit file already released
```

Cause. An attempt to execute a RELEASE command failed because the audit file is already released.

Effect. The command is rejected.

Recovery. None.

```
* ERROR *   Audit file does not exist
```

Cause. An attempt to execute a RELEASE command failed because the audit file does not exist.

Effect. The command is rejected.

Recovery. None.

```
* ERROR *   Audit file in use - unable to release
```

Cause. An attempt to release an audit file failed because the specified file is the current audit file.

Effect. The command is rejected.

Recovery. None.

```
* ERROR *   Audit file is foreign - unable to release
```

Cause. An attempt to execute a RELEASE command failed because the specified file is not an audit file.

Effect. The command is rejected.

Recovery. None.

```
* ERROR *   Audit Pool does not exist
```

Cause. An attempt to execute an audit service command failed because the audit pool does not exist.

Effect. The command is rejected.

Recovery. Specify another audit pool and retry the command.

```
* ERROR *   Audit Pool has data - RELEASE files suggested
```

Cause. An attempt to execute a DELETE AUDIT POOL command failed because the audit pool contains unreleased files.

Effect. The command is rejected.

Recovery. Release all files in the audit pool and then retry the command.

```
* ERROR *   Audit Pool is defined as CURRENT
```

Cause. An attempt to execute a DELETE AUDIT POOL command failed because the audit pool is the current audit pool.

Effect. The command is rejected.

Recovery. Select a different audit pool to be the current pool and then retry the command.

```
* ERROR *   Audit Pool is defined as NEXT
```

Cause. An attempt to execute a DELETE AUDIT POOL command failed because the audit pool is the next audit pool.

Effect. The command is rejected.

Recovery. Select a different audit pool to be the next audit pool and then retry the command.

```
* ERROR *   Audit Pool is full - RELEASE files suggested
```

Cause. An attempt to execute a NEXTFILE command failed because no files are available in the current audit pool.

Effect. The command is rejected.

Recovery. Release files in the current audit pool, increase the MAXFILES limit in the audit pool, or select a different audit pool.

```
** ERROR ** CANNOT ACCESS LIKE OBJECT: COMMAND TERMINATED.
```

Cause. SAFECOM cannot read the authentication record for the user specified with the LIKE attribute, or SAFECOM cannot read the authorization record for the object specified with the LIKE attribute.

Effect. The command is not executed.

Recovery. Typically, this error occurs because you specified a user for whom no authentication record exists, or you specified an object for which no authorization record exists. If this is the case, reenter the command with a valid user or object specification.

```
* ERROR * CANNOT ADD SECURITY-GROUP SECURITY-AUDITOR: ALREADY EXISTS
```

Cause. You attempted to add the SECURITY-AUDITOR security group when it already exists.

Effect. The command is not executed.

Recovery. None.

```
* ERROR *   CANNOT ADD SECURITY-GROUP SECURITY-AUDITOR: SECURITY VIOLATION
```

Cause. A user with insufficient privileges attemped to add the SECURITY-AUDITOR security group.

Effect. The command is not executed.

Recovery. None.

```
* ERROR * CANNOT ADD SECURITY-GROUP SECURITY-PRV-ADMINISTRATOR: ALREADY
EXISTS
```

Cause. You attempted to add the SECURITY-PRV-ADMINISTRATOR security group when it already exists.

Effect. The command is not executed.

Recovery. None.

```
* ERROR *   RECORD FOR OBJECTTYPE SECURITY-PRV-ADMINISTRATOR: SECURITY
VIOLATION
```

Cause. A user with insufficient privileges attempted to add the SECURITY-PRV-ADMINISTRATOR security group.

Effect. The command is not executed.

Recovery. None.

```
* ERROR *   CANNOT ADD SECURITY-GROUP SECURITY-MEDIA-ADMIN: ALREADY EXISTS
```

Cause. You attempted to add the SECURITY-MEDIA-ADMIN security group when it already exists.

Effect. The command is not executed.

Recovery. None.

```
* ERROR *  CANNOT ADD SECURITY-GROUP SECURITY-MEDIA-ADMIN: SECURITY
VIOLATION
```

Cause. A user with insufficient privileges attempted to add the SECURITY-MEDIA-ADMIN security group.

Effect. The command is not executed.

Recovery. None.

```
* ERROR * CANNOT ADD SECURITY-GROUP SECURITY-PERSISTENCE-ADMIN: ALREADY
EXISTS
```

Cause. A user attempts to add the SECURITY-PERSISTENCE-ADMIN security group when it already exists.

Effect. The command does not execute.

Recovery. None.

```
* ERROR * CANNOT ADD SECURITY-GROUP SECURITY-PERSISTENCE-ADMIN: SECURITY
VIOLATION
```

Cause. A user with insufficient privileges attempts to add the SECURITY-PERSISTENCE-ADMIN security group.

Effect. The command does not execute.

Recovery. None.

```
* ERROR * CANNOT ADD DISKFILE filename: OWNER MUST BE SPECIFIED FOR
NON-EXISTENT DISKFILES
```

Cause. You attempted to add a persistent protection record for a disk file that does not exist, specify the OWNER attribute.

Effect. The command is not executed.

Recovery. Specify OWNER [*owner-id*] and retry the command.

```
* ERROR *   CANNOT ADD DISKFILE filename : SUBVOLUME RESERVED FOR OSS
```

Cause. You attempted to add a protection record for a disk file that resides on a subvolume reserved for OSS. Safeguard cannot protect OSS disk files.

Effect. The command is not executed.

Recovery. Specify a disk file on another subvolume and retry the command.

```
* ERROR *   CANNOT ADD DISKFILE filename : VOLUME NOT FOUND
```

Cause. An attempt to add an authorization record for a disk file failed because the specified disk volume does not exist.

Effect. An authorization record for the disk file is not added to the object database.

Recovery. Wait for the volume to become available and then retry the command. Or specify another volume and retry the command.

```
* ERROR *   CANNOT ADD SUBVOLUME subvol-name : SUBVOLUME RESERVED FOR OSS
```

Cause. You attempted to add a protection record for a subvolume that is reserved for OSS. Safeguard cannot protect OSS subvolumes.

Effect. The command is not executed.

Recovery. Specify another subvolume and retry the command.

```
* ERROR *   CANNOT ADD objtype objname : FILE ERROR = ###
```

Cause. The SMP encountered the indicated file-system error while attempting to add an object-authorization record to the object database.

Effect. An authorization record for the object is not added to the object database.

Recovery. Report the problem to your system manager. Your system manager should ensure that the Safeguard software has been installed properly. If your system manager cannot solve the problem, report the error to your HP representative. (The *System Messages Manual describes* file-system errors.)

```
* ERROR *   CANNOT ADD objtype objname : SECURITY VIOLATION
```

Cause. You lack the authority required to add an authorization record for the indicated object. (Only the owner of a disk file, the owner's group manager, or the local super ID can add an authorization record for a disk file. Only a member of the local super group can add an authorization record for a device or disk volume.)

Effect. An authorization record for the object is not added to the object database.

Recovery. If the object is a disk file, ask the owner of the disk file to give you ownership of the disk file (with ALTER DISKFILE ... OWNER) and then reenter the ADD command. Alternatively, ask the owner of the disk file to add an authorization record for the disk file. If the object is a device or disk volume, ask a member of the super group to add an authorization record for the object.

```
* ERROR *   CANNOT ADD objtyp objname : SMP ERROR = ####
```

Cause. The SMP encountered an internal error while attempting to add an object-authorization record to the object database.

Effect. An authorization record for the object is not added to the object database.

Recovery. Write down the SMP error number and contact your HP representative. (Your HP representative will need the SMP error number to correct the problem.)

```
* ERROR *   CANNOT OPEN $ZSMP :  FILE ERROR = ###
```

Cause. SAFECOM encountered the indicated file-system error while attempting to open the SMP.

Effect. The command is not executed.

Recovery. Report the problem to your system manager. Your system manager should ensure that the Safeguard software has been installed properly. If your system manager cannot solve the problem, report the error to your HP representative. (The *System Messages Manual* describes file-system errors.)

```
* ERROR *   CANNOT OPEN $ZSMP :  NO SUCH PROCESS
```

Cause. The Safeguard management process ($ZSMP) is not running or has not been started on the system.

Effect. The command is not executed.

Recovery. Report the problem to your system manager. Your system manager should ensure that the Safeguard software has been installed properly.

```
* ERROR *   CANNOT OPEN $ZSMP :  SECURITY VIOLATION
```

Cause. You lack the authority required to access the SMP. For example, your command requested an operation on a remote object, and you have not been granted access to the remote object's system. You can also get this warning if your system manager has created an ACL for the SMP process name, and the ACL does not grant you READ or WRITE access authority to the SMP process.

Effect. The command is not executed.

Recovery. Ask your system manager to grant you the authority you need to access the SMP.

```
* ERROR *   CANNOT OPEN $ZSMP: $smp-name IN USE
```

Cause. The command failed because the SMP has already been opened by the maximum number of SAFECOM processes it allows. (Too many people are using SAFECOM at the same time.)

Effect. The command is not executed.

Recovery. Report the problem to your system manager. Wait a while and try to use SAFECOM again.

```
* ERROR *   CANNOT OPEN $ZSMP: $smp-name NOT FOUND
```

Cause. The command failed for either of two reasons:

- No SMP is running on your system or on the system of a remote object that you tried to manage with a SAFECOM command.
- The SMP running on your system or on a remote system was not started with the correct process name, $ZSMP.

Effect. The command is not executed for users or objects managed by the SMP for which the open failed.

Recovery. Report the problem to your system manager. If it is a local problem, your system manager can create a SMP with the correct name. If the problem is on a remote system, your system manager can ask the system manager on that system to correct the problem.

```
* ERROR *   COMMAND FAILED : FILE ERROR = ###
```

Cause. The SMP encountered the indicated file-system error while attempting to execute a STOP command.

Effect. The command is not executed.

Recovery. Report the problem to your system manager. Your system manager should ensure that the Safeguard software has been installed properly. If your system manager cannot solve the problem, report the error to your HP representative. (The *System Messages Manual* describes file-system errors.)

```
* ERROR *   COMMAND FAILED : SECURITY VIOLATION
```

Cause. The command failed because you lack the authority required to perform the command.

Effect. The command is not executed.

Recovery. Ask someone with the required authority to execute the command.

```
* ERROR *   COMMAND FAILED : SMP ERROR = ####
```

Cause. The SMP encountered an internal error while attempting to execute the STOP command.

Effect. The command is not executed.

Recovery. Write down the SMP error number and contact your HP representative. (Your HP representative will need the SMP error number to correct the problem.)

```
* ERROR *   DISKFILE filename : SUBVOLUME RESERVED FOR OSS
```

Cause. You attempted an operation on a disk file that resides on a subvolume reserved for OSS.

Effect. The command is not executed.

Recovery. Specify a disk file on another subvolume and retry the command.

```
* ERROR *   DISKFILE filename : WHERE NOT MATCHED
```

Cause. No disk file names were found to match the criteria specified in a WHERE clause.

Effect. The command is not executed.

Recovery. Specify another name pattern or another WHERE criterion and then retry the command.

```
* ERROR *   Failure starting process; error (n,nn) from PROCESS_CREATE_
```

Cause. One of the attributes in the event-exit process configuration record is invalid. The value *n*,*nn* indicates a file-system or PROCESS_CREATE_ error.

Effect. The process cannot be started when ENABLED is set to ON.

Recovery. Correct the invalid attribute and then retry the command.

```
* ERROR *   File system error num from procedure name, file audit-file
```

Cause. An attempt to execute an audit service command was rejected because the specified file’s disk volume is down.

Effect. The command is rejected.

Recovery. Bring up the disk volume and retry the command.

```
* ERROR *   Max process entries 1 exceeded.
```

Cause. An attempt to was made to add another event-exit process when one already exists.

Effect. The command is rejected.

Recovery. Delete the existing event-exit process configuration record and then retry the command.

```
* ERROR *   Maximum number of Terminals (LUs) defined
```

Cause. An attempt was made to add another terminal definition with the ADD TERMINAL command, but the maximum number of terminals has already been defined. The current maximum is about 420 terminals.

Effect. The command is rejected.

Recovery. None.

```
* ERROR *   Name conflicts with existing alias name.
```

Cause. An ADD USER command specified a user name that matches an existing user alias. (In this check, the user alias is treated as case insensitive.)

Effect. The command is not executed.

Recovery. Reenter the command with a different user name or delete the conflicting alias name and retry the command.

```
* ERROR *   Name conflicts with existing user name.
```

Cause. You attempted to add a user alias that already exists as a user name.

Effect. The command is not executed.

Recovery. Reenter the command with a user alias that does not match an existing user name.

```
* ERROR *   No objects of type obj-type matching obj-name were found.
```

Cause. A command specified a nonexistent object name.

Effect. The command is rejected.

Recovery. Specify an existing object name and retry the command.

```
* ERROR *   Password in history
```

Cause. The password specified in the command has already been used (duplicates an entry in the user’s password history record) and the PASSWORD-HISTORY global configuration attribute is in effect.

Effect. The authentication record for the user is not modified.

Recovery. Reenter the command with a suitable password.

```
* ERROR *   Password length less than PASSWORD-MINIMUM-LENGTH n
```

Cause. The password specified in the command contained fewer characters than required by the PASSWORD-MINIMUM-LENGTH global configuration attribute.

Effect. The command is not executed.

Recovery. Reenter the command with a suitable password of an appropriate length.

```
* ERROR *   process enabled - ALTER ENABLED OFF prior to DELETE
```

Cause. An attempt was made to delete the event-exit process while it was enabled.

Effect. The command is rejected.

Recovery. Disable the event-exit process and then retry the command.

```
* ERROR *   RECORD FOR objtype objname : ALL OWNERS ARE DENIED
```

Cause. An attempt to manage an authorization record for the indicated object failed. If it had been allowed, there would no owner of the record. (No one could manage it.)

Effect. The authorization record for the object is not changed.

Recovery. Make sure at least one user will have owner authority, correct the command, and then retry it.

```
* ERROR *   RECORD FOR objtype objname : ALREADY EXISTS
```

Cause. An attempt to add an authorization record for the indicated object failed because an authorization record for the object already exists.

Effect. The authorization record for the object is not changed.

Recovery. None.

```
* ERROR *   RECORD FOR objtype objname : FILE ERROR = ###
```

Cause. The SMP encountered the indicated file-system error while attempting to access the authorization record for the indicated object.

Effect. The command is not executed for the object.

Recovery. Report the problem to your system manager. Your system manager should ensure that the Safeguard software has been installed properly. If your system manager cannot solve the problem, report the error to your HP representative. (The *System Messages Manual* describes file-system errors.)

```
* ERROR *   RECORD FOR objtype objname : LICENSE ONLY PROGRAM OBJECT FILES
```

Cause. An attempt to set the LICENSE attribute ON for a disk file failed because the file is not a program object file.

Effect. The command is rejected.

Recovery. None.

```
* ERROR *   RECORD FOR objtype objname : NOT FOUND
```

Cause. No authorization record exists in the object database for the indicated object.

Effect. The command is not executed for the object.

Recovery. Possibly you misspelled the object name. If so, reenter the command with the correct object name.

```
* ERROR *   RECORD FOR objtype objname : PROGID ONLY PROGRAM OBJECT FILES
```

Cause. An attempt to set the PROGID attribute ON for a disk file failed because the file is not a program object file.

Effect. The command is rejected.

Recovery. None.

```
* ERROR *   RECORD FOR objtype objname : SECURITY VIOLATION
```

Cause. You lack the authority required to perform the requested operation on the indicated object. Only the owner of an object, the owner’s group manager, or the local super ID can alter, freeze, thaw, or delete an object’s authorization record.

Effect. The command is not executed for the object.

Recovery. Ask the owner of the object to perform the operation.

```
* ERROR *   RECORD FOR objtype objname : SMP ERROR = ####
```

Cause. The SMP encountered an internal error while attempting to execute the command.

Effect. The command is not executed for the indicated object.

Recovery. Write down the SMP error number and contact your HP representative. (Your HP representative will need the SMP error number to correct the problem.)

```
* ERROR *   RECORD FOR objtype objname : SQL OBJECTS NOT SAFEGUARD
PROTECTED
```

Cause. An attempt to add a Safeguard protection record for a SQL object failed.

Effect. The command is rejected.

Recovery. None. SQL objects cannot be added to the Safeguard database.

```
* ERROR *   RECORD FOR objtype objname : TOO MANY ACL ENTRIES (> 50)
```

Cause. An attempt to alter or add an authorization record for the object failed because the resulting record contained too many ACL entries.

Effect. The authorization record for the object is not changed or added.

Recovery. If possible, eliminate some ACL entries.

```
* ERROR *   Requested primary group is not defined
```

Cause. An attempt was made to assign a user to a primary group that is not in the user’s group list.

Effect. The command is rejected.

Recovery. Verify that the specified group exists and add the user to the group, or choose a group in the user’s group list. Then retry the command.

```
* ERROR *   Security violation
```

Cause. The user is not authorized to execute this command.

Effect. The command is rejected.

Recovery. Log on with a user ID that is authorized to execute the command and then retry.

```
* ERROR *   Small Userid file
```

Cause. There are two possible causes: (1) The command contained some DEFAULT-PROTECTION attributes, which cannot fit within the record size defined for older, small-size USERID files. (2) The command specified the value of the PASSWORD-HISTORY configuration attribute as greater than 20, which exceeds the record size defined for older, small-size USERID files.

Effect. The command is not executed.

Recovery. For cause 1, either reenter the command without the DEFAULT-PROTECTION attributes, or convert the USERID file to the larger record size and then reenter the command. For cause 2, either reenter the command with a value of 20 or less, or convert the USERID file to the larger record size and then reenter the command.

```
* ERROR *   Subject Group Name invalid
```

Cause. The group name specified does not match the group name for users in the specified group number. For example, assume the ADMIN group is group number 1. This message would be issued when you try to add FINANCE.JOHN as the user associated with the 1,11 user ID.

Effect. An authentication record for the user is not added to the subject database.

Recovery. Reenter the command with the group name that matches the group number used (for example, ADMIN, in the preceding case).

```
* ERROR *   Subject is not a member of requested primary group.
```

Cause. The user does not belong to the group specified in a PRIMARY-GROUP option.

Effect. The command is not executed.

Recovery. Add the user to the group and then retry the command.

```
* ERROR *   SUBVOLUME subvol-name: SUBVOLUME RESERVED FOR OSS
```

Cause. You attempted an operation on a subvolume reserved for OSS.

Effect. The command is not executed.

Recovery. Specify another subvolume and retry the command.

```
* ERROR *   Terminal (LU) in use - FREEZE prior to DELETE
```

Cause. You attempted to delete a terminal that is not frozen.

Effect. The command is rejected.

Recovery. Use a FREEZE TERMINAL command to freeze the terminal and then retry the DELETE TERMINAL command

Recovery.

```
* ERROR *   The group group-name has members.
```

Cause. An attempt was made to delete a group that has members.

Effect. The command is not executed.

Recovery. Remove all members from the group and reenter the command.

```
* ERROR *   The group number group-number was not found.
```

Cause. The requested group number was not found.

Effect. The command is not executed.

Recovery. Specify the correct group number and reenter the command.

```
* ERROR *   The object obj-type obj-name is already defined
```

Cause. An ADD command specified an object name that already has a protection record.

Effect. The command is rejected.

Recovery. Correct the spelling of the object name or specify an object name that is not protected. Then retry the command.

```
* ERROR *   The requested group id group-number is already defined.
```

Cause. The requested group number already exists.

Effect. The ADD GROUP command is not executed.

Recovery. Specify a different group number and reenter the command.

```
* ERROR *   The requested group id group-id is reserved.
```

Cause. An attempt to add the group failed because the group number is reserved for future use by HP.

Effect. The command is not executed.

Recovery. Specify a number within the permissible range and reenter the command.

```
* ERROR *   The requested group id group-id is restricted.
```

Cause. An attempt to add a group failed because the group number corresponds to a restricted group and can have only fixed group names.

Effect. The command is not executed.

Recovery. Specify the correct group name or group numbers within the range, if you are adding a restricted group or a file sharing group respectively.

```
* ERROR *   The requested group name group-name is already defined.
```

Cause. A case-sensitive search found that the group name already exists.

Effect. The ADD GROUP command is not executed.

Recovery. Specify a different group name and reenter the command.

```
* ERROR *   The requested group name group-name is not defined.
```

Cause. You specified a group that does not exist.

Effect. The GROUP command is not executed.

Recovery. Specify the correct group name and reenter the command

```
* ERROR *   The requested group name group-name is restricted.
```

Cause. An attempt to add a group failed because the group name corresponds to a restricted group and can have only fixed group number.

Effect. The command is not executed.

Recovery. Specify a different group name if you are adding a file sharing group; or specify the correct group number if you are adding a restricted group; and reenter the command.

```
* ERROR *   The requested user ID user is already defined.
```

Cause. The specified user ID already exists.

Effect. The command is not executed.

Recovery. Specify a different user ID or specify a different command. Then retry the command.

```
* ERROR *   Token name has an invalid value
```

Cause. The named attribute has an invalid value.

Effect. The command is rejected.

Recovery. For the correct value, refer to the *Safeguard Management Programming Manual*. Correct the attribute value and then retry the command.

```
** ERROR ** UNABLE TO CONVERT TIMESTAMP: Ambiguous LCT
```

Cause. The Guardian procedure CONVERTTIMESTAMP failed with an error.

Effect. The command is not executed.

Recovery. Make appropriate corrections to the daylight savings time (DST) table and retry the command.

```
** ERROR ** UNABLE TO CONVERT TIMESTAMP: DST range error
```

Cause. The Guardian procedure CONVERTTIMESTAMP failed with an error.

Effect. The command is not executed.

Recovery. Make appropriate corrections to the daylight savings time (DST) table and retry the command.

```
** ERROR ** UNABLE TO CONVERT TIMESTAMP: DST table not loaded
```

Cause. The Guardian procedure CONVERTTIMESTAMP failed with an error.

Effect. The command is not executed.

Recovery. Make appropriate corrections to the daylight savings time (DST) table and retry the command.

```
** ERROR ** UNABLE TO CONVERT TIMESTAMP: ERROR UNKNOWN
```

Cause. The Guardian procedure CONVERTTIMESTAMP failed with an error.

Effect. The command is not executed.

Recovery. Make appropriate corrections to the daylight savings time (DST) table and retry the command.

```
** ERROR ** UNABLE TO CONVERT TIMESTAMP: Impossible LCT
```

Cause. The Guardian procedure CONVERTTIMESTAMP failed with an error.

Effect. The command is not executed.

Recovery. Make appropriate corrections to the daylight savings time (DST) table and retry the command.

```
* ERROR *   Underlying user ID does not exist.
```

Cause. An ADD ALIAS command specified a user ID that does not exist.

Effect. The command is rejected.

Recovery. Retry the command using the correct user ID.

```
* ERROR *   User has aliases associated with it.
```

Cause. A DELETE USER command specified a user that has aliases associated with it.

Effect. The command is rejected.

Recovery. Delete the aliases and retry the command.

```
EXTENDED DATA SEGMENT INITIALIZATION ERROR n
```

Cause. The program specified in a RUN command required more memory than is currently available.

Effect. The command is rejected.

Recovery. Retry the command when more memory is available.

```
EXTENDED SEGMENT SWAP FILE ERROR nnn
```

Cause. The program specified in a RUN command required more disk space than is currently available.

Effect. The command is rejected.

Recovery. Retry the command when more disk space is available.

```
GROUP-ID OR USER-ID NOT DEFINED; COMMAND NOT EXECUTED.
```

Cause. A *group-number* or *member-number* not defined for this system was used.

Effect. The command is not executed.

Recovery. Reenter the command with a valid *group-number* and *member-number*. (You can use the TACL USERS command to find the correct numbers associated with a user’s user name.)

```
GROUP-NAME OR USER-NAME NOT DEFINED; COMMAND NOT EXECUTED.
```

Cause. A *group-name* or *member-name* not defined for this system was used.

Effect. The command is not executed.

Recovery. Reenter the command with a valid *group-name* and *member-name*. (You can use the TACL USERS command to find the correct spelling of a person’s *group-name* and *member-name*.)

```
ILLEGAL HOME TERMINAL, ERROR nnn
```

Cause. The TERM option in a RUN specifies an illegal terminal.

Effect. The command is rejected.

Recovery. Correct the TERM option and retry the command.

```
ILLEGAL LIBRARY FILE FORMAT
FILE HAS UNDEFINED DATA BLOCKS
```

Cause. The LIB option in a RUN command specifies a library file that is corrupted.

Effect. The command is rejected.

Recovery. Specify a different library or restore the library. Then retry the command.

```
ILLEGAL LIBRARY FILE FORMAT
FILE NOT FIXED-UP BY BINDER
```

Cause. The LIB option in a RUN command specifies a library file that is corrupted.

Effect. The command is rejected.

Recovery. Specify a different library or restore the library. Then retry the command.

```
ILLEGAL LIBRARY FILE FORMAT
HEADER INITSEGS NOT CONSISTENT WITH SIZE
```

Cause. The LIB option in a RUN command specifies a library file that is corrupted.

Effect. The command is rejected.

Recovery. Specify a different library or restore the library. Then retry the command.

```
ILLEGAL LIBRARY FILE FORMAT
INVALID PEP
```

Cause. The LIB option in a RUN command specifies a library file that is corrupted.

Effect. The command is rejected.

Recovery. Specify a different library or restore the library. Then retry the command.

```
ILLEGAL LIBRARY FILE FORMAT
LIB FILE HAS MAIN PROCEDURE
```

Cause. The LIB option in a RUN command specifies a program file, not a library file.

Effect. The command is rejected.

Recovery. Specify a library file and retry the command.

```
ILLEGAL LIBRARY FILE FORMAT
NO DATA PAGES
```

Cause. The LIB option in a RUN command specifies a library file that is corrupted.

Effect. The command is rejected.

Recovery. Specify a different library or restore the library. Then retry the command.

```
ILLEGAL LIBRARY FILE FORMAT
NOT A DISK FILE
```

Cause. The LIB option in a RUN command specifies a library file that is not a disk file.

Effect. The command is rejected.

Recovery. Specify a different library file that is a disk file and retry the command.

```
ILLEGAL LIBRARY FILE FORMAT
NOT CORRECT FILE STRUCTURE
```

Cause. The LIB option in a RUN command specifies a library file that is corrupted.

Effect. The command is rejected.

Recovery. Specify a different library or restore the library. Then retry the command.

```
ILLEGAL LIBRARY FILE FORMAT
NOT FILE CODE 100
```

Cause. The LIB option in a RUN command specifies a library file that is not a library file.

Effect. The command is rejected.

Recovery. Specify a different library and retry the command.

```
ILLEGAL LIBRARY FILE FORMAT
NPERR^BADFILE ERROR SUBCODE nnn
```

Cause. The LIB option in a RUN command specifies a library file that is corrupted.

Effect. The command is rejected.

Recovery. Specify a different library or restore the library. Then retry the command.

```
ILLEGAL LIBRARY FILE FORMAT
REQUIRES LATER VERSION OF NONSTOP KERNEL
```

Cause. The LIB option in a RUN command specifies a library file that requires a later product version of the operating system.

Effect. The command is rejected.

Recovery. Specify a different library or install a later version of the operating system. Then retry the command.

```
ILLEGAL LIBRARY FILE FORMAT
RESIDENT SIZE GREATER THAN CODE AREA
```

Cause. The LIB option in a RUN command specifies a library file that is corrupted.

Effect. The command is rejected.

Recovery. Specify a different library or restore the library. Then retry the command.

```
ILLEGAL LIBRARY FILE FORMAT
UNRESOLVED REFERENCES FROM DATA BLOCK TO CODE BLOCK
```

Cause. The LIB option in a RUN command specifies a library file that is corrupted.

Effect. The command is rejected.

Recovery. Specify a different library or restore the library. Then retry the command.

```
ILLEGAL PROCESS DEVICE SUBTYPE
```

Cause. The program object file specified in a RUN command is corrupted.

Effect. The command is rejected.

Recovery. Specify a different program file or restore the file. Then retry the command.

```
ILLEGAL PROGRAM FILE FORMAT
FILE HAS UNDEFINED DATA BLOCKS
```

Cause. The program object file specified in a RUN command is corrupted.

Effect. The command is rejected.

Recovery. Specify a different program file or restore the file. Then retry the command.

```
ILLEGAL PROGRAM FILE FORMAT
FILE NOT FIXED-UP BY BINDER
```

Cause. The program object file specified in a RUN command is corrupted.

Effect. The command is rejected.

Recovery. Specify a different program file or restore the file. Then retry the command.

```
ILLEGAL PROGRAM FILE FORMAT
HEADER INITSEGS NOT CONSISTENT WITH SIZE
```

Cause. The program object file specified in a RUN command is corrupted.

Effect. The command is rejected.

Recovery. Specify a different program file or restore the file. Then retry the command.

```
ILLEGAL PROGRAM FILE FORMAT
INVALID PEP
```

Cause. The program object file specified in a RUN command is corrupted.

Effect. The command is rejected.

Recovery. Specify a different program file or restore the file. Then retry the command.

```
ILLEGAL PROGRAM FILE FORMAT
NO DATA PAGES
```

Cause. The program object file specified in a RUN command is corrupted.

Effect. The command is rejected.

Recovery. Specify a different program file or restore the file. Then retry the command.

```
ILLEGAL PROGRAM FILE FORMAT
NO MAIN PROCEDURE
```

Cause. The program object file specified in a RUN command is a library file.

Effect. The command is rejected.

Recovery. Specify a program file and retry the command.

```
ILLEGAL PROGRAM FILE FORMAT
NOT A DISK FILE
```

Cause. The program object file specified in a RUN command is not a disk file.

Effect. The command is rejected.

Recovery. Specify a different program file and retry the command.

```
ILLEGAL PROGRAM FILE FORMAT
NOT CORRECT FILE STRUCTURE
```

Cause. The program object file specified in a RUN command is corrupted.

Effect. The command is rejected.

Recovery. Specify a different program file or restore the file. Then retry the command.

```
ILLEGAL PROGRAM FILE FORMAT
NOT FILE CODE 100
```

Cause. The program object file specified in a RUN command is not a program file.

Effect. The command is rejected.

Recovery. Specify a different program file and retry the command.

```
ILLEGAL PROGRAM FILE FORMAT
NPERR^BADFILE ERROR SUBCODE nnn
```

Cause. The program object file specified in a RUN command is corrupted.

Effect. The command is rejected.

Recovery. Specify a different program file or restore the file. Then retry the command.

```
ILLEGAL PROGRAM FILE FORMAT
REQUIRES LATER VERSION OF NONSTOP KERNEL
```

Cause. The program file specified in a RUN command requires a later product version of the operating system.

Effect. The command is rejected.

Recovery. Specify a different program file or install a product later version of the operating system. Then retry the command.

```
ILLEGAL PROGRAM FILE FORMAT
RESIDENT SIZE GREATER THAN CODE AREA
```

Cause. The program object file specified in a RUN command is corrupted.

Effect. The command is rejected.

Recovery. Specify a different program file or restore the file. Then retry the command.

```
ILLEGAL PROGRAM FILE FORMAT
UNRESOLVED REFERENCES FROM DATA BLOCK TO CODE BLOCK
```

Cause. The program object file specified in a RUN command is corrupted.

Effect. The command is rejected.

Recovery. Specify a different program file or restore the file. Then retry the command.

```
ILLEGAL SYNTAX; COMMAND NOT EXECUTED.
```

Cause. The command was entered improperly. For example, a required comma (,) is missing, or an attribute name is misspelled.

Effect. The command is not executed.

Recovery. Look up the correct syntax for the command, check for proper spelling, and reenter the corrected command.

```
INTERNAL SAFECOM SPI ERROR nnn
```

Cause. SAFECOM encountered an internal error attempting to interpret a command.

Effect. The command is not executed.

Recovery. Exit SAFECOM and then rerun SAFECOM. If the error persists, contact your HP representative.

```
INTERNAL SCANNER PROCEDURE ERROR; COMMAND NOT EXECUTED.
```

Cause. SAFECOM encountered an internal error attempting to interpret a command.

Effect. The command is not executed.

Recovery. Exit SAFECOM and then rerun SAFECOM. If the error persists, contact your HP representative.

```
INVALID DATE SPECIFIED; COMMAND NOT EXECUTED.
```

Cause. An improper or invalid date was specified (usually, an invalid day-number or an improper year).

Effect. The command is not executed.

Recovery. Reenter the command, specifying a valid date.

```
INVALID MONTH SPECIFIED; COMMAND NOT EXECUTED.
```

Cause. An invalid month was specified.

Effect. The command is not executed.

Recovery. Reenter the command, specifying a valid month.

```
INVALID OBJECT TYPE SPECIFIED; COMMAND NOT EXECUTED.
```

Cause. An object type was specified that is not valid for the objects named.

Effect. The command is not executed.

Recovery. Reenter the command, naming the object type valid for the objects specified.

```
INVALID SECURITY GROUP SPECIFIED; COMMAND NOT EXECUTED.
```

Cause. An invalid security group name was specified.

Effect. The command is not executed.

Recovery. Correct the name and then retry the command.

```
LIBRARY FILE ERROR nnn
```

Cause. A problem exists with the library file specified by the LIB option in a RUN command.

Effect. The command is not executed.

Recovery. Look up the error number in the *System Messages Manual* and resolve the problem according to instructions given there.

```
LIBRARY FILE IS LOCKED
```

Cause. The LIB option in a RUN command specifies a library file that is locked.

Effect. The command is rejected.

Recovery. Wait until the library file is unlocked and retry the command.

```
MULTIPLE SPECIFICATIONS NOT ALLOWED (EXCEPT ACCESS); COMMAND NOT EXECUTED.
```

Cause. An attribute other than the ACCESS attribute was specified more than once in the same command.

Effect. The command is not executed.

Recovery. Reenter the command specifying each attribute only once.

```
NAME MUST BE IN A0000000 FORMAT; COMMAND NOT EXECUTED.
```

Cause. You specified an audit file name in an incorrect format.

Effect. The command is not executed.

Recovery. Reenter the command using the proper format for the audit file name.

```
NEWPROCESS ERROR nnnnnnn
```

Cause. A NEWPROCESS failure occurred during the attempted execution of a RUN command.

Effect. The command is not executed.

Recovery. Look up the error number in the *Operator Messages Manual* and resolve the problem according to instructions given there.

```
NO HELP IS AVAILABLE
```

Cause. No help exists for the specified topic.

Effect. The command is not executed.

Recovery. Reenter the command specifying a topic for which help exists, or type HELP or HELP COMMANDS to see lists of topics for which help is available.

```
NO PROCESS CONTROL BLOCK AVAILABLE
```

Cause. The system limit on process control blocks (PCBs) has been reached.

Effect. The command is not executed.

Recovery. Retry the command later.

```
NO RESIDENT SPACE FOR MEMORY MAPS
```

Cause. Insufficient memory is available to execute the specified program.

Effect. The command is not executed.

Recovery. Retry the command later.

```
NO SUCH LINE; COMMAND NOT EXECUTED
```

Cause. The line number or text string specified in an FC, ?, or ! command does not exist in the command history buffer.

Effect. The command is not executed.

Recovery. Reenter the command with a line number or text string that exists in the buffer.

```
ONLY DEFINED GROUP-ID'S AND USER-ID'S MAY BE ADDED; COMMAND NOT EXECUTED.
```

Cause. A *`group-number`* or *`member-number`* not defined for this system was used.

Effect. The command is not executed.

Recovery. Reenter the command with a valid *`group-number`* and *`member-number`*. (You can use the TACL USERS command to find the correct numbers associated with a user's user name.)

```
Only super group USERS and ALIAS can become member of this group
```

Cause. An attempt was made to add a non-super group member to a restricted group.

Effect. The command is not executed.

Recovery. Ensure that you are adding a super group member to the restricted group and reenter the command.

```
OPERATION NOT PERMITTED ON THIS GROUP; COMMAND NOT EXECUTED
```

Cause. The group specified in the command does not support ACL.

Effect. The command is not executed.

Recovery. Reenter the command with a group having group number less than 65536.

```
PARSER CALLED APPLY WITH BAD PRODUCTION RULE.
```

Cause. SAFECOM encountered an internal error while attempting to interpret the command.

Effect. The command is not executed.

Recovery. Contact your HP representative.

```
PROCESS NAME ERROR nnn
```

Cause. The NAME option in a RUN command specified an invalid or duplicate process name.

Effect. The command is not executed.

Recovery. Look up the error number in the *System Messages Manual* and resolve the problem according to instructions given there.

```
PROGRAM AND LIBRARY FILES MUST BE DIFFERENT
```

Cause. The program and library files specified in a RUN command have the same names.

Effect. The command is not executed.

Recovery. Change one of the names and retry the command.

```
PROGRAM FILE ERROR nnn
```

Cause. A file error occurred during an attempt to execute a RUN command.

Effect. The command is not executed.

Recovery. Look up the error number in the *System Messages Manual* and resolve the problem according to instructions given there.

```
PROGRAM FILE IS LOCKED
```

Cause. The program object file specified in a RUN command is locked.

Effect. The command is rejected.

Recovery. Wait until the program file is unlocked and retry the command.

```
PROMPT STRING SHOULD BE LESS THAN 81 CHARACTERS; COMMAND NOT EXECUTED.
```

Cause. The specified prompt string is too long.

Effect. The command is rejected.

Recovery. Shorten the prompt to fewer than 81 characters and then retry the command.

```
SAFECOM RUNS ONLY IN NONSTOP SYSTEMS; THIS IS A NONSTOP1+ SYSTEM
```

Cause. Your system is an HP NonStop 1+ system.

Effect. SAFECOM abends.

Recovery. None. SAFECOM runs only on NonStop systems.

```
SAFEGUARD/NONSTOP KERNEL VERSIONS (Xnn/Zmm) ARE INCOMPATIBLE; COMMAND NOT
EXECUTED.
```

Cause. Your version of the Safeguard software does not match the version of the operating system currently running on your system.

Effect. The command is not executed.

Recovery. Ask your system manager to install a version of the Safeguard software that matches the version of the operating system currently running on your system.

```
SECURITY STRING MUST BE IN "xxxx" FORMAT; COMMAND NOT EXECUTED.
```

Cause. The security string specified for a GUARDIAN SECURITY attribute is not four characters long.

Effect. The command is not executed.

Recovery. Reenter the command with a properly formed security string.

```
SECURITY STRING MUST CONTAIN O, G, A, U, C, or N; COMMAND NOT EXECUTED.
```

Cause. The security string specified for a GUARDIAN SECURITY attribute contains an illegal character.

Effect. The command is not executed.

Recovery. Reenter the command using legal characters in the security string.

```
SWAP FILE ERROR nnn
```

Cause. A file error occurred during an attempt to execute a RUN command.

Effect. The command is not executed.

Recovery. Look up the error number in the *System Messages Manual* and resolve the problem according to instructions given there.

```
SYSTEM SPECIFIED DOES NOT EXIST; COMMAND NOT EXECUTED.
```

Cause. A remote system named in the REMOTEPASSWORD attribute is not known at your node, or a system specified in a GUARDIAN VOLUME attribute does not exist.

Effect. The command is not executed.

Recovery. Reenter the command with the correct system name or up the EXPAND line handlers to make the system name known at your node.

```
THE ENDING QUOTE MARK IS MISSING; COMMAND NOT EXECUTED.
```

Cause. A quoted string is not properly terminated with a trailing quote (") character.

Effect. The command is not executed.

Recovery. Fix the command by correctly terminating the quoted string and reenter the command.

```
THIS CHARACTER IS UNRECOGNIZABLE; COMMAND NOT EXECUTED.
```

Cause. The current command contains a non-ASCII character (a character whose octal representation value exceeds %177).

Effect. The command is not executed.

Recovery. Reenter the command. If the error persists, it might indicate either a hardware problem or an internal error in SAFECOM. Contact your HP representative.

```
THIS NAME IS TOO LONG (> 31 CHARACTERS); COMMAND NOT EXECUTED.
```

Cause. The command contains a name or keyword whose length exceeds 31 characters. Typically, this error occurs when a word is misspelled or when a missing punctuation character causes two words to be concatenated.

Effect. The command is not executed.

Recovery. Correct the spelling of the name or keyword, or supply the missing punctuation, and reenter the command.

```
THIS NAME MUST BE NO LONGER THAN n CHARACTERS; COMMAND NOT EXECUTED.
```

Cause. The command contains a name or keyword whose length exceeds that allowed. Typically, this error occurs when a word is misspelled or when a missing punctuation character causes two words to be concatenated.

Effect. The command is not executed.

Recovery. Correct the spelling of the name or keyword, or supply the missing punctuation, and reenter the command.

```
THIS NUMBER IS OUT OF RANGE; COMMAND NOT EXECUTED.
```

Cause. A number in the command exceeds its maximum allowable value.

Effect. The command is not executed.

Recovery. Correct the number to within the allowable range and reenter the command.

```
THIS NUMBER IS RESERVED; COMMAND NOT EXECUTED.
```

Cause. An attempt to add the group failed because the group number is reserved for future use by HP.

Effect. The command is not executed.

Recovery. Specify a group number within the allowable range and reenter the command.

```
THIS NUMBER MUST BE GREATER THAN 0; COMMAND NOT EXECUTED.
```

Cause. A number in the command is less than the minimum allowable value.

Effect. The command is not executed.

Recovery. Correct the number to within the allowable range and reenter the command.

```
THIS NUMBER MUST BE LESS THAN OR EQUAL TO n; COMMAND NOT EXECUTED.
```

Cause. A number in the command exceeds its maximum allowable value.

Effect. The command is not executed.

Recovery. Correct the number to within the allowable range and reenter the command.

```
THIS TOKEN MUST BE NO LONGER THAN 255 CHARACTERS.
```

Cause. Descriptive text exceeds 255 characters.

Effect. The command is not executed.

Recovery. Shorten the text to 255 characters or fewer and reenter the command.

```
TOO MANY COMPLEX COMMANDS IN LINE; COMMAND NOT EXECUTED.
```

Cause. Too many commands were entered on a single command line.

Effect. The command indicated by the circumflex characters (^^^) is not executed. Any commands following the indicated command are not executed.

Recovery. Enter the sequence of commands on two or more command lines.

```
TOO MANY ITEMS IN GROUP LIST; COMMAND NOT EXECUTED.
```

Cause. Too many groups were specified in a command.

Effect. The command is rejected.

Recovery. Reduce the number of groups in the command to 32 or fewer and then retry the command.

```
TOO MANY MEMBERS SPECIFIED; COMMAND NOT EXECUTED.
```

Cause. More than 32 members are being added to a group, or more than 32 members are being removed from a group.

Effect. The command is rejected.

Recovery. Specify 32 or fewer members in the offending MEMBER clause, and retry the command.

```
UNABLE TO ALLOCATE SEGMENT FOR SAFECOM.
```

Cause. Insufficient memory is available in the CPU in which SAFECOM is running.

Effect. The SAFECOM session does not run properly.

Recovery. Try again later or try running SAFECOM in different CPU.

```
UNABLE TO OPEN THE OBEY FILE; FOUR OBEY FILES ARE ALREADY OPEN.
```

Cause. Four command files are already open, and the fourth command file contains an OBEY command that attempts to open a fifth command file. Only four command files can be opened at the same time.

Effect. This OBEY command is not executed, and the next line (if any) from the fourth OPEN command file is executed.

Recovery. Structure nested command files so that no more than four command files are open at the same time.

```
UNABLE TO WRITE OPEN MESSAGE TO PROCESS; ERROR nnnn.
```

Cause. An error was encountered while attempting to open a process started through the RUN command.

Effect. The command is not executed.

Recovery. Diagnose the file-system error included in the message, correct the error, and then retry the command.

```
UNDEFINED GROUP; COMMAND NOT EXECUTED
```

Cause. A specified group name does not exist. Possibly an uppercase name was typed in lowercase.

Effect. The command is rejected.

Recovery. Check for spelling or typing errors, correct the group name, and then retry the command.

```
UNDEFINED SUBJECT; COMMAND NOT EXECUTED
```

Cause. A specified member in an ADD or ALTER group command does not exist.

Effect. The command is rejected.

Recovery. Specify a valid user or alias and then retry the command.

```
UNLICENSED PRIVILEGED PROGRAM
```

Cause. The program object file specified in a RUN command contains unlicensed privileged code.

Effect. The command is rejected.

Recovery. Contact your system administrator, have the program licensed, and then retry the command.

```
USER NAMES MUST CONTAIN ALPHANUMERIC CHARACTERS ONLY; COMMAND NOT
EXECUTED.
```

Cause. An ADD USER command specified a user name that contains an illegal character.

Effect. The command is rejected.

Recovery. Reenter the command and specify a valid user name. Only alphabetic characters and numbers are allowed in the name.

```
USERID FILE NOT ACCESSIBLE (CCL); COMMAND NOT EXECUTED.
```

Cause. Some of the standard library procedures (for example, USERIDTOUSERNAME) cannot open or read the USERID file. Typically, this file might be missing or not protected by a Safeguard ACL when ACL-REQUIRED-DISKFILE is in effect.

Effect. The command is not executed.

Recovery. If the USERID file is missing, recover it immediately. If ACL-REQUIRED-DISKFILE is in effect, either turn it off or protect this file with an authorization record that includes an appropriate ACL.

```
* WARNING *   Cannot remove derived group group-name.
```

Cause. An attempt was made to remove a user from that user’s administrative group.

Effect. The command is accepted, but the user is not removed from the administrative group.

Recovery. None.

```
WARNING - PROCESS HAS UNDEFINED EXTERNAL(S)
```

Cause. The program object file contains external references that cannot be resolved.

Effect. The command is executed, but results might be spurious.

Recovery. Inform your system administrator. Retry the command after the program or library file has been corrected.

```
* WARNING * RECORD FOR objtype objname : NOT FOUND
```

Cause. No authorization record exists in the object database for the indicated object.

Effect. The INFO command is not executed for the object.

Recovery. Possibly you misspelled the object name. If so, reenter the command with the correct object name.

```
* WARNING * RECORD FOR SECURITY-GROUP SECURITY-AUDITOR: NOT FOUND
```

Cause. You attempted to perform alter, delete, freeze, thaw, or info command on the SECURITY-AUDITOR security group that does not exist.

Effect. The command is not executed.

Recovery. None.

```
* WARNING * RECORD FOR SECURITY-GROUP SECURITY-PRV-ADMINISTRATOR: NOT
FOUND
```

Cause. You attempted to perform alter, delete, freeze, thaw, or info command on the SECURITY-PRV-ADMINISTRATOR security group that does not exist.

Effect. The command is not executed.

Recovery. None.

```
* WARNING * RECORD FOR SECURITY-GROUP SECURITY-MEDIA-ADMIN: NOT FOUND
```

Cause. You attempted to perform alter, delete, freeze, thaw, or info command on the SECURITY-MEDIA-ADMIN security group that does not exist.

Effect. The command is not executed.

Recovery. None.

```
* WARNING * RECORD FOR SECURITY-GROUP SECURITY-PERSISTENCE-ADMIN: NOT
FOUND
```

Cause. A user attempts to perform alter, delete, freeze, thaw, or info command on the SECURITY-PERSISTENCE-ADMIN security group that does not exist.

Effect. The command does not execute.

Recovery. None.

```
* WARNING * RECORD FOR SECURITY-GROUP SECURITY-MEDIA-ADMIN: NOT FOUND
```

Cause. You are using SAFECOM in syntax checking mode. The only commands that are executed are SYNTAX, ASSUME, OBEY, and EXIT.

Effect. The command is not executed.

Recovery. Execute the SYNTAX command if you want to leave syntax mode.

```
WARNING - STARTUP MESSAGE NOT READ.
```

Cause. The program specified in a RUN command stopped prematurely.

Effect. The command is not successfully executed.

Recovery. Retry the command later.

```
* WARNING *   User would have too many groups.
```

Cause. An attempt was made to execute a command that would make a user a member of more than 32 groups.

Effect. The command is accepted, but the user is not added to the group’s member list. Other parts of the command are executed.

Recovery. Remove the user from another group and then add the user to the member list of this group.

```
* WARNING * RECORD FOR DISKFILE diskfilename: DISKFILE OWNER CHANGED TO
MATCH PROT RECORD OWNER AND PROGID NOTSET AS BOTH DIFFER.
```

Cause. An attempt was made to set the `PROGID` for the diskfile record whose primary owner for protection record and the diskfile flab owner are different. The owner attribute was not provided in the command.

Effect. The command is accepted, however, the `PROGID` is not set and the diskfile flab owner is changed to match the protection record owner.

Recovery. None.

Note. This warning message is supported only on systems running J06.15 and later J-series RVUs and H06.26 and later H-series RVUs.

```
* WARNING * RECORD FOR DISKFILE diskfilename: DISKFILE FLAB OWNER IS
CHANGED TO MATCH PROTECTION RECORD OWNER.
```

Cause. An attempt was made to alter a diskfile record attribute (other than `PROGID`). The primary owner for protection record and the diskfile flab owner are different. The owner attribute is not provided in the command.

Effect. The command is accepted, but the diskfile flab owner is changed to match protection record owner.

Recovery. None.

Note. This warning message is supported only on systems running J06.15 and later J-series RVUs and H06.26 and later H-series RVUs.

```
WHERE EXPRESSION TOO LONG; COMMAND NOT EXECUTED.
```

Cause. More than 32 WHERE items were specified in a WHERE clause.

Effect. The command is not executed.

Recovery. Reduce the number of WHERE items to 32 or fewer and then retry the command.

```
WILD CARDS ARE NOT ALLOWED HERE; COMMAND NOT EXECUTED
```

Cause. A name in the command contains an illegal use of wild-card characters.

Effect. The command is not executed.

Recovery. Review the valid use of wild-card characters, correct the name, and retry the command.

```
*ERROR* DISKFILE-PATTERN MUST SPECIFY A WILDCARD FOR SUBVOL OR FILE
```

Cause. A pattern-spec does not include at least one wildcard in either the subvolume or file name.

Effect. The command is not executed.

Recovery. Review the valid use of wild-card characters, correct the name, and retry the command.

```
*ERROR* TEXT DESCRIPTION FIELD CONTAINS INVALID CHARACTERS; COMMAND NOT
EXECUTED
```

Cause. The text string specified for the TEXT-DESCRIPTION attribute in a command contained nonprintable characters.

Effect. The command is not executed.

Recovery. Remove the nonprintable characters or replace them with printable characters. The printable characters are:

- Lowercase letters a through z
- Uppercase letters A through Z
- Digits 0 through 9
- Special characters { } ! @ # $ % ^ & * ( ) - _ + = { } ~ ‘ : ; ? / > . < , .
- Space

Note. The nonprintable characters error message is supported only on systems running G06.27 and later G-series RVUs and H06.06 and later H-series RVUs.

```
* ERROR *   RECORD FOR DISKFILE filename : TRUST ONLY PROGRAM OBJECT FILES
```

Cause. An attempt to set the TRUST attribute to ME or SHARED for a disk file failed because the file is not a program object file.

Effect. The command is rejected.

Recovery. None

```
ERROR: THIS NUMBER MUST BE LESS THAN OR EQUAL TO MAXPASSWORDLEN; COMMAND
NOT EXECUTED.
```

Cause. PASSWORD-MINIMUM-LENGTH attribute is greater than PASSWORD-MAXIMUM-LENGTH, when PASSWORD-ENCRYPT is ON and PASSWORD-ALGORITHM is HMAC256.

Effect. The command is not executed.

Recovery. PASSWORD-MINIMUM-LENGTH attribute should be lesser than or equal to PASSWORD-MAXIMUM-LENGTH.

Note. This error message is supported only on systems running G06.31 and later G-series RVUs and H06.08 and later H-series RVUs.

```
ERROR: THIS NUMBER MUST BE LESS THAN OR EQUAL TO 8; COMMAND NOT EXECUTED.
```

Cause. PASSWORD-MINIMUM-LENGTH attribute is greater than eight, when PASSWORD-ENCRYPT is OFF or, PASSWORD-ENCRYPT is ON and PASSWORD-ALGORITHM is DES.

Effect. The command is not executed.

Recovery. PASSWORD-MINIMUM-LENGTH attribute should be less than or equal to eight.

Note. This error message is supported only on systems running G06.31 and later G-series RVUs and H06.08 and later H-series RVUs.

```
ERROR: PASSWORD-MAXIMUM-LENGTH CANNOT BE MODIFIED UNLESS
PASSWORD-ALGORITHM= HMAC256 AND ENCRYPT = ON; COMMAND NOT EXECUTED
```

Cause. PASSWORD-ALGORITHM is DES and PASSWORD-ENCRYPT is ON or, PASSWORD-ALGORITHM is DES and PASSWORD-ENCRYPT is OFF, when PASSWORD-MAXIMUM-LENGTH is not equal to eight.

Effect. The command is not executed.

Recovery. PASSWORD-MAXIMUM-LENGTH is equal to eight or PASSWORD-ALGORITHM is HMAC256 and PASSWORD-ENCRYPT is ON.

Note. This error message is supported only on systems running G06.31 and later G-series RVUs and H06.08 and later H-series RVUs.

```
ERROR: PASSWORD-MINIMUM-LENGTH CANNOT BE GREATER THAN EIGHT UNLESS
PASSWORD-ALGORITHM= HMAC256 AND ENCRYPT = ON; COMMAND NOT EXECUTED
```

Cause. PASSWORD-ALGORITHM is DES and PASSWORD-ENCRYPT is OFF, when PASSWORD-MINIMUM-LENGTH is greater than eight.

Effect. The command is not executed.

Recovery. PASSWORD-MINIMUM-LENGTH is equal to eight or PASSWORD-ALGORITHM is HMAC256 and PASSWORD-ENCRYPT is ON.

Note. This error message is supported only on systems running G06.31 and later G-series RVUs and H06.08 and later H-series RVUs.

```
THIS NUMBER CANNOT BE LESS THAN 8 OR GREATER THAN 64; COMMAND NOT EXECUTED
```

Cause. PASSWORD-ALGORITHM is HMAC256 and PASSWORD-ENCRYPT is ON, when PASSWORD-MAXIMUM-LENGTH is less than eight or greater than 64.

Effect. The command is not executed.

Recovery. PASSWORD-MAXIMUM-LENGTH should be greater than or equal to PASSWORD-MINIMUM-LENGTH.

Note. This error message is supported only on systems running G06.31 and later G-series RVUs and H06.08 and later H-series RVUs.

```
PASSWORD-MAXIMUM-LENGTH  <n> MUST BE GREATER THAN OR EQUAL TO
PASSWORD-MINIMUM-LENGTH  <n> ; COMMAND NOT EXECUTED
```

Cause. PASSWORD-ALGORITHM is HMAC256 and PASSWORD-ENCRYPT is ON and PASSWORD-MAXIMUM-LENGTH is less than PASSWORD-MINIMUM-LENGTH.

Effect. The command is not executed.

Recovery. PASSWORD-MAXIMUM-LENGTH should be greater than or equal to PASSWORD-MINIMUM-LENGTH.

Note. This error message is supported only on systems running G06.31 and later G-series RVUs and H06.08 and later H-series RVUs.

```
MAX AUDIT EXCLUDE VALUES 5 EXCEEDED
```

Cause. An attempt to specify audit exclusion failed because the number of values specified for the attribute AUDIT-EXCLUDE-VALUE was more than five.

Effect. The command is rejected.

Recovery. Specify five or lesser number of values for the attribute AUDIT-EXCLUDE-VALUE.

Note. This error message is supported only on systems running J06.03 and later J-series RVUs, H06.14 and later H-series RVUs, and G06.32 and later G-series RVUs.

```
BOTH THE FIELDS AUDIT-EXCLUDE-FIELD and AUDIT-EXCLUDE-VALUE ARE REQUIRED
```

Cause. An attempt to specify audit exclusion failed because both the attributes AUDIT-EXCLUDE-FIELD and AUDIT-EXCLUDE-VALUE were not set

Effect. The command is rejected.

Recovery. Specify values for both attributes AUDIT-EXCLUDE-FIELD and AUDIT-EXCLUDE-VALUE in a single command line.

Note. This error message is supported only on systems running J06.03 and later J-series RVUs and H06.14 and later H-series RVUs, and G06.32 and later G-series RVUs.

```
AUDIT-EXCLUDE-FIELD and AUDIT-EXCLUDE-VALUE mismatch
```

Cause. The values specified by attribute AUDIT-EXCLUDE-VALUE is invalid for the corresponding fieldname specified by AUDIT-EXCLUDE-FIELD.

Effect. The command is rejected.

Recovery. Review the valid use of values set by AUDIT-EXCLUDE-VALUE and retry the command.

Note. This error message is supported only on systems running J06.03 and later J-series RVUs, H06.14 and later H-series RVUs, and G06.32 and later G-series RVUs.

B Disk-File Access Rules

Table B-1 on page B-2 shows how disk file access rules are evaluated depending on how the Safeguard software applies the access control lists (ACL) in disk file, volume, and subvolume protection records.

FIRST-RULE, FIRST-ACL, and ALL are the settings allowed for the Safeguard configuration attribute COMBINATION-DISKFILE. This attribute defines the manner in which overlapping ACLs are resolved for access to volumes, subvolumes, and disk files.

FIRST-RULE indicates the Safeguard software uses the first ACL that contains the specified user ID. FIRST-ACL indicates the Safeguard software uses the first ACL it finds regardless of whether the ACL contains the specified user ID. ALL indicates the Safeguard software uses all available ACLs.

CHECK-DISKFILE-PATTERN establishes whether ACLs from a disk file pattern's protection record can be used to determine disk file access. The FIRST value says to first perform a disk file pattern search for a matching pattern, and only if the result of the search is NORECORD, then will a normal search of remaining object protection records occur. The LAST value says to first perform a normal search of all object protection records (except diskfile pattern), and only if the result of the search is NORECORD, then will a disk file pattern search for a matching pattern be performed. The OFF value says to not perform any disk file pattern searches to determine disk file access. The ONLY value says to perform only the pattern search and do not do the normal search. OFF is the initial value. This attribute defines part of the SAFEGUARD global configuration. For more diskfile-pattern information, see the *Safeguard User's Guide*.

In Table B-1 on page B-2, *`Level`* refers to the direction in which the Safeguard software searches ACLs. The evaluation depends on the direction of the search. The search direction is determined by Safeguard configuration attribute DIRECTION-DISKFILE, which can be set to either VOLUME-FIRST or FILENAME-FIRST.

If the search direction is VOLUME-FIRST, the volume ACL is searched first, subvolume ACL second, and disk file ACL third. If the search direction is FILENAME-FIRST, the disk file ACL is searched first, subvolume second, and volume third.

The CHECK-VOLUME, CHECK-SUBVOLUME, and CHECK-FILENAME configuration attributes allow you to selectively enable or disable the checking of ACLs at a particular level. For example, if CHECK-VOLUME is OFF, Safeguard does not check volume ACLs for attempts to access a disk file.   If one of these configuration attributes is set to OFF, the access result is the same as if that level had No Record (indicated by NR in Table B-1 on page B-2). However, if a disk file protection record exists and if CHECK-VOLUME, CHECK-SUBVOLUME, CHECK-FILENAME and ACL-REQUIRED-DISKFILE are OFF, this is treated as a special frozen ACL case. Only the primary owner of the disk file, primary owner's local group manager, and the local super ID are allowed access. As a special case, if an authorization event-exit process (SEEP) is running, access is granted based on SEEP's decision (allow or deny access) instead of the frozen ACL rules.

The settings of CHECK-VOLUME, CHECK-SUBVOLUME, and CHECK-FILENAME have no effect when an attempt is made to create a disk file. Any attempt to create a disk file is subject to access checking at all levels, regardless of the settings of these configuration attributes.

Table B-1 uses these abbreviations:

Y	ACL evaluates to YES	Permit	Access Permitted
N	ACL evaluates to NO	Deny	Access Denied
NM	ACL contains No Mention	G90	Guardian rules apply
NR	No ACL exists (No Record)		

Table B-1. Disk-File Access Evaluation (page 1 of 3)

Levels			**Combinations/Evaluations**		
First	**Second**	**Third**	**FIRST-ACL**	**FIRST-RULE**	**ALL**
NM	**N**	**Y**	**Deny**	**Deny**	**Deny**
Y	Y	Y	Permit	Permit	Permit
Y	Y	N	Permit	Permit	Deny
Y	Y	NM	Permit	Permit	Deny
Y	Y	NR	Permit	Permit	Permit
Y	N	Y	Permit	Permit	Deny
Y	N	N	Permit	Permit	Deny
Y	N	NM	Permit	Permit	Deny
Y	N	NR	Permit	Permit	Deny
Y	NM	Y	Permit	Permit	Deny
Y	NM	N	Permit	Permit	Deny
Y	NM	NM	Permit	Permit	Deny
Y	NM	NR	Permit	Permit	Deny
Y	NR	Y	Permit	Permit	Permit
Y	NR	N	Permit	Permit	Deny
Y	NR	NM	Permit	Permit	Deny
Y	NR	NR	Permit	Permit	Permit
N	Y	Y	Deny	Deny	Deny
N	Y	N	Deny	Deny	Deny
N	Y	NM	Deny	Deny	Deny
N	Y	NR	Deny	Deny	Deny
N	N	Y	Deny	Deny	Deny
N	N	N	Deny	Deny	Deny

Table B-1. Disk-File Access Evaluation (page 2 of 3)

Levels			**Combinations/Evaluations**		
First	**Second**	**Third**	**FIRST-ACL**	**FIRST-RULE**	**ALL**
NM	**N**	**Y**	**Deny**	**Deny**	**Deny**
N	N	NM	Deny	Deny	Deny
N	N	NR	Deny	Deny	Deny
N	NM	Y	Deny	Deny	Deny
N	NM	N	Deny	Deny	Deny
N	NM	NM	Deny	Deny	Deny
N	NM	NR	Deny	Deny	Deny
N	NR	Y	Deny	Deny	Deny
N	NR	N	Deny	Deny	Deny
N	NR	NM	Deny	Deny	Deny
N	NR	NR	Deny	Deny	Deny
NM	Y	Y	Deny	Permit	Deny
NM	Y	N	Deny	Permit	Deny
NM	Y	NM	Deny	Permit	Deny
NM	Y	NR	Deny	Permit	Deny
NM	N	N	Deny	Deny	Deny
NM	N	NM	Deny	Deny	Deny
NM	N	NR	Deny	Deny	Deny
NM	NM	Y	Deny	Permit	Deny
NM	NM	N	Deny	Deny	Deny
NM	NM	NM	Deny	Deny	Deny
NM	NM	NR	Deny	Deny	Deny
NM	NR	Y	Deny	Permit	Deny
NM	NR	N	Deny	Deny	Deny
NM	NR	NM	Deny	Deny	Deny
NM	NR	NR	Deny	Deny	Deny
NR	Y	Y	Permit	Permit	Permit
NR	Y	N	Permit	Permit	Deny
NR	Y	NM	Permit	Permit	Deny
NR	Y	NR	Permit	Permit	Permit
NR	N	Y	Deny	Deny	Deny
NR	N	N	Deny	Deny	Deny
NR	N	NM	Deny	Deny	Deny

Table B-1. Disk-File Access Evaluation (page 3 of 3)

Levels			**Combinations/Evaluations**		
First	**Second**	**Third**	**FIRST-ACL**	**FIRST-RULE**	**ALL**
NM	**N**	**Y**	**Deny**	**Deny**	**Deny**
NR	N	NR	Deny	Deny	Deny
NR	NM	Y	Deny	Permit	Deny
NR	NM	N	Deny	Deny	Deny
NR	NM	NM	Deny	Deny	Deny
NR	NM	NR	Deny	Deny	Deny
NR	NR	Y	Permit	Permit	Permit
NR	NR	N	Deny	Deny	Deny
NR	NR	NM	Deny	Deny	Deny
NR	NR	NR	G90 *	G90 *	G90 *

* Indicates that access is denied if ACL-REQUIRED

Table B-2. CHECK-DISKFILE-PATTERN settings

Result from:		**CHECK-DISKFILE-PATTERN value**			
Normal	**Pattern**	**OFF**	**FIRST**	**LAST**	**ONLY**
Y	Y	Y[1]	Y[4]	Y[3]	Y[6]
Y	N	Y[1]	N[4]	Y[3]	N[6]
Y	NR	Y[1]	Y[2]	Y[3]	NR[6]

N the request is denied (NO)
Y the request is granted (YES)
NR no norecord was found (NORECORD)

CHECK-DISKFILE-PATTERN OFF searches only for normal protection records.
CHECK-DISKFILE-PATTERN FIRST searches for a pattern protection record, and if the result is NORECORD, then searches for a normal protection record.
CHECK-DISKFILE-PATTERN LAST searches for a normal protection record, and if the result is NORECORD, then searches for a pattern protection record.
CHECK-DISKFILE-PATTERN ONLY searches only for a pattern protection record.

[1] NORMALOFF Check-Diskfile-Pattern is OFF and the OUTCOME is determined by only searching for a normal protection record.
[2] NORMALFIRST Check-Diskfile-Pattern is FIRST and the OUTCOME is determined by a normal protection record because the pattern search resulted in norecord.
[3] NORMALLAST Check-Diskfile-Pattern is LAST and the OUTCOME is determined by a pattern protection record because the normal search resulted in norecord.
[4] PATTERNFIRST Check-Diskfile-Pattern is FIRST and the OUTCOME is determined by a pattern protection record.
[5] PATTERNLAST Check-Diskfile-Pattern is LAST and the OUTCOME is determined by a pattern protection record because the normal search resulted in norecord.
[6] PATTERNONLY Check-Diskfile-Pattern is ONLY and the OUTCOME is determined by only search for a pattern protection record.

Table B-2. CHECK-DISKFILE-PATTERN settings

Result from:		CHECK-DISKFILE-PATTERN value			
Normal	**Pattern**	**OFF**	**FIRST**	**LAST**	**ONLY**
N	Y	N[1]	Y[4]	N[3]	Y[6]
N	N	N[1]	N[4]	N[3]	N[6]
N	NR	N[1]	N[2]	N[3]	NR[6]
NR	Y	NR[1]	Y[4]	Y[5]	Y[6]
NR	N	NR[1]	N[4]	N[5]	N[6]
NR	NR	NR[1]	NR[2]	NR[5]	NR[6]

N the request is denied (NO)
Y the request is granted (YES)
NR no norecord was found (NORECORD)

CHECK-DISKFILE-PATTERN OFF searches only for normal protection records.
CHECK-DISKFILE-PATTERN FIRST searches for a pattern protection record, and if the result is NORECORD, then searches for a normal protection record.
CHECK-DISKFILE-PATTERN LAST searches for a normal protection record, and if the result is NORECORD, then searches for a pattern protection record.
CHECK-DISKFILE-PATTERN ONLY searches only for a pattern protection record.

[1] NORMALOFF Check-Diskfile-Pattern is OFF and the OUTCOME is determined by only searching for a normal protection record.
[2] NORMALFIRST Check-Diskfile-Pattern is FIRST and the OUTCOME is determined by a normal protection record because the pattern search resulted in norecord.
[3] NORMALLAST Check-Diskfile-Pattern is LAST and the OUTCOME is determined by a pattern protection record because the normal search resulted in norecord.
[4] PATTERNFIRST Check-Diskfile-Pattern is FIRST and the OUTCOME is determined by a pattern protection record.
[5] PATTERNLAST Check-Diskfile-Pattern is LAST and the OUTCOME is determined by a pattern protection record because the normal search resulted in norecord.
[6] PATTERNONLY Check-Diskfile-Pattern is ONLY and the OUTCOME is determined by only search for a pattern protection record.

Table B-3 describes CHECK-DISKFILE-PATTERN settings when it is set to MID and Direction-Diskfile is set to Filename-First. In this case DISKFILE-PATTERN ACL gets evaluated after DISKFILE ACL evaluation. If the DISKFILE-PATTERN ACL evaluation results in NORECORD, Normal ACL evaluates SUBVOLUME and VOLUME ACL. The first column indicates the DISKFILE ACL setting and the second column indicates the DISKFILE PATTERN setting. Last three columns indicate the final access evaluation based on the Safeguard global configuration attribute COMBINATION-DISKFILE values FIRST-ACL, FIRST-RULE and ALL.

Note. This setting is only supported by systems running J06.08 and later J-series RVUs and H06.18 and later H-series RVUs.

Table B-4 describes CHECK-DISKFILE-PATTERN settings when it is set to MID and Direction-Diskfile is set to Volume-First. In this case DISKFILE-PATTERN ACL gets evaluated after VOLUME and SUBVOLUME ACL evaluation. If the DISKFILE-PATTERN ACL evaluation results in NORECORD, Normal ACL evaluates DISKFILE ACL. The first column indicates the VOLUME ACL setting, the second column indicates the SUBVOLUME ACL setting and the third column indicates the DISKFILE PATTERN setting. The last three columns show the final access evaluation based on the Safeguard global configuration attribute COMBINATION-DISKFILE values FIRST-ACL, FIRST-RULE and ALL.

Note. This setting is only supported by systems running J06.08 and later J-series RVUs and H06.18 and later H-series RVUs.

Table B-3. CHECK-DISKFILE-PATTERN settings when value is MID and Direction Diskfile is Filename-First

Configuration Direction Diskfile: Filename-First		Evaluation (Combination Diskfile)		
DISKFILE ACL	**DISKFILE PATTERN ACL**	**FIRST-ACL**	**FIRST-RULE**	**ALL**
Y	Y	Permit	Permit	Permit
Y	N	Permit	Permit	Deny
Y	NR	Permit	Permit	Normal
N	Y	Deny	Deny	Deny
N	N	Deny	Deny	Deny
N	NR	Deny	Deny	Deny
NM	Y	Deny	Permit	Deny
NM	N	Deny	Deny	Deny
NM	NR	Deny	Normal	Deny
NR	Y	Permit	Permit	Permit
NR	N	Deny	Deny	Deny
NR	NR	Normal	Normal	Normal

Y - ACL evaluates a YES
N - ACL evaluates a NO
NR - No ACL exists (NORECORD)
NM - ACL contains no mention
Permit - Access permitted
Deny - Access Denied
Normal - Normal ACL evaluation for SUBVOL and VOLUME

Table B-4. CHECK-DISKFILE-PATTERN settings when value is MID and Direction Diskfile is Volume-First

Configuration Direction Diskfile: Volume-First			Evaluation (Combination Diskfile)		
VOLUME ACL	SUBVOLUME ACL	DISKFILE PATTERN ACL	FIRST-ACL	FIRST-RULE	ALL
Y	Y	Y	Permit	Permit	Permit
Y	Y	N	Permit	Permit	Deny
Y	Y	NR	Permit	Permit	Normal
Y	N	Y	Permit	Permit	Deny
Y	N	N	Permit	Permit	Deny
Y	N	NR	Permit	Permit	Deny
Y	NM	Y	Permit	Permit	Deny
Y	NM	N	Permit	Permit	Deny
Y	NM	NR	Permit	Permit	Deny
Y	NR	Y	Permit	Permit	Permit
Y	NR	N	Permit	Permit	Deny
Y	NR	NR	Permit	Permit	Normal
N	Y	Y	Deny	Deny	Deny
N	Y	N	Deny	Deny	Deny
N	Y	NR	Deny	Deny	Deny
N	N	Y	Deny	Deny	Deny
N	N	N	Deny	Deny	Deny
N	N	NR	Deny	Deny	Deny
N	NM	Y	Deny	Deny	Deny
N	NM	N	Deny	Deny	Deny
N	NM	NR	Deny	Deny	Deny
N	NR	Y	Deny	Deny	Deny
N	NR	N	Deny	Deny	Deny
N	NR	NR	Deny	Deny	Deny
NM	Y	Y	Deny	Permit	Deny
NM	Y	N	Deny	Permit	Deny

Y - ACL evaluates a YES
N - ACL evaluates a NO
NR - No ACL exists (NORECORD)
NM - ACL contains no mention
Permit - Access permitted
Deny - Access Denied
Normal - Normal ACL evaluation for DISKFILE

Table B-4. CHECK-DISKFILE-PATTERN settings when value is MID and Direction Diskfile is Volume-First

Configuration Direction Diskfile: Volume-First			**Evaluation (Combination Diskfile)**		
VOLUME ACL	**SUBVOLUME ACL**	**DISKFILE PATTERN ACL**	**FIRST-ACL**	**FIRST-RULE**	**ALL**
NM	Y	NR	Deny	Permit	Deny
NM	Y	Y	Deny	Deny	Deny
NM	N	N	Deny	Deny	Deny
NM	N	NR	Deny	Deny	Deny
NM	N	Y	Deny	Permit	Deny
NM	NM	N	Deny	Deny	Deny
NM	NM	NR	Deny	Normal	Deny
NM	NM	Y	Deny	Permit	Deny
NM	NR	N	Deny	Deny	Deny
NM	NR	NR	Deny	Normal	Deny
NR	NR	Y	Permit	Permit	Permit
NR	Y	N	Permit	Permit	Deny
NR	Y	NR	Permit	Permit	Normal
NR	Y	Y	Deny	Deny	Deny
NR	N	N	Deny	Deny	Deny
NR	N	NR	Deny	Deny	Deny
NR	N	Y	Deny	Permit	Deny
NR	NM	N	Deny	Deny	Deny
NR	NM	NR	Deny	Normal	Deny
NR	NM	Y	Permit	Permit	Permit
NR	NR	N	Deny	Deny	Deny
NR	NR	NR	Normal	Normal	Normal

Y - ACL evaluates a YES
N - ACL evaluates a NO
NR - No ACL exists (NORECORD)
NM - ACL contains no mention
Permit - Access permitted
Deny - Access Denied
Normal - Normal ACL evaluation for DISKFILE

Index