



# Trusted Solaris Audit Administration

---

Sun Microsystems, Inc.  
901 San Antonio Road  
Mountain View, CA 94303-4900  
U.S.A.

Part Number 805-8057  
November 1999

Copyright 1999 Sun Microsystems, Inc. 901 San Antonio Road, Palo Alto, California 94303-4900 U.S.A. All rights reserved.

This product or document is protected by copyright and distributed under licenses restricting its use, copying, distribution, and decompilation. No part of this product or document may be reproduced in any form by any means without prior written authorization of Sun and its licensors, if any. Third-party software, including font technology, is copyrighted and licensed from Sun suppliers.

Parts of the product may be derived from Berkeley BSD systems, licensed from the University of California. UNIX is a registered trademark in the U.S. and other countries, exclusively licensed through X/Open Company, Ltd.

Sun, Sun Microsystems, the Sun logo, docs.sun.com, AnswerBook, AnswerBook2, Trusted Solaris, and Solaris are trademarks, registered trademarks, or service marks of Sun Microsystems, Inc. in the U.S. and other countries. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. in the U.S. and other countries. Products bearing SPARC trademarks are based upon an architecture developed by Sun Microsystems, Inc.

The OPEN LOOK and Sun™ Graphical User Interface was developed by Sun Microsystems, Inc. for its users and licensees. Sun acknowledges the pioneering efforts of Xerox in researching and developing the concept of visual or graphical user interfaces for the computer industry. Sun holds a non-exclusive license from Xerox to the Xerox Graphical User Interface, which license also covers Sun's licensees who implement OPEN LOOK GUIs and otherwise comply with Sun's written license agreements.

Federal Acquisitions: Commercial Software – Government Users Subject to Standard License Terms and Conditions

DOCUMENTATION IS PROVIDED "AS IS" AND ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT, ARE DISCLAIMED, EXCEPT TO THE EXTENT THAT SUCH DISCLAIMERS ARE HELD TO BE LEGALLY INVALID.

---

Copyright 1999 Sun Microsystems, Inc. 901 San Antonio Road, Palo Alto, Californie 94303-4900 Etats-Unis. Tous droits réservés.

Ce produit ou document est protégé par un copyright et distribué avec des licences qui en restreignent l'utilisation, la copie, la distribution, et la décompilation. Aucune partie de ce produit ou document ne peut être reproduite sous aucune forme, par quelque moyen que ce soit, sans l'autorisation préalable et écrite de Sun et de ses bailleurs de licence, s'il y en a. Le logiciel détenu par des tiers, et qui comprend la technologie relative aux polices de caractères, est protégé par un copyright et licencié par des fournisseurs de Sun.

Des parties de ce produit pourront être dérivées du système Berkeley BSD licenciés par l'Université de Californie. UNIX est une marque déposée aux Etats-Unis et dans d'autres pays et licenciée exclusivement par X/Open Company, Ltd.

Sun, Sun Microsystems, le logo Sun, docs.sun.com, AnswerBook, AnswerBook2, Trusted Solaris, et Solaris sont des marques de fabrique ou des marques déposées, ou marques de service, de Sun Microsystems, Inc. aux Etats-Unis et dans d'autres pays. Toutes les marques SPARC sont utilisées sous licence et sont des marques de fabrique ou des marques déposées de SPARC International, Inc. aux Etats-Unis et dans d'autres pays. Les produits portant les marques SPARC sont basés sur une architecture développée par Sun Microsystems, Inc.

L'interface d'utilisation graphique OPEN LOOK et Sun™ a été développée par Sun Microsystems, Inc. pour ses utilisateurs et licenciés. Sun reconnaît les efforts de pionniers de Xerox pour la recherche et le développement du concept des interfaces d'utilisation visuelle ou graphique pour l'industrie de l'informatique. Sun détient une licence non exclusive de Xerox sur l'interface d'utilisation graphique Xerox, cette licence couvrant également les licenciés de Sun qui mettent en place l'interface d'utilisation graphique OPEN LOOK et qui en outre se conforment aux licences écrites de Sun.

CETTE PUBLICATION EST FOURNIE "EN L'ETAT" ET AUCUNE GARANTIE, EXPRESSE OU IMPLICITE, N'EST ACCORDEE, Y COMPRIS DES GARANTIES CONCERNANT LA VALEUR MARCHANDE, L'APTITUDE DE LA PUBLICATION A REPONDRE A UNE UTILISATION PARTICULIERE, OU LE FAIT QU'ELLE NE SOIT PAS CONTREFAISANTE DE PRODUIT DE TIERS. CE DENI DE GARANTIE NE S'APPLIQUERAIT PAS, DANS LA MESURE OU IL SERAIT TENU JURIDIQUEMENT NUL ET NON AVENU.



# Contents

---

**Preface 11**

**1. Auditing Basics 15**

Auditing Overview 15

The Audit Mechanism 17

Audit Startup 17

Audit Classes and Events 18

Audit Records 20

Audit Flags 21

Definitions of Audit Flags 21

Audit Flag Syntax 22

Audit Storage 23

Auditing a Workstation 24

Auditing User Exceptions 26

Process Audit Characteristics 28

The audit\_data File 29

The Audit Daemon's Role 29

What Makes a Directory Suitable for Storing Audit Data 30

Keeping Audit Files Manageable 30

The audit\_warn Script 31

	Controlling Audit Costs	32
	Auditing Efficiently	33
	The auditconfig Command	34
	Setting Audit Policies	35
<b>2.</b>	<b>Auditing Setup</b>	<b>37</b>
	Planning Auditing at Your Site	37
	Planning What to Audit	38
	Planning a Site-Specific Event-to-Class Mapping	40
	Planning Space for Audit Records	41
	Planning the Rollout	44
	Rolling Out Auditing at Your Site	45
	System Administrator's Audit Setup Tasks	45
	Security Administrator's Audit Setup Tasks - Basic	46
	Audit Shutdown and Startup Procedures	47
	▼ To Disable Auditing	48
	▼ To Enable Auditing	49
	Basic Audit Setup Procedures	51
	▼ To Create Dedicated Audit Partitions	51
	▼ To Execute Commands that Require Privilege	52
	▼ To Remove Free Space (Optional)	53
	▼ To Protect an Audit File System	54
	▼ To Create an Audit Directory	56
	▼ To Share an Audit File System	57
	▼ To Mount an Audit File System	57
	▼ To Reserve Free Space on an Audit File System	58
	▼ To Specify the Audit File Storage Locations	59
	▼ To Set Audit Flags	60
	▼ To Set User Exceptions to the Audit Flags	61

▼ To Warn of Audit Trouble	62
▼ To Set Audit Policy Permanently	62
▼ To Distribute Audit Configuration Files to a Network of Workstations	63
▼ To Allocate and Deallocate Devices	65
Advanced Audit Setup Procedures	66
▼ To Add Audit Classes	66
▼ To Add Audit Events	67
▼ To Change Event-Class Mappings	68
▼ To Set Public Object Bit on Publicly Accessible Files	69
Dynamic Procedures	70
▼ To Determine Current Audit Policy	70
▼ To Create an Admin_High Workspace	71
▼ To Set Audit Policy Temporarily	71
▼ To Change Audit Flags Dynamically	73
▼ To Stop the Audit Daemon	74
▼ To Start the Audit Daemon	74
▼ To Send Audit Records to a New Audit File	75
<b>3. Audit Trail Management and Analysis</b>	<b>77</b>
The Audit Trail	77
How the Audit Trail Is Created	78
Audit Record Format	79
Order of Audit Tokens	79
Human-Readable Audit Record Format	80
Reading an Audit Token	80
Reading an Audit Record	81
Audit Files	82
Audit File Naming	83
Audit Files Management	85

	Merging the Audit Trail	85
	Selecting Records from the Audit Trail	85
	Using the <code>auditreduce</code> and <code>praudit</code> Commands	86
	Audit Files Backup and Recovery	93
	▼ To Back Up Audit Files	93
	▼ To Restore Audit Files	94
<b>4.</b>	<b>Troubleshooting Auditing</b>	<b>97</b>
	Preventing Audit Trail Overflow	97
	▼ To Prevent Audit Trail Overflow by Planning Ahead	98
	▼ To Handle an Audit Filesystem Overflow	99
	Cleaning up an Audit File Marked <code>not_terminated</code>	99
	▼ To Clean Up a <code>not_terminated</code> Audit File	100
	Using the <code>sequence</code> Token for Debugging	100
	▼ To Add the <code>sequence</code> Token to the Audit Record	101
	▼ To Prevent the <code>sequence</code> Token from Being Part of Audit Records	101
	Starting the Audit Daemon Manually	102
	Workstations are Being Audited Differently	102
	▼ To Set Audit Class Mappings for Attributable Events	103
	▼ To Set Audit Class Mappings for Non-Attributable Audit Events	103
	Finding Failed Login Attempts	103
<b>A.</b>	<b>Event-to-Class Mappings</b>	<b>105</b>
	Audit Events Listed by Audit Class	105
	Events in Audit Class <code>aa</code>	108
	Events in Audit Class <code>ao</code>	110
	Events in Audit Class <code>ap</code>	114
	Events in Audit Class <code>cl</code>	114
	Events in Audit Class <code>fa</code>	115
	Events in Audit Class <code>fc</code>	117

Events in Audit Class fd	118
Events in Audit Class fm	119
Events in Audit Class fn	121
Events in Audit Class fr	121
Events in Audit Class fw	122
Events in Audit Class io	123
Events in Audit Class ip	124
Events in Audit Class lo	126
Events in Audit Class na	127
Events in Audit Class no	128
Events in Audit Class nt	130
Events in Audit Class ot	131
Events in Audit Class pm	131
Events in Audit Class ps	133
Events in Audit Class as	134
Events in Audit Class ss	136
Events in Audit Class ax	137
Events in Audit Class xa	137
Events in Audit Class xc	138
Events in Audit Class xl	139
Events in Audit Class xp	140
Events in Audit Class xs	143

## **B. Audit Record Descriptions 147**

Audit Record Structure 147

Audit Token Structure 148

acl Token 151

arbitrary Token 151

arg Token 153

attr Token 153  
clearance Token 154  
exec\_args Token 154  
exec\_env Token 155  
exit Token 156  
file Token 156  
groups Token (Obsolete) 157  
header Token 157  
host Token 158  
in\_addr Token 159  
ip Token 159  
ipc Token 160  
ipc\_perm Token 161  
iport Token 162  
liaison Token 162  
newgroups Token 162  
opaque Token 163  
path Token 164  
priv Token 164  
privilege Token 165  
process Token 166  
return Token 167  
seq Token 167  
slabel Token 168  
socket Token 168  
socket-inet Token 169  
subject Token 169  
text Token 170



trailer Token	171
xatom Token	171
xclient Token	172
xcolormap Token	172
xcursor Token	173
xfont Token	173
xgc Token	173
xpixmap Token	174
xproperty Token	174
xselect Token	174
xwindow Token	175
Audit Records	175
General Audit Record Structure	175
Kernel-Level Generated Audit Records	176
Kernel-Level Pseudo-Events	257
X Server Protocol Audit Records	257
User-Level Generated Audit Records	280
<b>C. Audit Reference</b>	<b>315</b>
<b>Index</b>	<b>323</b>



# Preface

---

Auditing is a security feature required for a C2 rating in TCSEC. C2 discretionary-access control and identification and authentication features are provided by the standard Solaris system. The Trusted Solaris operating environment is in evaluation for a B1+ Trusted Computer System Evaluation Criteria (TCSEC) evaluation from the U.S. National Security Agency, and has earned an ITSEC evaluation in the United Kingdom of assurance level E3 and functionality F-B1.

---

## Who Should Use This Book

*Trusted Solaris Audit Administration* is intended for the system administrator whose duties include setting up and maintaining auditing file systems, and for the security administrator whose duties include determining what will be audited and analyzing the auditing trail. The system administrator should be familiar with file system administration, such as NFS-mounting, sharing directories, exporting directories, and creating disk partitions. The security administrator should be familiar with the site security policy, and with the help of the system administrator, be able to create and modify shell scripts.

---

## How This Book Is Organized

Chapter 1, explains the system management and configuration of the auditing subsystem. Topics discussed include managing audit trail storage, determining global and per-user preselection, and setting site-specific configuration options.

Chapter 2, covers setting up and maintaining auditing at your site. The latter part of the chapter contains procedures for setting up and maintaining auditing.

Chapter 3, describes how the audit daemon creates the audit trail, and how to manage audit files and read the contents. The latter part of the chapter contains procedures for merging audit files, selecting records, reading the audit trail, and backing up the trail.

Chapter 4, contains procedures for troubleshooting the auditing subsystem.

Appendix A, lists audit events by their default audit class and alphabetically. It also connects them to their system calls and user commands.

Appendix B, describes in detail the content of the audit records generated, including a description of every audit token.

Appendix C lists and describes the man pages added for the auditing subsystem in the Trusted Solaris 7 environment, and file protections on the auditing subsystem.

---

## Related Books

All sites should have the following books or information available when setting up auditing:

### From Sun Microsystems

- *Trusted Solaris 7 Release Notes*

Describes any late-breaking news about auditing, including known problems.

- *Trusted Solaris Administrator's Procedures*

Describes administration tasks, such as assuming a role, in detail.

### From Elsewhere

- *Your site security policy*

Describes the security policy and security procedures at your site.

Other books on auditing that might be of interest include:

- *A Guide to Understanding Audit in Trusted Systems*

- *Auditing in a UNIX System*

- *DoD Trusted Computer System Evaluation Criteria (the Orange Book)*

- *Compartmented Mode Workstation Evaluation Criteria*
- *Guideline for Trusted Facility Management and Audit*, Virgil D. Gligor, 1985

---

## Ordering Sun Documents

The Sun Software Shop stocks select manuals from Sun Microsystems, Inc. You can purchase individual printed manuals and AnswerBook2™ CDs.

For a list of documents and how to order them, visit the Software Shop at <http://www.sun.com/software/shop/>.

---

## Accessing Sun Documentation Online

The docs.sun.com<sup>SM</sup> Web site enables you to access Sun technical documentation online. You can browse the docs.sun.com archive or search for a specific book title or subject. The URL is <http://docs.sun.com>.

---

## Typographic Conventions

The following table describes the typographic conventions used in this book.

TABLE P-1 Typographic Conventions

Typeface or Symbol	Meaning	Example
AaBbCc123	The names of commands, files, and directories; on-screen computer output	Edit your <code>.login</code> file. Use <code>ls -a</code> to list all files. <code>machine_name%</code> you have mail.
<b>AaBbCc123</b>	What you type, contrasted with on-screen computer output	<code>machine_name%</code> <b>su</b> Password:

TABLE P-1 Typographic Conventions (continued)

Typeface or Symbol	Meaning	Example
<i>AaBbCc123</i>	Command-line placeholder: replace with a real name or value	To delete a file, type <b>rm</b> <i>filename</i> .
<i>AaBbCc123</i>	Book titles, new words, or terms, or words to be emphasized.	Read Chapter 6 in <i>User's Guide</i> . These are called <i>class</i> options. You must be <i>root</i> to do this.
<Do this>	Used in examples: follow the instruction in the brackets.	<b>praudit -d</b> '<press Tab key>'

## Shell Prompts in Command Examples

The following table shows the default system prompt and administrative role prompts for the C shell, Bourne shell, and Korn shell.

TABLE P-2 Shell Prompts

Shell	Prompt
C shell prompt	machine_name%
root role prompt	#
Bourne shell, Korn shell prompt, secadmin role, admin role prompt	\$

# Auditing Basics

---

This chapter explains how auditing works on one workstation and on a network of Trusted Solaris workstations.

- “Auditing Overview” on page 15
- “The Audit Mechanism” on page 17

Auditing makes it possible to:

- Monitor security-relevant events that take place on a workstation
- Record the events in a network-wide audit trail
- Detect misuse or unauthorized activity (by analyzing the audit trail)
- Review patterns of access, and see the access histories of individuals and objects
- Discover attempts to bypass the protection mechanisms
- Discover extended use of privilege that occurs when a user assumes an administrative role
- Supply additional assurance that attempts to bypass protection mechanisms are recorded and discovered

Auditing may serve as a deterrent: if users know that their actions are likely to be audited, they may be less likely to attempt malicious activities.

---

## Auditing Overview

Auditing is included in the full release and is part of the release media of the Trusted Solaris operating environment. By default, auditing is enabled. It can be disabled by modifying two files. All of the auditing software is included in the initial system installation in the following packages:

- SUNWcar – Core architecture
- SUNWcsr – Core SPARC
- SUNWcsu – Core SPARC
- SUNWtsr – Trusted Solaris policy
- SUNWtsu – Trusted Solaris policy
- SUNWhea – Header files
- SUNWman – On-line manual pages

Auditing in Trusted Solaris is enabled by default, configurable by the system and security administrators, and extensible. By default, audit records are stored in *workstation\_name*: /var/audit/. Events in the audit classes *login\_logout* and *non-attribute* are audited for the root user.

The system administrator can provide dedicated partitions for audit records. The audit analyst can collect all records from all workstations in a Trusted Solaris network into one audit trail. The auditing records from a network of workstations can be viewed as one large file. Record selection using a variety of criteria is possible.

After audit data is collected into one audit trail, selection (called *post-selection*) and interpretation tools enable the audit reviewer to examine specific parts of the audit trail. For example, records can be selected for individual users or groups, for a host name, for a certain type of event on a specific day, or for a time of day.

To simplify audit administration, Trusted Solaris auditing provides classes of auditable events. When the security administrator specifies a class of events to be audited, all events in that class are audited. User commands or kernel system calls are auditable events. Classes of events to be audited can be specified per workstation. Specific users or roles (like *root*, for example) can be audited specially.

The security administrator can modify and extend the provided event - class mappings. For some events, event details that are not required by site security policy can be omitted from the audit record. Audit classes can be audited for failure, for success, or for both per workstation. Selecting which activities to monitor is called *pre-selection*. When the auditing subsystem encounters error conditions, the security administrator can specify whom to email with the information.

In the Trusted Solaris auditing subsystem, audit records are protected from snooping by the sensitivity label *admin\_high*. Audit configuration files are accessible by the appropriate administrative role only, and sending records to the audit queue requires privilege. A special privilege, *proc\_audit\_appl*, is provided for ISVs and integrators to add their applications' audit records to the audit queue. Audit event numbers from 32768 to 65535 are available for third-party trusted applications.

Successful auditing depends on two other security features: identification and authentication. At login, after a user supplies a user name and password, a unique audit ID is associated with the user's process. The audit ID is inherited by every process started during the login session. Even when users change identity, for



example, by assuming an administrative role, all of their actions are tracked with the same audit ID.

The rest of this chapter describes the auditing subsystem. Chapter 2 describes how to set up and administer auditing. The latter part of the chapter contains setup and maintenance procedures. Chapter 3, describes the audit trail, how to manage its files, and how to read them. The latter part of the chapter contains typical procedures for managing and analyzing the audit trail.

---

## The Audit Mechanism

Auditing is enabled by an audit daemon that uses six configurable audit files: `audit_class(4)`, `audit_event(4)`, `audit_control(4)`, `audit_user(4)`, `audit_startup(1M)`, and `audit_warn(1M)`. These files are in the `/etc/security` directory and determine what to audit, where to put the audit logs, and what to do when there is trouble. By default, events in the `lo` (login/logout) audit class are audited for the root role, the audit records are written to the `/var/audit` directory, and no one receives mail when there is trouble.

You can suspend and re-enable auditing without rebooting the workstation, and you can dynamically change what is being audited.

## Audit Startup

Auditing is enabled when the audit daemon starts, usually when the workstation is booted (see the `auditd(1M)` man page). When troubleshooting, the daemon can be started manually by executing `/usr/sbin/auditd` in an `admin_high` shell in the `secadmin` role.

The existence of a file with the path name `/etc/security/audit_startup` causes the audit daemon to be run automatically when the system enters multiuser mode. The file is actually an executable script that is invoked as part of the startup sequence just prior to the execution of the audit daemon (see the `audit_startup(1M)` man page). A default `audit_startup` script that automatically configures the event-to-class mappings and sets the audit policies is created during audit package installation.

The security administrator can edit the `audit_startup` script to alter the default audit policy. See “Setting Audit Policies” on page 35 for more information on audit policy.

# Audit Classes and Events

Security-relevant actions may be audited. The system actions that are auditable are defined as *audit events* in the `/etc/security/audit_event` file. Each auditable event is defined in the file by a symbolic name, an event number, a set of preselection classes, and a short description (see the `audit_event(4)` man page).

Most events are attributable to an individual user. However, some events are *nonattributable* because they occur at the kernel-interrupt level or before a user is identified and authenticated. Nonattributable events are auditable as well.

Each audit event is also defined as belonging to an audit class or classes. Administrators name an audit class (called an *audit flag*) when specifying for the audit daemon what is to be audited. When naming a class, one simultaneously addresses all of the events in that class. The mapping of audit events to classes is configurable and the classes themselves are configurable. These configuration changes are made in the `audit_event` file; new classes are added to the `audit_class` file.

Whether an auditable event is recorded in the audit trail depends on whether the administrator preselects an audit class that includes the specific event.

## Audit Classes

The file `/etc/security/audit_class` stores class definitions. Site-specific definitions can be added and default definitions can be changed. Each entry in the file has the form:

*mask:name:description*

Each class is represented as a bit in the mask, which is an unsigned integer, giving 32 different available classes plus two global classes, `all` and `no`. `all` is a conjunction of all allowed classes; `no` is the invalid class. Events mapped to the `no` class are not audited. Events mapped solely to the `no` class are not audited even if the `all` class is turned on. Below is a sample `audit_class` file:

```
0x00000000:no:invalid class
0x00000001:fr:file read
0x00000002:fw:file write
0x00000004:fa:file attribute access
0x00000008:fm:file attribute modify
0x00000010:fc:file create
0x00000020:fd:file delete
0x00000040:cl:file close
0x00000100:nt:network
0x00000200:ip:ipc
0x00000400:na:non-attribute
0x00001000:lo:login or logout
0x00002000:ax:x server
0x00004000:ap:application
0x000f0000:ad:administrative
```

(continued)

```

0x00010000:ss:change system state
0x00020000:as:system-wide administration
0x00040000:aa:audit administration
0x00080000:ao:other administration
0x00300000:pc:process
0x00100000:ps:process start/stop
0x00200000:pm:process modify
0x20000000:io:ioctl
0x40000000:fn:fcntl
0x80000000:ot:other
0xffffffff:all:all classes

```

If the `no` class is actually turned on for auditing, the audit trail fills up with records for the audit event `AUE_NULL`.

## Kernel Events

Events generated by the kernel (system calls) have event numbers between 1 and 2047. The event names for kernel events begin with `AUE_`, followed by an uppercase mnemonic for the event. For example, the event number for the `creat()` system call is 4 and the event name is `AUE_CREAT`.

Within kernel events there is one pseudo-event defined, `AUE_UPRIV`, which audits use-of-privilege decisions.

When the `AUE_UPRIV` pseudo-event is preselected, audit information is collected internally *even if* the underlying kernel event is not selected. For example, if the kernel event `AUE_OPEN_R` is not selected for auditing but the pseudo-event `AUE_UPRIV` is enabled, the kernel event `AUE_OPEN_R` will be written to the audit trail if a use-of-privilege decision was part of the system call `AUE_OPEN_R`.

## User-Level Events

Events generated by trusted application software outside the kernel range from 2048 to 65535. The event names begin with `AUE_`, followed by a lowercase mnemonic for the event. The file `/etc/security/audit_event` lists individual events in numerical order. For a listing of events by class, see Appendix A. The following table shows general categories of user-related events.

TABLE 1-1 Audit Event Categories

Number Range	Type of Event
2048-65535	User-level audit events
2048-32767	Reserved for Solaris and Trusted Solaris user-level programs
32768-65536	Available for third-party applications

## Non-Attribute Events

Events that are not attributable to a user, such as AUE\_ENTERPROM.

## Audit Records

Each audit record describes the occurrence of a single audited event and includes such information as who did the action, which files were affected, what action was attempted, and where and when it occurred.

The type of information saved for each audit event is defined as a set of *audit tokens*. Each time an audit record is created for an event, the record contains some or all of the tokens defined for it, depending on the nature of the event and the audit policy. The audit record descriptions in Appendix B list all the audit tokens defined for each event and what each token means.

Audit tokens construct audit records in an audit file. An audit trail is one or more audit files in a distributed system. The construction of the audit trail is shown in Figure 1-1. The audit trail may be converted to a human readable format by `praudit` (see the `praudit(1M)` man page). Specific audit records can be selectively chosen using the `auditreduce(1M)` command. See Chapter 3, for details.

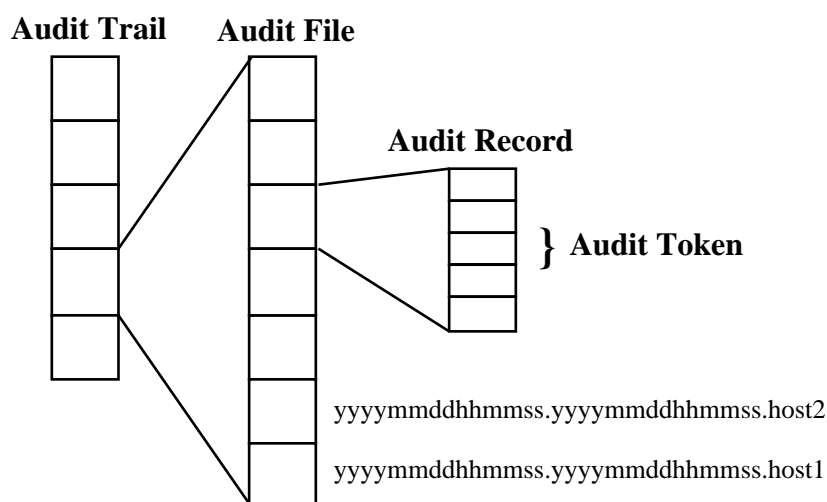


Figure 1-1 From the Audit Token to the Audit Trail

## Audit Flags

Audit flags are the short names for the audit classes. Audit flags are used to indicate which classes to audit in the `audit_control(4)` file, the `audit_user(4)` file, and as arguments to the `auditconfig(1M)` command.

The `audit_control` file is described in “Auditing a Workstation” on page 24. The `audit_user` file is described in “The `audit_user` File” on page 26.

## Definitions of Audit Flags

Each predefined audit class is listed in Table A-1. The table includes the audit flag (which is the short name that stands for the class), the long name, its audit mask, and a pointer to the list of audit events that by default are in that audit class. The system administrator uses the audit flags in the auditing configuration files to specify which classes of events to audit. Additional classes can be defined and existing classes can be renamed by modifying the `audit_class(4)` file.

# Audit Flag Syntax

Depending on the prefixes, a class of events can be audited whether it succeeds or fails, or only if it succeeds or only if it fails. The format of the audit flag is shown here.

```
prefixflag
-lo      # audit for failure
+lo      # audit for success
lo       # audit for success and failure
```

The audit flag `+lo` means “all successful attempts to log in and log out”. The audit flag `-lo` means “all failed attempts to log in”. (You cannot fail an attempt to logout.). The audit flag `lo` means “all successful attempts to log in and log out and all failed attempts to log in”.

---

**Note** - The audit class `xs` should not be audited for failure; failures will place a lot of noise in the audit trail. The correct audit flag syntax would be `+xs`. See the `audit_class(4)` file for more information on X server audit classes.

---

For another example, the `+all` flag refers to all successful attempts of any kind.



---

**Caution** - The `all` flag can generate large amounts of data and fill up audit file systems quickly, so use it only if you have extraordinary reasons to audit everything

---

The following table shows prefixes that specify whether the audit class is audited for success or failure or both.

TABLE 1-2 Prefixes Used in Audit Flags

Prefix	Definition
none	Audit for both success and failure
+	Audit for success only
-	Audit for failure only

## Prefixes to Modify Previously Set Audit Flags

Use the modify prefixes in any of three ways: in the flags line in the `audit_control(4)` file to modify already-specified flags, as flags in the user's entry in the `audit_user(4)` file, or as arguments to the `auditconfig(1M)` command.

The prefixes in Table 1-3 along with audit flags, turn on or turn off previously specified audit classes. These prefixes turn on or off previously specified flags only.

**TABLE 1-3** Prefixes Used to Modify Already-Specified Audit Flags

Prefix	Definition
<code>^-</code>	Turn off for failed attempts
<code>^+</code>	Turn off for successful attempts
<code>^</code>	Turn off for both failed and successful attempts

The `^-` prefix is used in the flags line in the following example from an `audit_control` file.

```
flags:lo,ad,-all,^-fc
```

## Audit Storage

On every workstation, the `/etc/security/audit` directory contains subdirectories with all the audit log files. The `/etc/security` directory contains files related to audit configuration. Because the `/etc/security` directory contains the per-workstation `audit_data` file, which is used by the audit daemon at boot time, the `/etc/security` directory must be part of the root file system.

The audit postselection tools look in directories under `/etc/security/audit` by default. For this reason, the path name of the mount point for the first audit file system on an audit server is in the form: `/etc/security/audit/server-name` (where *server-name* is the name of the audit server). If more than one audit partition is on an audit server, the name of the second mount point is:

`/etc/security/audit/server-name.1`, the third is  
`/etc/security/audit/server-name.2`, and so forth.

For example, the names of the audit file systems available on the audit file server `audubon` are `/etc/security/audit/audubon` and `/etc/security/audit/audubon.1`.

Each audit file system has a subdirectory named `files`. This `files` subdirectory is where the audit files are located and where the `auditreduce` commands looks for

them. For example, the audit file system on audit server audubon has a files subdirectory whose full path name is: `/etc/security/audit/audubon/files`.

The local `audit_control` file on each workstation directs the audit daemon to put the audit files in the files subdirectory. For example, the `dir:` line for the `audit_control` file on a workstation mounting the audit file system from eagle is:

```
dir: /etc/security/audit/eagle/files
```

The extra level of hierarchy prevents a workstation's local root file system from filling with audit files when (for whatever reason) the `/etc/security/audit/server-name[.suffix]` directory is not available on the audit server. Because the files subdirectory is present on the audit server and the clients use the same naming convention for their local audit log files, `/etc/security/audit/client-name`, audit files cannot be created unintentionally in the local mount-point directory if the mount fails.

## Permissions on Audit Directories

Audit directory permissions on the `/etc/security/audit/workstation-name` directory and the files directory directly beneath it are shown in the following table.

TABLE 1-4 Audit Directory and File Permissions

Owner	Group	Permissions
audit	audit	750

## Auditing a Workstation

Auditing is set per workstation by the security administrator in the file `audit_control`. This file on each workstation is read by the audit daemon (see the `audit_control(4)` man page). The `audit_control` file is located in the `/etc/security` directory.

A separate `audit_control` file is maintained on each workstation because the `dir:` lines, and perhaps the `minfree:` line are specific to the workstation. In a distributed system, the other lines should be identical.



You specify four kinds of information in four kinds of lines in the `audit_control` file:

- The *audit flags* line (`flags:`) contains the audit flags that define what classes of events are audited for all users on the workstation. The audit flags specified here are referred to as the *machine-wide audit flags* or the *machine-wide audit preselection mask*. Audit flags are separated by commas, with no spaces.
- The *nonattributable flags* line (`naflags:`) contains the audit flags that define what classes of events are audited when an action cannot be attributed to a specific user. The flags are separated by commas, with no spaces.
- The *audit threshold* line (`minfree:`) defines the minimum free-space level for all audit file systems. See “What Makes a Directory Suitable for Storing Audit Data” on page 30.

The `minfree` percentage must be greater than or equal to 0. The default is 20 percent.

- The *directory definition* lines (`dir:`) define which audit file systems and directories the workstation will use to store its audit trail files.

There may be one or more directory definition lines. The order of the `dir:` lines is significant, because the `auditd` command opens audit files in the directories in the order specified (see the `audit(1M)` man page). The first audit directory specified is the primary audit directory for the workstation, the second is the secondary audit directory where the audit daemon puts audit trail files when the first one fills, and so forth.

The security administrator modifies the default `audit_control` file during the configuration process on each workstation.

After the `audit_control` file is configured, the security administrator on a distributed system distributes it to the other workstations. After any change in the file, the administrator runs `audit -s` on every workstation on the network to instruct the audit daemon to reread its `audit_control` file.

---

**Note** - The `audit -s` command does not change the preselection mask for existing processes. Use `auditconfig`, `setaudit` (see the `getaudit(2)` man page), or `auditon(2)` for existing processes.

---

## Sample audit\_control File

Following is a sample `audit_control` file for the workstation `willet`. `willet` uses two audit file systems on the audit server `egret`, and a third audit file system mounted from the audit administration server `audubon`, which is used to store audit records only when the audit file system on `egret` fills up or is unavailable. The `minfree` value of 20 percent specifies that the warning script (see the `audit_warn(1M)` man page) is run when the file systems are 80 percent filled and the audit data for the current workstation will be stored in the next available audit

directory, if any. The flags specify that all logins and administrative operations are to be audited (whether or not they succeed), and that failures of all types except failures to create a file system object are to be audited.

```
flags:lo,ad,-all,^-fc
naflags:lo,nt
minfree:20
dir:/etc/security/audit/egret/files
dir:/etc/security/audit/egret.1/files
#
# Audit filesystem used when egret fills up
#
dir:/etc/security/audit/audubon
```

## Auditing User Exceptions

The security administrator sets up auditing for the default configuration. You may want all users and administrators to be audited according to the system-wide audit flags you specified in the `audit_control` file. To fine-tune auditing for individual users, you add user entries to the `audit_user` file. You may also choose to add audit flags to users' entries at the time you add new users, and you should probably set up auditing for the new user just after you unlock the account and configure the security attributes for that user.

---

**Note** - Alterations to a static auditing database (`audit_control`, `audit_user`, `audit_startup`, or `audit_warn`) on one workstation should be copied to all workstations on the network. See "To Distribute Audit Configuration Files to a Network of Workstations" on page 63.

---

In addition to supplying the per-user audit control information in the static databases, you can dynamically adjust the state of auditing while a user's processes are active on a single workstation.

### The `audit_user` File

If it is desirable to audit some users differently from others, the administrator can edit the `audit_user` file to add audit flags for individual users. If specified, these flags are combined with the system-wide flags specified in the audit control file to determine which classes of events to audit for that user. The flags the administrator adds to the user's entry in the `audit_user` file modify the defaults from the `audit_control` file in two ways: by specifying a set of event classes that are never to be audited for this user or by specifying a set of event classes that are always to be audited.

So, what is audited for an individual user is the combination of the workstation audit flags and the user's always and never audit flags, as shown below.

What is audited = (Workstation Audit Flags + User's Always Audit) - User's Never Audit

In the `audit_user` file entry for each user, there are three fields. The first field is the *username*, the second field is the *always-audit* field, the third is the *never-audit* field.

The two auditing fields are processed in sequence, so auditing is enabled by the first field and turned off by the second.

---

**Note** - Avoid the common mistake of leaving the `all` set in the *never-audit* field. This causes all auditing to be turned off for that user, overriding the flags set in the *always-audit* field.

---

Using the *never-audit* flags for a user is not the same as removing classes from the *always-audit* set. For example, suppose (as shown in the examples below), you have a user `katya` for whom you want to audit everything except successful reads of file system objects. (This is a good way to audit almost everything for a user while generating only about three-quarters of the audit data that would be produced if all data reads were also audited.) You also want to apply the system defaults to `katya`. Here are two possible `audit_user` entries.

The correct entry

```
katya:all,^+fr:
```

The incorrect entry:

```
katya:all:+fr
```

The first example says, "always audit everything except successful file-reads." The second example says "always audit everything, but never audit successful file-reads." The second example is incorrect because it overrides the system default. The first example achieves the desired effect: any earlier default applies, as well as what is specified in the `audit_user` entry.

---

**Note** - Successful events and failed events are treated separately, so a process can (for example) generate more audit records when an error occurs than when the event is successful.

---

Dynamic controls refer to controls put in place by the administrator while processes are running. These persist only while the affected processes (and any of their children) exist, but will not continue in effect at the next login. Dynamic controls apply to one workstation at a time, since the audit command only applies to the current workstation where you are logged in.

Each process has two sets of one-bit flags for audit classes. One set controls whether the process is audited when an event in the class is requested successfully; the other set, when an event is requested but fails (for any reason). It is common for processes to be more heavily audited for failures than for successes, since this can be used to detect attempts at browsing and other types of attempts at violating system security.

## Process Audit Characteristics

The following audit characteristics are set at initial login:

- Process preselection mask
- Audit ID (AUID)
- Audit Session ID
- Terminal ID (port ID, workstation ID)

### Process Preselection Mask

When a user logs in, `login` combines the workstation-wide audit flags from the `audit_control` file with the user-specific audit flags (if any) from the `audit_user` file, to establish the *process preselection mask* for the user's processes. The process preselection mask specifies whether events in each audit event class are to generate audit records.

The algorithm for obtaining the process preselection mask is as follows: the audit flags from the `flags:` line in the `audit_control` file are added to the flags from the *always-audit* field in the user's entry in the `audit_user` file. The flags from the *never-audit* field from the user's entry in the `audit_user` file are then subtracted from the total.

*user's process preselection mask = (flags: line + always audit flags) - never audit flags*

### Audit ID

A process also acquires its audit ID when the user logs in, and this audit ID is inherited by all child processes started by the user's initial process. The audit ID helps enforce accountability. Even after a user assumes a role, the audit ID remains the same. The audit ID that is saved in each audit record allows the administrator to always trace actions back to the original user that logged in.

## Audit Session ID

The audit session ID is assigned at login and inherited by all descendant processes.

## Terminal ID

The terminal ID consists of the host name and the Internet address, followed by a unique number that identifies the physical device on which the user logged in. Most of the time the login will be through the console and the number that corresponds to the console device will be 0.

## The audit\_data File

When `auditd` starts on each workstation, it creates the file `/etc/security/audit_data`. The format of the file consists of a single entry with the two fields separated by a colon (see the `audit_data(4)` man page). The first field is the audit daemon's process ID, and the second field is the path name of the audit file to which the audit daemon is currently writing audit records. Here is an example:

```
# cat /etc/security/audit_data
116:/etc/security/audit/egret.1/files/19910320100002.not_terminated.tern
```

## The Audit Daemon's Role

The following list summarizes what the audit daemon, `auditd(1M)`, does.

- `auditd` opens and closes audit log files in the directories specified in the `audit_control` file in the order in which they are specified.
- `auditd` reads audit data from the kernel and writes it to an audit file.
- `auditd` executes the `audit_warn` script when the audit directories fill past limits specified in the `audit_control` file. The script, by default, sends warnings to the `audit_warn` alias and to the console. Your site should customize `audit_warn` to suit your needs. The `audit_warn` script is described in “The `audit_warn` Script” on page 31.
- With the system default configuration, when all audit directories are full, processes that generate audit records are suspended and `auditd` writes a message to the console and to the `audit_warn` alias. (The auditing policy can be reconfigured

with the `auditconfig` command.) At this point only the system administrator could log in to write audit files to tape, delete audit files from the system, or do other cleanup.

When the audit daemon starts as the workstation is brought up to multiuser mode, or when the audit daemon is instructed by the `audit -s` command to reread the file after the file has been edited, `auditd` determines the amount of free space necessary and reads the list of directories from the `audit_control` file and uses those as possible locations for creating audit files.

The audit daemon maintains a pointer into this list of directories, starting with the first. Every time the audit daemon needs to create an audit file, it puts the file into the first available directory in the list, starting at the audit daemon's current pointer.

## What Makes a Directory Suitable for Storing Audit Data

A directory is *suitable* for storing audit records if it is accessible to the audit daemon, which means that it must be mounted, that the network connection (if remote) permits successful access, and that the permissions on the directory allow access. Also in order for a directory to be suitable for audit files, it must have sufficient free space remaining. You can edit the `minfree:` line in the `audit_control` file to change the default of 20 percent. To give an example of how the `minfree` percentage is applied, if the default minimum free space of 20 percent is accepted, an email notice is sent to the `audit_warn` alias whenever a file system becomes more than 80 percent full.

When no directories on the list have enough free space left, the daemon starts over from the beginning of the list and picks the first accessible directory that has any space available until the hard limit is reached. In the default configuration, if no directories are suitable, the daemon stops processing audit records, and they accumulate within the kernel until all processes generating audit records are suspended.

## Keeping Audit Files Manageable

To keep audit files at a manageable size, a `cron` job can be set up that periodically switches audit files (see the `cron(1M)` man page). Intervals might range from once per hour to twice per day, depending on the amount of audit data being collected. The data can then be filtered to remove unnecessary information and then compressed.

---

# The audit\_warn Script

Whenever the audit daemon encounters an unusual condition while writing audit records, it invokes the `/etc/security/audit_warn` script. See the `audit_warn(1M)` man page. This script can be customized by your site to warn of conditions that might require manual intervention or to handle them automatically. For all error conditions `audit_warn` writes a message to the console and sends a message to the `audit_warn` alias. This alias should be set up by the administrator after enabling auditing.

When the following conditions are detected by the audit daemon, it invokes `audit_warn`.

- An audit directory has become more full than the `minfree` value allows. (The `minfree` or soft limit is a percentage of the space available on an audit file system.)

The `audit_warn` script is invoked with the string `soft` and the name of the directory whose space available has gone below the minimum. The audit daemon switches automatically to the next suitable directory, and writes the audit files there until this new directory reaches its `minfree` limit. The audit daemon then goes to each of the remaining directories in the order listed in `audit_control`, and writes audit records until each is at its `minfree` limit.

- All the audit directories are more full than the `minfree` threshold.

The `audit_warn` script is invoked with the string `allsoft` as an argument. A message is written to the console and mail is sent to the `audit_warn` alias.

When all audit directories listed in `audit_control` are at their `minfree` limits, the audit daemon switches back to the first one, and writes audit records until the directory completely fills.

- An audit directory has become completely full with no space remaining.

The `audit_warn` script is invoked with the string `hard` and the name of the directory as arguments. A message is written to the console and mail is sent to the `audit_warn` alias.

The audit daemon switches automatically to the next suitable directory with any space available, if any. The audit daemon goes to each of the remaining directories in the order listed in `audit_control`, and writes audit records until each is full.

- All the audit directories are completely full. The `audit_warn` script is invoked with the string `allhard` as an argument.

In the default configuration, a message is written to the console and mail is sent to the `audit_warn` alias. The processes generating audit records are suspended. The audit daemon goes into a loop waiting for space to become available and resumes processing audit records when that happens. While audit records are not being

processed, no auditable activities take place—every process that attempts to generate an audit record is suspended.

- An internal error occurs: another audit daemon process is already running (string `ebusy`), a temporary file cannot be used (string `tmpfile`), the `auditsvc(2)` system call fails (string `auditsvc`), or a signal was received during auditing shutdown (string `postsigterm`).

Mail is sent to the `audit_warn` alias.

- A problem is discovered with the `audit_control` file's contents. By default, mail is sent to the `audit_warn` alias and a message is sent to the console.

---

## Controlling Audit Costs

Because auditing consumes system resources, you must control the degree of detail that is recorded. When you decide what to audit, consider the following three costs of auditing:

- Costs in increased processing time
- Costs of analysis of audit data
- Costs of storage of audit data

The cost in increased processing time is the least significant of the three costs of auditing. The first reason is that auditing generally does not occur during computational-intensive tasks—image processing, complex calculations, and so forth. The other reason that processing cost is usually insignificant is that single-user workstations have plenty of extra CPU cycles.

The cost of analysis is roughly proportional to the amount of audit data collected. The cost of analysis includes the time it takes to merge and review audit records, and the time it takes to archive them and keep them in a safe place.

The fewer records you generate the less time it takes to analyze them, so upcoming sections describe how you can reduce the amount of data collected, while still providing enough coverage to achieve your site's security goals.

Storage cost is the most significant cost of auditing. The amount of audit data depends on the following:

- Number of users
- Number of workstations
- Amount of use
- Degree of security required

Because the factors vary from one situation to the next, no formula can determine in advance the amount of disk space to set aside for audit data storage.



Full auditing (with the `all` flag) can fill up a disk quickly. Even a simple task like compiling a program of modest size (for example, 5 files, 5000 lines total) in less than a minute could generate thousands of audit records, occupying many megabytes of disk space. Therefore, it is very important to use the preselection features to reduce the volume of records generated. For example, not auditing the `fr` class can reduce the audit volume by more than two-thirds. Efficient audit file management is also important after the audit records are created to reduce the amount of storage required.

---

## Auditing Efficiently

What to audit, when to audit it, and where to store the files are factors to consider when enforcing your site's security goals while auditing more efficiently. For example, you might try:

- Random auditing of only a certain percentage of users at any one time.
- Real-time monitoring of the audit data for unusual behaviors. (You set up procedures to monitor the audit trail as it is generated for certain activities and to trigger higher levels of auditing of particular users or workstations when suspicious events occur.) See "To Read a Current Audit File" on page 87 for an example.
- Setting the public object flag on publicly accessible files or directories. This reduces the potential size of the audit trail while not compromising security, because the viewing of publicly accessible files and directories is not generally interesting for audit purposes. Files so marked do not generate audit records for the following audit events, even if the classes for those events are turned on for auditing:

AUE\_ACCESS, AUE\_STAT, AUE\_LSTAT, AUE\_READLINK, AUE\_STATFS,  
AUE\_FSTATFS, AUE\_PATHCONF, AUE\_OPEN\_R, AUE\_FGETCMWLABEL,  
AUE\_GETCMWFSRANGE, AUE\_GETCMWLABEL, AUE\_GETFILEPRIV,  
AUE\_LGETCMWLABEL, AUE\_GETMLDADORN, AUE\_GETSLDNAME,  
AUE\_OSTAT, AUE\_FUSERS, AUE\_STATVFS, AUE\_XSTAT, and AUE\_LXSTAT.  
The list may not be exhaustive.

See "To Set Public Object Bit on Publicly Accessible Files" on page 69 for the procedure.

- Reducing the disk-storage requirements for audit files by combining, reducing, and compressing them (see "To Combine Selected Audit Files" on page 90), and developing procedures for storing them offline.

---

# The auditconfig Command

The `auditconfig` command provides a command line interface to get and set audit configuration information and audit policy. It can be used in the `audit_startup(1M)` script to set audit policies when the audit daemon is started. See the `auditconfig(1M)` man page and “Dynamic Procedures” on page 70, for examples of the use of the `auditconfig` command.

<code>-chkconf</code>	Check the configuration of kernel audit event to class mappings and report any inconsistencies.						
<code>-conf</code>	Reconfigure kernel event to class mappings at runtime to match the current mappings in the <code>audit_event</code> file.						
<code>-getcond</code>	Get the workstation's auditing condition. The possible responses are.  <table><tr><td><code>auditing</code></td><td>Auditing is enabled and turned on.</td></tr><tr><td><code>no auditing</code></td><td>Auditing is enabled but turned off.</td></tr><tr><td><code>disabled</code></td><td>The audit module is not enabled.</td></tr></table>	<code>auditing</code>	Auditing is enabled and turned on.	<code>no auditing</code>	Auditing is enabled but turned off.	<code>disabled</code>	The audit module is not enabled.
<code>auditing</code>	Auditing is enabled and turned on.						
<code>no auditing</code>	Auditing is enabled but turned off.						
<code>disabled</code>	The audit module is not enabled.						
<code>-setcond <i>condition</i></code>	Set the workstation's auditing condition: <code>auditing</code> or <code>noaudit</code> . To disable auditing, modify the <code>audit</code> script and the <code>system(4)</code> file and reboot. See “To Disable Auditing” on page 48 for the procedure.						
<code>-getclass <i>event_number</i></code>	Get the preselection classes to which the specified event is mapped.						
<code>-setclass <i>event_number</i></code>	Set the preselection classes to which the specified event is mapped.						
<code>-lsevent</code>	Display the currently configured (runtime) kernel and user audit event information.						
<code>-getpinfo <i>pid</i></code>	Get the audit ID, preselection mask, terminal ID, and audit session ID of the specified process.						

<code>-setkmask +/audit_flags</code>	Set the kernel preselection mask for non-attribute events to the specified audit flags.
<code>-setkmaskac</code>	Set the kernel preselection mask for non-attribute events to the classes specified in the <i>naflags</i> field of the <code>audit_control</code> file.
<code>-setpmask pid flags</code>	Set the preselection mask of the specified process.
<code>-setsmask asid flags</code>	Set the preselection mask of all processes with the specified audit session ID.
<code>-setumask audit_flags</code>	Set the preselection mask of all processes with the specified user audit ID.
<code>-lspolicy</code>	Display the list of audit policies with a short description of each one.
<code>-getpolicy</code>	Get the current audit policy flags.
<code>-setpolicy policy_flag</code>	Set the audit policy flags to the specified policies. See “Setting Audit Policies” on page 35.

---

## Setting Audit Policies

You can use `auditconfig` with the `-setpolicy` option to change the default Trusted Solaris audit policies. Setting audit policies means to add optional audit tokens to the audit record. The `auditconfig` command with the `-lspolicy` argument shows the audit policies that are optional. See “To Determine Current Audit Policy” on page 70 for the audit policies and their short descriptions. The following gives longer descriptions of the less easily understood policy flags.




---

**Caution** - To run auditing in an evaluated configuration, you cannot have the `cnt` policy or the `passwd` policy turned on. They *must* be turned off.

---

<code>ahlt</code>	Halt the machine if an asynchronous audit event occurs which can not be delivered to the audit queue. The default is not to halt the workstation.
-------------------	---

`cnt` Do not suspend auditable actions when the queue is full; just count how many audit records are dropped. The default is suspend.

---

**Note** - To return to the default, remove the `cnt` policy. See “To Set Audit Policy Temporarily” on page 71 for examples of replacing, adding, and removing audit policies.

---

`path` Add secondary `path` tokens to audit record. These secondary paths are typically the path names of dynamically linked shared libraries or command interpreters for shell scripts. By default they are not included.

`seq` Include a sequence number in every audit record. The default is to not include. (The sequence number could be used to analyze a crash dump to find out whether any audit records are lost.)

## Auditing Setup

---

The focus of this chapter is on setting up auditing for a network of Trusted Solaris workstations. It also describes how to set up auditing for a non-networked Trusted Solaris workstation.

- “Planning Auditing at Your Site” on page 37
- “Planning the Rollout” on page 44
- “Rolling Out Auditing at Your Site” on page 45
- “Audit Shutdown and Startup Procedures” on page 47
- “Basic Audit Setup Procedures” on page 51
- “Advanced Audit Setup Procedures” on page 66
- “Dynamic Procedures” on page 70

---

## Planning Auditing at Your Site

When the system administrator and security administrator configure the first workstation for Trusted Solaris, auditing is enabled and a limited number of audit records are collected to a default audit location, *workstation\_name*: `/var/audit`. The security administrator needs to plan what to audit and whether to customize site-specific event-to-class mappings. The system administrator plans disk space (local and remote) for the audit files, an audit administration server, and the order of installation.

Planning auditing for a non-networked workstation is a bit simpler. For a single workstation, customizing event-to-class mappings may not be worth the time. Your most important task is to ensure that auditing does not slow down your work. Planning the size and locations of auditing partitions can prevent work slowdown,

and a regular maintenance schedule can automatically back up and free up the audit partition for more audit records.

## Planning What to Audit

Trusted Solaris collects user actions and non-attributable (in the class `na`, `non-attribute`) events into audit classes. It is these audit classes, each of which holds a number of events, that are audited for success, for failure, or for both.

Before configuring auditing, understand the audit flags and the types of events they flag. Develop a philosophy of auditing for your organization that is based on the amount of security your site requires, and the types of users you administer.

Unless the process audit preselection mask is modified dynamically, the audit characteristics in place when a user logs in are inherited by all processes during the login session, and, unless the databases are modified, the process preselection mask applies in all subsequent login sessions.

See Appendix A for a list of provided audit classes. The list shows per audit class, each audit event's corresponding system call or user command, and points you to its audit record format.

The security administrator plans what to audit based on the site security policy. You can configure a system-wide setup and user exceptions/additions.

### 1. Decide if non-attributable events should be audited.

The audit flag `na` represents the non-attributable class of events. For example, accessing the PROM, booting, and remote mounting are non-attributable events. See "Events in Audit Class `na` " on page 127 for a list of the events in the default `non-attribute` class.

When you audit a class, you audit all events in that class. If you want to customize the non-attributable class, see "Planning a Site-Specific Event-to-Class Mapping" on page 40.

To audit non-attributable events, you will enter the `na` flag on the `naflags:` line of the `audit_control` file.

### 2. Decide whether to audit them for success, for failure, or for both.

To audit non-attributable events for success, the `naflags:` line of the `audit_control` file would look like:

```
naflags:+na
```

To audit non-attributable events for failure:

naflags:-na

To audit non-attributable events for both:

naflags:na

### 3. Decide if *all* events will be audited.

---

**Note** - The class `all` includes all auditable events in the Trusted Solaris software system. While unusual circumstances may dictate use of this class, typically you would avoid auditing all events.

---

### 4. If you are not going to audit all events, repeat Steps Step 1 on page 38 and Step 2 on page 38 for the other audit classes for the class `na`.

The decisions you have made in Steps Step 3 on page 39 and Step 4 on page 39 you will enter in the `audit_control` file when establishing auditing on the first workstation.

To audit these events, you will enter their flag on the `flags:` line of the `audit_control` file, just as you entered the `na` flag in the `naflags:` line.

### 5. Determine if there are particular users or roles that should be audited slightly differently than the system-wide setup.

You will enter user exceptions to the system setup in the `audit_user` file.

### 6. Be consistent.

All workstations in a Trusted Solaris network should have identical `naflags:` entries in their `audit_control` files.

All workstations in a Trusted Solaris network should have identical `flags:` entries in their `audit_control` files.

All workstations in a Trusted Solaris network should have identical `audit_user` files.

## Considerations When Planning What to Audit

What is audited at your site is based on your site policy and the costs of auditing (time, efficiency, disk space). For a discussion of the costs, see “Controlling Audit Costs ” on page 32. The following are factors to consider when using auditing as it is implemented in the Trusted Solaris environment.

- Every audit record stands alone, so records can quickly fill up disk space.

Therefore, you might want to start with a small amount of auditing and see how the audit partitions fill. You can then make more educated estimates of disk requirements and an audit archiving schedule. You can refine audit classes as you get an estimate of the size of the audit trail.

- The number of events in an audit class does not necessarily correlate to how many records are generated.

For example, the `file read` class contains about the same number of events as the `login` or `logout` class. Enabling the `file read` class for success is likely to generate many more records than enabling the `login` or `logout` class for success.

- Auditing for failure locates abnormal events; auditing for success monitors system use.

If site policy requires monitoring of system use, you will want to set aside more space for the audit trail than if you are auditing for abnormal events.

- Auditing for failure may generate many fewer records than auditing for success.

For example, auditing for failure of `file read` events in a Trusted Solaris system of sophisticated users can generate many fewer records than turning on the `file read` class for success.

- Configuring the audit classes differently, or setting up new audit classes for audit events can more efficiently satisfy your site requirements. By excluding audit events that site policy does not require to be audited, the audit trail is smaller.

For example, you may want to create a class `de` for devices. When configuring devices, audit the class for success to generate a record of what devices have been set up and tested. When all devices have been configured, you may want to audit the class for failure.

- Configuring some classes to be audited intermittently may satisfy your site requirements.

For example, you may want to audit the audit class you created, `de`, intermittently. A cron job, or the command `auditconfig(1M)`, enable you to turn auditing on and off for particular classes and set other audit flags dynamically.

## Planning a Site-Specific Event-to-Class Mapping

*Optional:* Skip this section if you are using the default event-to-class mappings provided by Trusted Solaris. Do not skip this section if you have decided to rearrange what events are assigned to what classes, or to create new classes or new events.

Trusted Solaris allows 32 audit classes, including the class `all`. Your site may add classes until the total number is 32.

The security administrator plans site-specific mappings. To plan site-specific mappings:



1. **Decide what classes are needed.**
2. **Decide what events belong in what classes.**
  - a. **Decide what events should be copied to another class or classes.**

An audit event can belong to more than one class. For example, the audit event `AUE_RENAME` belongs to the classes `file create` and `file delete` in the default event-to-class mapping.
  - b. **Decide what events should be moved to another class or classes.**
  - c. **Decide what events should be added to a class or classes.**
3. **For each class, decide whether to audit it for success, for failure, or for both.**

When new software programs include audit events not provided by Trusted Solaris 2.5.1 software, add the events to existing classes or create a new classes for the new events.

## Considerations When Changing Event-to-Class Mappings

The following are factors to consider when changing the contents of default audit classes and creating new ones in the Trusted Solaris environment.

- This document, *Trusted Solaris Audit Administration*, reports the default auditing configuration.

Document your site's modifications to the auditing defaults, and make the document available to the administrators handling audit administration.
- If you are networked, you must change the auditing configuration files on all the workstations when you change the files on one workstation.

A network of Trusted Solaris workstations behaves like one workstation. When auditing is enabled, it is enabled on every workstation, and every workstation is audited for the same classes, has the same defaults, has the same user exceptions, and has the same event-to-class mappings as every other Trusted Solaris workstation in the network.

## Planning Space for Audit Records

Storing audit records on a non-networked workstation involves setting up at least two local partitions dedicated to audit records, one primary and one backup, and planning a maintenance schedule.

Storing audit records for a network of workstations involves setting up a local (backup) partition dedicated to audit records, plus a network of audit servers with

partitions for remote (primary) audit storage, and plus an audit administration server from which the entire *audit trail* can be monitored. The audit trail is every audit file (audit files hold audit records generated on a workstation) created by every workstation on the network.

## Planning Space on a Non-Networked Workstation

On a non-networked workstation, plan the size of a disk partition to hold audit records. For efficiency, it is best to place the audit records on a separate disk. For safety, you may want to create two audit partitions on that disk, one as the primary storage area and the other as a backup if the first partition gets full. Set filesystem security attributes to set on the audit directory to prevent snooping on the audit trail.

### 1. Estimate the volume of auditing between audit record backups.

Balance your security needs against the availability of disk space for audit trail storage.

A rule of thumb is to assign 200 MB of space per workstation. However, the disk space requirements for the workstation are based on how much auditing you perform and may be far greater than this figure.

“Controlling Audit Costs ” on page 32 and “Auditing Efficiently” on page 33 provide guidance on how to reduce storage requirements.

### 2. Decide at what point the audit file system sends a warning that it is filling up.

You will specify what is called the *minfree limit* for audit partitions in the `audit_control` file. This is the percentage of disk space remaining when the audit administrator is sent an email message (by the `audit_warn` alias) that the disk is getting full. The default is to send the warning when there is 20% disk space remaining. This percentage is tunable.

## Planning Space on a Network of Workstations

A networked system should include audit servers to store audit files for users’ workstations, an audit administration server for central audit analysis and backup, and a local audit partition on every workstation. You may want to set filesystem security attributes on the directories and mount points to prevent snooping on the audit trail. Create a worksheet to record your auditing plan, or use another mechanism that helps you track the auditing network that you set up.

### 1. Determine how much auditing your site needs to do.

Balance your site’s security needs against the availability of disk space for audit trail storage.

A rule of thumb is to assign 200 MB of space for each workstation that will be on the distributed system, but remember that the disk space requirements at your site is based on how much auditing you perform and may be far greater than this figure per workstation. If you are able to dedicate a local and a remote disk for

auditing, one way to set up audit partitions is to divide each disk into two partitions.

“Controlling Audit Costs ” on page 32 and “Auditing Efficiently” on page 33 provide guidance on how to reduce storage requirements while still maintaining site security.

**2. Decide at what point each audit file system for the workstation sends a warning that it is filling up.**

You will specify what is called the *minfree limit* for audit partitions in the `audit_control` file. This is the percentage of disk space remaining when the audit administrator is sent an email message (by the `audit_warn` alias) that the disk is getting full. The default is to send the warning when there is 20% disk space remaining. This percentage is tunable.

**3. Determine which workstations will be audit servers.**

The system administrator and you will install these workstations before installing the audit client workstations.

**4. Plan a local audit partition for each workstation.**

The local partition provides a backup in cases where the audit server's partitions are full or when the network is unreachable.

**5. Determine which clients will use which audit file systems on which audit server.**

Lay out the auditing network. The following figure shows an audit server, `egret`, with file systems `/etc/security/audit/egret[.n]/files` available to store remote hosts' audit records.

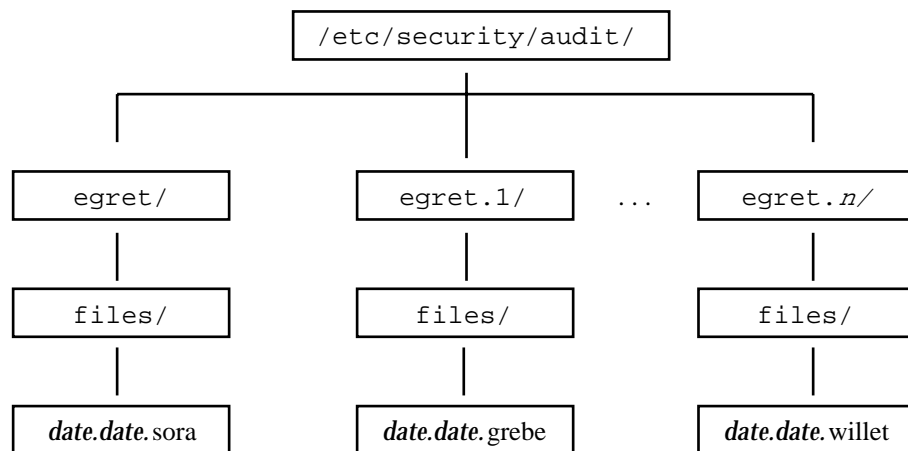


Figure 2-1 Audit Server `egret`'s Audit File Systems

## 6. Follow the naming conventions for audit file systems.

As illustrated in the figure, the convention for naming the audit file systems on a workstation is:

```
/etc/security/audit/workstationname/files  
/etc/security/audit/workstationname.1/files  
/etc/security/audit/workstationname.2/files  
/etc/security/audit/workstationname.3/files ...
```

For an explanation of the naming scheme, see “Audit Storage” on page 23.

## Planning the Rollout

Rolling out the auditing plan to the workstations is a job coordinated by the system administrator, who sets up the disks and the network of audit storage, and the security administrator, who decides what is to be audited and enters the information in the audit configuration files. Together, you want to set up an audited network of workstations where:

- From one workstation, the audit analyst is able to read every audit file on every workstation in the network, and the system operator is able to back up every audit file on every workstation on the network.

*How:* Create an administration server, and mount all audit directories on the server.

- The audit trail is not available for snooping.

*How:* Protect audit directories with appropriate discretionary access controls and mandatory access controls. You may want to audit directory access.

- Each workstation in a Trusted Solaris system is writing records to the audit trail from the first time it is in multiuser mode, and thereafter.

*How:* Create audit servers before you create user workstations. On all workstations, create a dedicated audit partition during installation.

- Every workstation is audited identically.

*How:* Create a central location for all audit configuration files: `audit_event`, `audit_class`, `audit_control`, `audit_startup`, `audit_user`, and `audit_warn`. The examples use the directory `/export/home/tmp` on the NIS+ master. Copy these files to a tape or diskette that is copied to every workstation.

- When an end user's workstation is configured, it is able to immediately send its audit records to an audit server.

*How:* Create the audit servers and configure them for receiving audit records before the end user workstations are set up. Create a procedure to copy the system-wide audit configuration files to each workstation and to modify the `audit_control` file for the audit storage locations for that workstation.

- End user's workstations are not slowed down by writing audit records.

*How:* Regular archiving of the audit trail frees up audit server disk space. Placing the local audit storage on a separate or little-used disk will enable the end user to work quickly when audit records are stored locally.

---

## Rolling Out Auditing at Your Site

To roll out auditing, the system administrator sets up the audit administration server, the audit file servers, the local audit partitions, and what usernames are warned of audit trouble. The security administrator edits the `audit_control(4)` file on the NIS+ root master, and edits other audit configuration files before copying them to a central directory for distribution by tape or floppy. The audit configuration files are copied from the tape to each workstation as it is configured by the install team. The security administrator edits the `dir:` lines in the `audit_control` file on each workstation before the system is rebooted.

---

**Note** - Administrators should understand that Trusted Solaris only records the security-relevant events that it is configured to record (that is, by preselection). Therefore any subsequent audit can only consider the events recorded. If auditing is not configured to record the security-relevant events for the particular system environment in which it operates, it will not be possible to audit. This may mean that attempts to breach the security of the system go undetected, or that the administrator is unable to detect the user responsible for an attempted breach of security. Administrators should regularly analyze audit trails to check for breaches of security.

---

## System Administrator's Audit Setup Tasks

TABLE 2-1 Basic Auditing Setup by the System Administrator

Task	For the procedure, see...
Create audit partitions	"To Create Dedicated Audit Partitions" on page 51
Create audit administration server	<i>Trusted Solaris Installation and Configuration</i> or <i>Trusted Solaris Administrator's Procedures</i>

**TABLE 2-1** Basic Auditing Setup by the System Administrator *(continued)*

<b>Task</b>	<b>For the procedure, see...</b>
Install audit file servers	Plan to install them before audit clients
Create files directory	"To Create an Audit Directory" on page 56
Export audit partitions (networks only)	"To Share an Audit File System" on page 57
Edit Aliases database	"To Warn of Audit Trouble" on page 62
Mount audit partitions (networks only)	"To Mount an Audit File System" on page 57

## Security Administrator's Audit Setup Tasks - Basic

**TABLE 2-2** Basic Auditing Setup by the Security Administrator

<b>Task</b>	<b>For the procedure, see...</b>
On first workstation	
Edit audit_control file	"To Set Audit Flags" on page 60
	"To Reserve Free Space on an Audit File System" on page 58
	"To Specify the Audit File Storage Locations" on page 59
Set Solaris security attributes	"To Protect an Audit File System" on page 54
Edit audit_user file	"To Set User Exceptions to the Audit Flags" on page 61

**TABLE 2-2** Basic Auditing Setup by the Security Administrator *(continued)*

Task	For the procedure, see...
Edit audit_startup file	"To Set Audit Policy Permanently" on page 62
Copy for distribution (networks only)	"To Distribute Audit Configuration Files to a Network of Workstations" on page 63
Set security attributes	"To Protect an Audit File System" on page 54

## Security Administrator's Audit Setup Tasks - Advanced

**TABLE 2-3** Advanced Auditing Setup by the Security Administrator

Task	For the procedure, see...
On first workstation	
	"To Add Audit Events" on page 67
Edit audit_event file	"To Change Event-Class Mappings" on page 68
Edit audit_class file	"To Add Audit Classes" on page 66
Copy for distribution (networks only)	"To Distribute Audit Configuration Files to a Network of Workstations" on page 63

## Audit Shutdown and Startup Procedures

The following procedures describe how to enable and disable auditing for one or more workstations. The commands should be run only on a diskfull workstation, and never on a diskless client.

Auditing tasks require commands and actions that are limited to particular roles and particular labels. Read each task for the administrative role that can perform it, and the label required. See “To Execute Commands that Require Privilege” on page 52 for how to assume a role and open a privileged shell.

## ▼ To Disable Auditing

1. As role **secadmin**, at label **admin\_low**, open the script `/etc/init.d/audit` using the Admin Editor.

---

**Note** - This should be done *only if* auditing is not a site security requirement, or in cases of audit file overflow. The security administrator is responsible.

---

2. Comment out the start script:

```
...
# Start the audit daemon
# if [ -f /etc/security/audit_startup ] ; then
#   echo "starting audit daemon"
#   /etc/security/audit_startup
#   /usr/sbin/auditd &
# fi
...
```

3. Write and quit the file.
4. Open the script `/etc/init.d/drvconfig` using the Admin Editor.
5. Add the following lines to the end of the file:

```
# Disable auditing

#

/usr/bin/adb -wk /dev/ksyms /dev/mem > /dev/null <<end
audit_active/W 0
end
```



6. Prevent spurious messages about the audit daemon at shutdown by commenting out the stop script in `/etc/init.d/audit`:

```
...
# Stop the audit daemon

#   if [ -f /etc/security/audit_startup ] ; then
#       /usr/sbin/audit -t
#   fi
```

7. Write and quit the file.
8. For the changes to take effect, reboot.

---

**Note** - A user or role requires authorization to reboot the workstation.

---

- a. Choose Shut Down from the TP (Trusted Path) menu.
- b. Confirm the shutdown.
- c. Enter `boot` at the `ok` prompt or `b` at the `>` prompt:

```
Type help for more information
<#2> ok boot
Type b (boot), c (continue), or n (new command mode)
> b
```

## ▼ To Enable Auditing

By default, auditing is enabled. If you have disabled auditing, enable it by reversing the above procedure.

1. As role `secadmin`, at label `admin_low`, open the script `/etc/init.d/audit` using the Admin Editor.
2. Remove the comments from the audit start script:

```
...
# Start the audit daemon
if [ -f /etc/security/audit_startup ] ; then
    echo "starting audit daemon"
    /etc/security/audit_startup
    /usr/sbin/auditd &
fi
...
```

**3. Write and quit the file.**

**4. Enable the audit daemon to exit gracefully at shutdown by removing the comments in the stop script in /etc/init.d/audit:**

```
...

# Stop the audit daemon
if [ -f /etc/security/audit_startup ] ; then
    /usr/sbin/audit -t
fi
```

**5. Write and quit the file.**

**6. Open the script /etc/init.d/drvconfig using the Admin Editor.**

**7. Comment out the Disable auditing lines:**

```
# Disable auditing

#
# /usr/bin/adb -wk /dev/ksyms /dev/mem > /dev/null <<end
# audit_active/W 0
# end
```

**8. Write and quit the file.**

**9. For the changes to take effect, reboot using the Shut Down menu item from the TP (Trusted Path) menu.**

---

# Basic Audit Setup Procedures

The following procedures describe how to set up auditing for one or more workstations.

## ▼ To Create Dedicated Audit Partitions

- ◆ During installation, the install team creates dedicated audit partition(s) when formatting the disks.

Use the naming convention `/etc/security/audit/workstation_name(.n)`

A diskfull workstation should have at least one local audit directory, which it can use as a directory of last resort, if unable to communicate with the audit server.

See “Audit Storage” on page 23 for an explanation of the naming convention.

On an audit file server, most partitions hold audit files, as is shown in the following example of the egret audit file server:

Disk	Slice	Mount point	Size
c0t2d0	s0	/etc/security/audit/egret	1.0 GB
	s1	/etc/security/audit/egret.1	.98 GB
	s2	entire disk	1.98 GB
c0t2d1	s0	/etc/security/audit/egret.2	502 MB
	s1	/etc/security/audit/egret.3	500 MB
	s2	entire disk	1002 MB

---

**Note** - Another disk holds egret’s / (root) and /swap partitions.

---

On a diskfull workstation, including the audit administration server, at least one partition should be dedicated to local audit files, as is shown in the following example of the workstation willet:

Disk	Slice	Mount point	Size (MB)
c0t3d0	s0	/	70
	s1	swap	180
	s2	entire disk	1002
	s3	/usr	350
	s4	/etc/security/audit/willet	202
	s7	/export/home	200

## Hints

A rule of thumb is to assign 200 MB of space for each workstation. However, the disk space requirements at your site will be based on how much auditing you perform and may be far greater than this figure.

Fewer and large partitions are more efficient than more and smaller ones.

---

**Note** - To add a disk to hold audit partitions after installing the workstation, see the *Solaris 7 System Administration Guide, Volume II*. To protect the disks with Trusted Solaris security attributes, see *Trusted Solaris Administrator's Procedures*.

---

## ▼ To Execute Commands that Require Privilege

Most commands for setting up auditing require the use of a profile shell, where commands can run with privilege. Auditing also requires the use of actions in the System\_Admin and Solstice\_Apps folders of Application Manager.

## Log In and Assume an Administrative Role

### 1. Log in to the workstation as yourself.

#### a. Enter your user name and press the Return key.

If the workstation is protected against anyone logging in, the Enable Logins dialog is displayed.

#### b. If you are allowed to enable logins, click the Yes button after Login:.

If you are not allowed to enable logins, ask the administrator to enable logins.

#### c. Enter your password.

#### d. Click OK to dismiss the dialog.

You are presented with the message of the day and a label builder screen. In a single-label system, the screen describes your session label. In a multilabel system, it presents you with a label builder to choose your session clearance.

- e. **Accept the default unless you have a reason not to.**

Press the Return key or click the OK button and be logged in.

2. **Assume an administrative role that you have been assigned.**
  - a. **Click the right mouse button in the middle of the Front Panel.**
  - b. **Choose Assume *administrative* Role from the menu.**
  - c. **At the password prompt, enter the password for that role.**

## Open a Profile Shell

1. **As an administrative role, in a workspace at the label required by the task, launch a terminal by right-clicking the background and selecting Tools > Terminal from the menu.**
2. **Enter the `clist` command.**

```
# clist
```

If a list of commands prints, you are in a profile shell.

## ▼ To Remove Free Space (Optional)

1. **As role `admin`, at label `admin_low`, unmount the audit partitions from the workstation by running the `umount(1M)` command.**

For example, on the audit file server `egret`:

```
egret$ umount /etc/security/audit/egret
egret$ umount /etc/security/audit/egret.1
egret$ umount /etc/security/audit/egret.2
egret$ umount /etc/security/audit/egret.3
```

- 2. Reduce reserved filesystem space on each partition to 0% with the command `tunefs -m 0`.**

The security administrator sets the reserved filesystem space (called the minfree limit) in the `audit_control(4)` file.

For example, on the audit file server egret:

```
egret$ tunefs -m 0 /etc/security/audit/egret
egret$ tunefs -m 0 /etc/security/audit/egret.1
egret$ tunefs -m 0 /etc/security/audit/egret.2
egret$ tunefs -m 0 /etc/security/audit/egret.3
```

Similarly, on the workstation willet:

```
willet$ umount /etc/security/audit/willet
willet$ tunefs -m 0 /etc/security/audit/willet
```

See the `tunefs(1M)` man page for more information on the advantages and disadvantages of tuning a file system.

## ▼ To Protect an Audit File System

- 1. As role `secadmin`, at label `admin_low`, set the appropriate file permissions on every audit file system while the file system is unmounted.**

For example, on the audit file server egret:

```
egret$ chmod -R 750 /etc/security/audit/egret
egret$ chmod -R 750 /etc/security/audit/egret.1
egret$ chmod -R 750 /etc/security/audit/egret.2
egret$ chmod -R 750 /etc/security/audit/egret.3
```

On the workstation willet:

```
willet$ chmod -R 750 /etc/security/audit/willet
```

2. As role secadmin, at label `admin_high`, set any Trusted Solaris security attribute defaults required by your site security policy on every audit file system while the file system is unmounted.

To run the command at the label `admin_high`, you must create an `admin_high` workspace. Follow the procedure in “To Create an Admin\_High Workspace” on page 71.

For example, the following command on the audit file server `egret` should be repeated for all of its audit partitions:

```
egret$ setfsattr -l ``admin_high;admin_high`` -s ``[admin_high]`` \
/etc/security/audit/egret
```

On the workstation willet:

```
willet$ setfsattr -l ``admin_high;admin_high`` -s ``[admin_high]`` \
/etc/security/audit/willet
```

The `-l` option ensures that all files created in the file system are labeled `admin_high`, and the `-s` option sets the partition's default sensitivity label for the audit files. See the `setfsattr(1M)` man page for more information.

---

**Note** - The local audit file systems must already be in the host's `/etc/vfstab` file.

---

## ▼ To Create an Audit Directory

### 1. As **admin**, at label `admin_high`, remount the local audit file systems.

Follow the procedure in “To Create an Admin\_High Workspace” on page 71 to get an `admin_high` process.

For example, on the audit file server `egret`:

```
egret$ mount /etc/security/audit/egret
egret$ mount /etc/security/audit/egret.1
egret$ mount /etc/security/audit/egret.2
egret$ mount /etc/security/audit/egret.3
```

Similarly, on the workstation `willet`:

```
willet$ mount /etc/security/audit/willet
```

### 2. Create a directory named `files` at the top of each mounted audit partition.

For example, on the audit file server `egret`:

```
egret$ mkdir /etc/security/audit/egret/files
egret$ mkdir /etc/security/audit/egret.1/files
egret$ mkdir /etc/security/audit/egret.2/files
egret$ mkdir /etc/security/audit/egret.3/files
```

On the workstation `willet`:

```
willet$ mkdir /etc/security/audit/willet/files
```



## ▼ To Share an Audit File System

1. As role **admin**, at label **admin\_low**, enter every local audit file system in the local host's **dfstab(4)** file.
  - a. Click the **Application Manager**, double-click the **System\_Admin** folder, and double-click the **Share Filesystems** action.

- b. Enter and protect each local audit directory in the **dfstab** file.

For example, the audit file server **egret** has the following entries:

```
share -F nfs -o ro -d "local audit files" /etc/security/audit/egret
share -F nfs -o rw=willet:audubon -d "audit willet" /etc/security/audit/egret.1
share -F nfs -o rw=grebe:audubon -d "audit grebe" /etc/security/audit/egret.2
share -F nfs -o rw=sora:audubon -d "audit sora" /etc/security/audit/egret.3
```

The workstation **willet** has the following entry:

```
share -F nfs -o ro -d "local audit files" /etc/security/audit/willet
```

2. Reboot the workstation by choosing **Shut Down** from the **TP** menu.

## ▼ To Mount an Audit File System

1. As role **admin** at label **admin\_low**, on **audubon**, the audit administration server, create a mount point for every audit directory in the **Trusted Solaris** network.

For example, on the audit administration server **audubon**:

```
audubon$ mkdir /etc/security/audit/willet
audubon$ mkdir /etc/security/audit/egret
audubon$ mkdir /etc/security/audit/egret.1
...
```

2. As role **admin**, at label **admin\_low**, enter every audit partition on the network in the audit administration server's **vfstab(4)** file.

Mount audit directories with the read-write (**rw**) option. Mount remote partitions using the **soft** option.

- a. Click the Application Manager, double-click the System\_Admin folder, and double-click the Set Mount Points action.
- b. Enter the mount points in the `vfstab(4)` file.

The following shows part of the `vfstab` file on audubon:

```
egret:/etc/security/audit/egret - /etc/security/audit/egret nfs - yes bg,soft,nopriv
egret:/etc/security/audit/egret.1 - /etc/security/audit/egret.1 nfs - yes bg,soft,nopriv
egret:/etc/security/audit/egret.2 - /etc/security/audit/egret.2 nfs - yes bg,soft,nopriv
egret:/etc/security/audit/egret.3 - /etc/security/audit/egret.3 nfs - yes bg,soft,nopriv
willet:/etc/security/audit/willet - /etc/security/audit/willet nfs - yes bg,soft,nopriv
...
```

3. On each workstation, create the mount points for the remote audit file servers' partitions that are used by the workstation, and enter them in the `vfstab(4)` file. Do this as role `admin`, at label `admin_low`.

For example, to create the mount points on the workstation `willet`:

```
willet$ mkdir /etc/security/audit/egret
willet$ mkdir /etc/security/audit/audubon.2
```

- a. Click the Application Manager, double-click the System\_Admin folder, and double-click the Set Mount Points action.
- b. Enter the mount points in the `vfstab(4)` file.

The following shows part of the `vfstab` file on `willet`:

```
egret:/etc/security/audit/egret - /etc/security/audit/egret nfs - yes bg,soft,nopriv
audubon:/etc/security/audit/audubon.2 - /etc/security/audit/audubon.2 nfs - yes nopriv
```

## ▼ To Reserve Free Space on an Audit File System

1. As role `secadmin`, at label `admin_low`, enter reserve free space in the `audit_control(4)` file.

- a. Open the `System_Admin` folder from the Application Manager.
  - b. Double-click the Audit Control action.
2. Enter a value between 10 and 20 on the `minfree:` line.

```
dir:/var/audit
flags:
minfree:20
naflags:
```

3. Write the file and quit the editor.

## ▼ To Specify the Audit File Storage Locations

1. As role `secadmin`, at label `admin_low`, enter audit storage locations in the `audit_control` file.
  - a. Open the `System_Admin` folder from the Application Manager.
  - b. Double-click the Audit Control action.
2. On the first workstation installed, enter its local audit file system as the value of the `dir:` line.

The following shows the `audit_control` file for `grebe`, the NIS+ root master.

```
dir:/etc/security/audit/grebe/files
flags:
minfree:20
naflags:
```

3. When the audit file servers have been installed and configured, add their (mounted) filesystem names plus their top-level directory, `files` to the `dir:` entry.

The mounted file systems are listed before the workstation's local file system, as in:

```
dir:/etc/security/audit/egret/files
dir:/etc/security/audit/egret.1/files
dir:/etc/security/audit/grebe/files
flags:
minfree:20
naflags:
```

**4. Write the file and exit the editor.**

- 5. As role secadmin in an admin\_high profile shell, execute the `audit -s` command to have the audit daemon re-read the `audit_control` file and write audit records to the designated directory.:**

```
$ audit -s
```

By default, the audit records have been stored in `/var/audit`. The audit records will now be stored in the first directory in the `audit_control` file.

## ▼ To Set Audit Flags

- 1. As role secadmin, at label admin\_low, enter system-wide audit flags in the `audit_control(4)` file.**
  - a. Open the System\_Admin folder from the Application Manager.**
  - b. Double-click the Audit Control action.**
- 2. Enter the `na` class in the `naflags:` line if your site is auditing non-attributable events.**

```
dir:/etc/security/audit/egret/files
dir:/etc/security/audit/egret.1/files
dir:/etc/security/audit/grebe/files
flags:
minfree:20
naflags:na
```

- 3. Enter other classes in the `flags:` line if your workstation is auditing user-level events.**

```
dir:/etc/security/audit/egret/files
dir:/etc/security/audit/egret.1/files
dir:/etc/security/audit/grebe/files
flags:lo,ad,-all,^-fc
minfree:20
naflags:na
```

See “Sample audit\_control File” on page 25 for an explanation of the syntax of the audit flags’ fields.

**4. Write the file and exit the editor.**

---

**Note** - On a distributed system, the audit flags in the `audit_control` file must be identical on every workstation on the network. See “To Distribute Audit Configuration Files to a Network of Workstations” on page 63 for a process to distribute master copies of files to all workstations on the network.

---

## ▼ To Set User Exceptions to the Audit Flags

As role `secadmin`, at label `admin_low`, enter user exceptions to system-wide audit flags in the `audit_user(4)` file.

**1. Open the System\_Admin folder from the Application Manager.**

**2. Double-click the Audit Users action.**

**3. Enter the user exceptions, write the file, and exit the editor.**

For example, the following entry audits the role `root` for logins and logouts, and never audits the `fc` class, even if it is being audited for the workstation. The `jane` entry audits her for all flags specified in the `audit_control` file except for successful `file_read` events. Null events, `no`, are never audited.

```
# User Level Audit User File
#
# File Format
#
#      username:always:never
#
root:lo:no,fc
jane:all,^+fr:no
```

## ▼ To Warn of Audit Trouble

1. As role **admin**, at label `admin_low`, enter mail alias to warn of audit trouble in the Aliases database.
  - a. Open the `Solstice_Apps` folder from the Application Manager.
  - b. Double-click the Database Manager.
  - c. Choose `NIS+`, and Aliases, and press Return.
2. Add an alias called `audit_warn(1M)` for notifying its members of audit trouble.

For example, this `audit_warn` alias emails the security administrator and the system administrator when the auditing subsystem needs attention.

Alias: `audit_warn`  
Expansion: `secadmin@grebe,admin@grebe`

## ▼ To Set Audit Policy Permanently

1. As role **secadmin**, at label `admin_low`, enter permanent audit policy in the `audit_startup(1M)` file.
  - a. Open the `System_Admin` folder from the Application Manager.
  - b. Double-click the Audit Startup action.
2. Create a script that calls the `auditconfig(1M)` command with policy options.

The sample `audit_startup(1M)` script below adds ACLs to audit records, halts the workstation when its audit file systems are full, and at startup, prints the current audit policy to standard i/o.

```
#!/bin/sh
auditconfig -setpolicy +slabel,+acl
auditconfig -setpolicy +ahlt
auditconfig -getpolicy
```

### 3. Write the file and exit the editor

---



**Caution** - To run auditing in an evaluated configuration, the `cnt` policy cannot be turned on; the `ahlt` policy (the default) cannot be turned off.

---

## ▼ To Distribute Audit Configuration Files to a Network of Workstations

1. During installation, as root, at label `admin_low`, create a directory on the first installed workstation to hold copies of the audit configuration files customized for your site.

The directory would include your customized versions of `audit_control`, `audit_user`, `audit_startup`, and `audit_warn`. If you have modified event-to-class mappings, it would include `audit_event` and `audit_class`. It would not include `audit_data`.

For example, on `grebe`, the first workstation in a network:

```
# mkdir /export/home/tmp
```

2. Copy the modified files from the `/etc/security` directory to the `/export/home/tmp` directory.

```
# cp /etc/security/audit_control /export/home/tmp/audit_control
# cp /etc/security/audit_user /export/home/tmp/audit_user
# cp /etc/security/audit_startup /export/home/tmp/audit_startup
# cp /etc/security/audit_event /export/home/tmp/audit_event
```

3. Allocate the tape or diskette device.

Follow the procedure in “To Allocate and Deallocate Devices” on page 65.

4. Run the `tar(1)` command to copy the contents of the `/export/home/tmp` directory to tape or to diskette.

- a. To copy to tape

```
# cd /export/home/tmp
# tar cv audit_control audit_user audit_startup audit_event
```

- b. To copy to diskette

```
# cd /export/home/tmp
# tar cvf /dev/diskette \
audit_control audit_user audit_startup audit_event
```

5. Deallocate the tape or diskette device and follow the instructions.

Follow the procedure in “To Deallocate a Device” on page 66.

6. As root, at label `admin_low`, as each new workstation is configured, copy the files from the tape or diskette to the correct directory on the new workstation.

- a. Prepare the directory for the new files.

```
# cd /etc/security
# mv audit_control audit_control.orig
# mv audit_startup audit_startup.orig
# mv audit_warn audit_user.orig
# mv audit_event audit_event.orig
```

- b. Allocate the appropriate device at the label `admin_low`.

Follow the procedure in “To Allocate and Deallocate Devices” on page 65.

- i. To copy from tape

```
# tar xv audit_control audit_user audit_startup audit_event
```



## ii. To copy from diskette

```
# tar xvf /dev/diskette \  
audit_control audit_user audit_startup audit_event
```

### a. Deallocate the device.

Follow the procedure in “To Deallocate a Device” on page 66.

7. As role secadmin, at label admin\_low, modify the audit\_control file on each new workstation with that workstation’s remote and local audit file systems.

## ▼ To Allocate and Deallocate Devices

The Device Manager allocates and deallocates devices.

1. In the role and in a workspace at the label required, click the left mouse button on the triangle above the Style Manager icon on the Front Panel.

The Trusted Desktop subpanel is displayed.

2. Click the Device Manager icon once.

Device Manager —



3. Double-click the device to be allocated.

mag\_tape\_0 allocates a tape device. floppy\_0 allocates a diskette.

4. Click OK in the label builder that appears.

The file you load will be labeled admin\_low.

5. Follow the directions in the window that is displayed.

## To Deallocate a Device

1. Go to the workspace where the Device Manager was allocated.
2. Double-click the device to deallocate it.  
A window appears listing devices being deallocated.
3. When prompted, remove the tape or diskette from the drive and label it appropriately.
4. Press the Return key to dismiss the window.
5. Click the top left button and select Close to close the Device Allocation Manager window.

---

## Advanced Audit Setup Procedures

The following procedures describe how to modify the default audit classes and audit events, and to set a public object bit on files and folders to reduce unnecessary auditing.

### ▼ To Add Audit Classes

1. As role `secadmin`, at label `admin_low`, add audit classes in the `audit_classes` file.
  - a. Open the `System_Admin` folder from the Application Manager.
  - b. Double-click the Audit Classes action.
2. Add the classes you planned in “Planning a Site-Specific Event-to-Class Mapping” on page 40, write the file, and exit the editor.



---

**Caution** - Do not reassign the hexadecimal numbers already in use.

---

3. As role `secadmin`, at label `admin_low`, open the Audit Events action to add the new class to each event in the new class.

For events in more than one class, use a comma (no space) to delimit the classes.

4. **Write the file and exit the editor.**
5. **Make any changes to `audit_control(4)` and `audit_user(4)` to audit the events in the new classes.**  
See “To Set Audit Flags” on page 60 and “To Set User Exceptions to the Audit Flags” on page 61 for details of the procedures.

---

**Note** - On a distributed system, the `audit_class`, `audit_event`, `audit_startup`, and `audit_user` files must be identical on every workstation on the network. See “To Distribute Audit Configuration Files to a Network of Workstations” on page 63 for a process to distribute master copies of files to all workstations on the network.

---

6. **Reboot, or as `secadmin` in an `admin_low` profile shell, run the `auditconfig(1M)` command with appropriate options.**  
In the following example, the audit session ID is 159, and the new classes are `gr` (for graphic applications) and `db` (for databases applications).

```
$ auditconfig -setsmask 159 gr,db
```

## ▼ To Add Audit Events

1. **As role `secadmin`, at label `admin_low`, add audit events in the `audit_event(4)` file.**
  - a. **Open the `System_Admin` folder from the Application Manager.**
  - b. **Double-click the Audit Events action.**
2. **Add the events you planned in “Planning a Site-Specific Event-to-Class Mapping” on page 40, write the file, and exit the editor.**  
For events in more than one class, use a comma (no space) to delimit the classes.

---

**Note** - Third-party applications can use the event numbers 32768 through 65536 only. See Table 1-1 for more information.

---

3. **Make any changes to `audit_control(4)` and `audit_user(4)` to audit the events in the new classes.**

See “To Set Audit Flags” on page 60 and “To Set User Exceptions to the Audit Flags” on page 61 for details of the procedures.

---

**Note** - On a distributed system, the `audit_class`, `audit_event`, `audit_startup`, and `audit_user` files must be identical on every workstation on the network. See “To Distribute Audit Configuration Files to a Network of Workstations” on page 63 for a process to distribute master copies of files to all workstations on the network.

---

4. **Reboot, or as `secadmin` in an `admin_low` profile shell, run the `auditconfig(1M)` command with appropriate options.**

In the following example, the audit session ID is 159, and the new events are in the classes `gr` (for graphic applications) and `db` (for databases applications).

```
$ auditconfig -setsmask 159 gr,db
```

## ▼ To Change Event-Class Mappings

1. **Change event-class mappings in the `audit_control(4)` file.**
  - a. **As role `secadmin`, at label `admin_low`, open the `System_Admin` folder from the Application Manager.**
  - b. **Double-click the Audit Events action.**
2. **Edit the file to change the class mapping for each event to be changed, write the file, and exit the editor.**

If you are changing events above number 2048, this is all you need to do.

---

**Note** - On a distributed system, the `audit_class`, `audit_event`, `audit_startup`, and `audit_user` files must be identical on every workstation on the network. See “To Distribute Audit Configuration Files to a Network of Workstations” on page 63 for a process to distribute master copies of files to all workstations on the network.

---

3. **If you modify a kernel event mapping (numbers 1 to 2047):**

- a. Reboot the system, or
- b. As role `secadmin`, at label `admin_low`, change the runtime event-to-class mappings:

```
$ auditconfig -conf
```

## ▼ To Set Public Object Bit on Publicly Accessible Files

Setting the public object bit can reduce the size of the audit trail when the audit record includes successful accesses of files or directories. Successful viewing, listing, or listing of a file or directory's attributes will not be written to the audit record when the file's public object bit is set.

- ◆ **As role `secadmin`, at label `admin_low`, set the public object bit on a local directory of publicly accessible files using the `setfattrflag(1)` command with the `-p 1` option.**

The following command sets the public object bit on the `/etc` directory. A search of the `/etc` directory, or a read of files in the `/etc` directory will not result in an audit record.

```
$ setfattrflag -p 1 /etc
$ getfattrflag /etc
Multilevel directory: no
Single level directory: no
Public object: yes
```

- ◆ **As `secadmin`, at label `admin_low`, set the public object bit on a mounted file system of publicly accessible files using the `attr_flag` security attribute.**

For example, the following entry in the `vfstab_adjunct(4)` file for the mounted file system `/spublic` sets the public object flag for all files in the file system.

```
#      Modified template.
#
/spublic; \
acc_acl=; \
mode=; \
attr_flag=public; \
gid=; \
uid=; \
slabel=; \
forced=; \
allowed=; \
low_range=; \
hi_range=; \
mld_prefix=mldroot;
#
```

See the man page for `mount(1M)` and "Managing Files and File Systems" in the *Trusted Solaris Administrator's Procedures* for more details.

---

## Dynamic Procedures

Dynamic controls apply to one workstation at a time, since the audit command only applies to the current workstation where you are logged in. Use dynamic controls to test auditing on a workstation (estimate volume of records, for example), or to add an auditing flag without having to reboot the workstation. However, if you make dynamic changes on one workstation for other than testing purposes, you should make the changes on all workstations.

---

**Note** - The following procedures work only when auditing is enabled.

---

### ▼ To Determine Current Audit Policy

The `auditconfig(1M)` command enables an appropriately configured role to determine audit policy and to see what policies can be set. If your role is not configured to determine the policy, or if auditing is turned off, the command `auditconfig -getpolicy` returns an error. The following example was run by the role `secadmin`, at label `admin_low`:

```

$ auditconfig -getpolicy
audit policies = none
$ auditconfig -lspolicy
policy string    description:
  arge    include exec environment args in audit recs
  argv    include exec args in audit recs
  cnt     when no more space, drop recs and keep a count
  group   include supplementary groups in audit recs
  seq     include a sequence number in audit recs
  trail   include trailer tokens in audit recs
  path    allow multiple paths per event
  acl     include ACL information in audit recs
  ahl     halt machine if we can't record an async event
  slabel  include sensitivity labels in audit recs
  passwd  include cleartext passwords in audit recs
  windata_down include downgraded information in audit recs
  windata_up include upgraded information in audit recs
  all     all policies
  none    no policies

```

## ▼ To Create an Admin\_High Workspace

To label files `admin_high` or to move them to an `admin_high` directory, to reset the audit daemon, and to make other changes in auditing requires an `admin_high` process. An `admin_high` process starts from an `admin_high` workspace.

1. **Click the right button on the Front Panel and choose Assume secadmin Role from the menu.**  
A secadmin role workspace becomes the current workspace.
2. **In the current workspace, click the right button on the workspace name (secadmin) button and choose Change Workspace SL from the menu.**
3. **In the label builder, click the ADMIN\_HIGH button.**
4. **Click OK at the bottom of the label builder.**

The color of the workspace button turns to black, indicating an `admin_high` workspace. An `admin_high` workspace is available only to an administrative role.

## ▼ To Set Audit Policy Temporarily

The `auditconfig` command enables you to change audit policy, such as whether to include `acl` information in the audit record. Since the policy variable is a dynamic

kernel variable, the policy that you set is in effect until the workstation next boots. See the `auditconfig(1M)` man page for a list of policy parameters.

- ◆ **To set policies in one invocation of the command, or to override all current policies, as role `secadmin` at label `admin_low`, separate the policies with commas (no spaces):**

```
$ auditconfig -setpolicy trail,seq
$ auditconfig -getpolicy
audit policies = trail,seq
$ auditconfig -setpolicy argv,acl
$ auditconfig -getpolicy
audit policies = argv,acl
```

- ◆ **To add policies to the current policies, as role `secadmin` at label `admin_low`, preface each added policy with a plus (+):**

```
$ auditconfig -setpolicy trail,seq
$ auditconfig -getpolicy
audit policies = trail,seq
$ auditconfig -setpolicy +argv
$ auditconfig -setpolicy +acl
$ auditconfig -getpolicy
audit policies = seq,trail,argv,acl
```

- ◆ **To remove policies from the current policies, as role `secadmin` at label `admin_low`, preface each policy to be removed with a minus (-):**

```
$ auditconfig -setpolicy trail,seq
$ auditconfig -getpolicy
audit policies = trail,seq
$ auditconfig -setpolicy -seq
$ auditconfig -getpolicy
```

(continued)



```
audit policies = trail
```

In the examples above, the `trail` and `seq` tokens are added to debug audit trail discrepancies. To set policies permanently, enter the `auditconfig` command in the `audit_startup(1M)` script. See “To Set Audit Policy Permanently” on page 62 for how to edit the script.




---

**Caution** - To run auditing in an evaluated configuration, the `cnt` policy cannot be turned on; the `ahlt` policy (the default) cannot be turned off.

---

## ▼ To Change Audit Flags Dynamically

The `auditconfig(1M)` command enables you to change audit flags dynamically, such as adding extra flags to a user, a session, or a process while the user, session, or process is active. Since the flags are added dynamically, they are in effect until the user logs out, the session ends, or the process ends.

- ◆ **To set a particular user to be additionally audited for successful file reads, as role `secadmin` at label `admin_low`:**

```
$ auditconfig -setumask audit_user_id +fr
```

- ◆ **To set a particular session to be additionally audited for failed file attribute access, as role `secadmin` at label `admin_low`:**

```
$ auditconfig -setmask audit_session_id -fa
```

- ◆ **To set a particular process to be additionally audited for successful and unsuccessful file attribute modifications, as role secadmin at label `admin_low`:**

```
$ ps -ef | grep application-to-be-monitored  
$ auditconfig -setpmask process_id fm
```

## ▼ To Stop the Audit Daemon

Only one audit daemon may run at a time. An attempt to start a second one will result in an error message, and the new one will exit. If there is a problem with the audit daemon, terminate the audit daemon gracefully, then restart it manually.

- ◆ **To stop the audit daemon in event of trouble, as role secadmin, at label `admin_high`:**

```
$ audit -t
```

This is not recommended. Audit records may be lost.

## ▼ To Start the Audit Daemon

The audit daemon starts when the workstation is brought up to multiuser mode, and restarts when the audit daemon is instructed by the `audit -s` command to reread an audit configuration file.

- ◆ **To restart the audit daemon in event of trouble or a change to an audit configuration file, as role secadmin, at label `admin_high`:**

```
$ audit -s
```

The pointer may be reset to the beginning of the list of audit directories when the administrator enters the `audit -s` command.

## ▼ To Send Audit Records to a New Audit File

- ◆ To change the current audit file for audit records being generated on the workstation, as role `secadmin` at label `admin_high`:

```
$ audit -n filename
```

The new file is created in the same directory as the current file. The directory must be able to contain files labeled `admin_high`.



## Audit Trail Management and Analysis

---

The tools described in this chapter manage the audit files generated on a workstation or on a distributed system. Managing the audit trail involves file tasks and interpretive tasks. File tasks handle disk space issues, such as combining multiple audit files into one and renaming files. Interpretive tasks cover audit analysis, such as selecting audit records based on audit event, user, host machine, and time of day. Sophisticated postprocessing using shell scripts can create auditing reports.

The chapter includes procedures in the following areas:

- “Using the `auditreduce` and `praudit` Commands” on page 86
- “Audit Files Backup and Recovery” on page 93

---

### The Audit Trail

The collection of all audit files in a distributed system is called the *audit trail*. The audit trail may consist of audit files in several audit directories, or an audit directory may contain several audit trails. Most often the audit directories will be separate audit file system partitions. Even though they can be included in other file systems, this is not recommended.

Audit files by default are stored in the *audit root directory*, defined as `/etc/security/audit/*/files`. Once each workstation has created an audit root directory, and the directories have been mounted (with mount points that follow the naming convention) on the audit administration server, the management tools, `auditreduce` and `praudit`, can examine the entire audit trail. See “Basic Audit Setup Procedures” on page 51 for how to set up an audit trail.

Even though it is possible to locate audit directories within other file systems that are not dedicated to auditing, this is not recommended. If other factors dictate placing

audit files on a partition not dedicated to auditing, only do so for directories of last resort. Directories of last resort would be directories where audit files would be written only when there is no other suitable directory available. One other scenario where locating audit directories outside of dedicated audit file systems could be acceptable would be in an environment where auditing is optional, and where it is more important to make full use of disk space than to keep an audit trail. Putting audit directories within other file systems is unworkable in a security-conscious production environment.

## How the Audit Trail Is Created

The *audit trail* is created by the audit daemon, `auditd(1M)`. The audit daemon starts on each workstation when the workstation is booted. After `auditd` starts, it is responsible for collecting the audit trail data and writing the audit records into *audit files*, which are also called *audit log files*. See the `audit.log(4)` man page for a description of the file format.

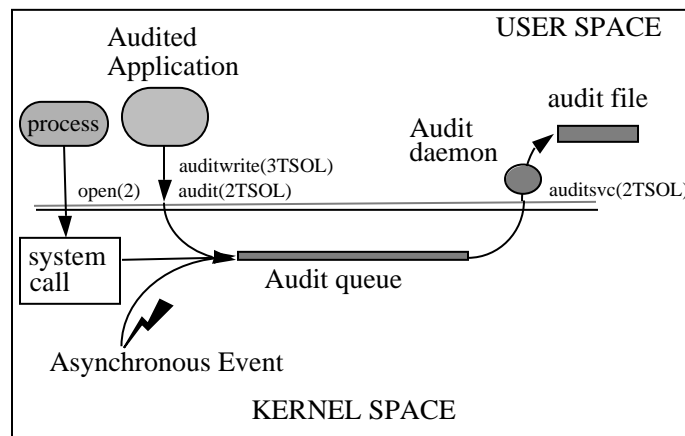


Figure 3-1 How Auditing Works

The audit daemon runs as root. All files it creates are owned by root. Even when `auditd` has no classes to audit, `auditd` continuously operates, looking for a place to put audit records. The `auditd` operations continue even if the rest of the workstation's activities are suspended because the kernel's audit buffers are full. The audit operations can continue because `auditd` is not audited.

---

# Audit Record Format

Audit files consist of self-contained audit records of user-level and kernel-level events that have been preselected for auditing by the security administrator. An *audit record* consists of a sequence of *audit tokens*, each of which describes an attribute of the event being audited. Each auditable event in the system generates a particular type of audit record. The audit record for each event has certain tokens within the record that describe the event. An audit record does not describe the audit event class to which the event belongs; that mapping is determined by an external table, the `/etc/security/audit_event` file.

Each audit token starts with a one-byte token type, followed by one or more data elements in an order determined by the type. The different audit records are distinguished by event type and different sets of tokens within the record. Some tokens, such as the `text` token, contain only a single data element, while others, such as the `process` token, contain several (including the audit user ID, real user ID, and effective user ID).

Audit records are stored and manipulated in binary form; however, the byte order and size of data is predetermined to simplify compatibility between different workstations.

“Audit Token Structure” on page 148, gives a detailed description of each data element in each token and shows sample output. “Audit Records” on page 175 lists all the audit records generated by Trusted Solaris 7 auditing. The records are listed alphabetically by kernel event and by user event. A cross-reference table to the audit records is found in Appendix A.

## Order of Audit Tokens

Each audit record begins with a `header` token and ends (optionally) with a `trailer` token. One or more tokens between the header and trailer describe the event. For user-level and kernel events, the tokens describe the process that performed the event, the objects on which it was performed, and the objects’ attributes, such as the owner or mode.

For example, the `AUE_LSTAT` kernel event, whose audit record is described in Table B-70, has the following tokens:

- `header`
- `path`
- `attribute` (optional)
- `subject`
- `return`

If the `trail` policy has been turned on using the `auditconfig` command, the `trailer` token appears in the audit record after the `return` token.

## Human-Readable Audit Record Format

This section provides examples of audit records in text format. Audit records are stored in binary format. Running the binary records through the `praudit` command produces text output, which can be sent to standard output, a printer, or a scripting program to produce reports. For a complete description of `praudit`, see the `praudit(1M)` man page. For an example of a scripting program, see “To Perform Selections Using a `praudit` Script” on page 92.

## Reading an Audit Token

The following examples of a `header` token show the form that `praudit` produces by default. Examples are also provided of raw (`-r`) and short (`-s`) options.

Every audit record begins with a `header` token. The `header` token gives information common to all audit records. When displayed by `praudit` in default format, a `header` token looks like the following example from `ioctl()`:

```
header,240,1,ioctl(2),Tue Sept 7 16:11:44 1999, + 270000 msec
```

The fields are:

- A token ID, here in text form, **header**
- The record length in bytes, including the `header` and `trailer` tokens, here **240**
- An audit record structure version number, here, version **1**
- An event ID identifying the type of audit event, here in text form, **ioctl(2)**
- An event ID modifier with descriptive information about the event type, here the descriptive field is empty
- The time and date the record was created, here  
**Tue Sept 7 16:11:44 1999, + 270000 msec**

Using `praudit -s`, the event description (`ioctl(2)` in the default `praudit` example above) is replaced with the event name (`AUE_IOCTL`), like this:

```
header,240,1,AUE_IOCTL,Tue Sept 7 16:11:44 1999, + 270000 msec
```

Using `praudit -r`, all fields are displayed as numbers (that may be decimal, octal, or hex), where 20 is the `header` token ID and 158 is the event number for this event.



20,240,1,158,,699754304, + 270000 msec

Note that `praudit` displays the time to millisecond resolution.

## Reading an Audit Record

Every audit record contains at least the `header` token and one other token. For example, the audit record for the audit event `AUE_login` contains five tokens. See Table B-247 for a full description of its audit record format.

When displayed by `praudit` in default format, the audit record for `AUE_login` looks like this, one token per line:

```
header,90,3,login - local,,Tue Jul 8 15:12:01 1997, +520002000 msec,  
text,emily  
text,successful login  
subject,emily,emily,staff,emily,staff,14094,14094,0 0 willet,  
return,success,0  
sequence,17  
trailer,90
```

The tokens are:

- A header token
- A text token (login name)
- A text token (for success or failure)
- A subject token
- A return token

When this audit file collected records, the audit policy tokens `sequence` and `trailer` were turned on, so all audit records including this one contain the following tokens:

- A sequence token
- A trailer token

Note the following features in the audit record:

- Each user's processes is assigned a unique audit ID that stays the same even when the user ID changes (14094)
- Each session has an audit session ID (14094)
- Audit records are self-contained

Because each audit record contains an audit ID that identifies the user who generated the event, and because audit records are self-contained, you can look at individual

audit records and get meaningful information without looking back through the audit trail.

The Trusted Solaris 7 audit records contain all the relevant information about an event and do not require you to refer to other audit records to interpret what occurred. For example, an audit record describing a file event contains the file's full path name starting at the root directory and a time and date stamp of the file's opening or closing.

---

**Note** - You should archive system administration files with audit file archives. Information that is referred to in the audit trail but changes as site personnel and equipment change, such as users and their UIDs, affects your ability to interpret records.

---

Using `praudit -l`, the audit record displays on one line, like this:

```
header,90,3,login - local,,Tue Jul 8 15:12:01 1997, +520002000 msec,text,emily,text,successful
login,subject,emily,emily, staff,emily,staff,14094,14094,0 0 willet,return,success,0,
sequence,17,trailer,90
```

Using `praudit -r` the audit record displays like this:

```
20,90,3,6152,0x0000,872028721,520002000
40,emily
40,successful login
36,6001,6001,10,6001,10,14094,14094,0 0 129.150.110.2
39,0,0
47,17
19,90
```

---

## Audit Files

Each audit file is a self-contained collection of records; the file's name identifies the time span during which the records were generated and the workstation that generated them. The contents of the audit files are binary, protected at the sensitivity label `admin_high`, and accessible in a profile shell only by an administrative role with the Audit Review profile.

# Audit File Naming

Audit files that are complete have names of the following form:

*start-time.finish-time.workstation*

where *start-time* is the time of the first audit record in the audit file, *finish-time* is the time of the last record, and *workstation* is the name of the workstation that generated the file. Some examples of these names can be found in “Example of a Closed Audit File Name ” on page 84.

If the audit log file is still active, it has a name of the following form:

*start-time.not\_terminated.workstation*

## How Audit File Names Are Used

The file name time stamps are used by the `auditreduce` command to locate files containing records for the specific time range that has been requested. This is important because there may be a month's supply or more of audit files online, and searching them all for records generated in the last 24 hours would be expensive.

## Time-Stamp Format and Interpretation

The *start-time* and *finish-time* are time stamps with one-second resolution; they are specified in Greenwich mean time. The format is four digits for the year, followed by two for each month, day, hour, minute, and second, as shown below.

YYYYMMDDHHMMSS

The time stamps are in GMT to ensure that they will sort in proper order even across a daylight saving time boundary. Because they are in GMT, the date and hour must be translated to the current time zone to be meaningful; beware of this whenever

manipulating these files with standard file commands rather than with `auditreduce`.

## Example of a File Name for a Still-Active File

The format of a file name of a still-active file is shown below:

*YYYYMMDDHHMMSS.not\_terminated.hostname*

Here is an example:

*19900327225243.not\_terminated.patchwork*

The audit log files are named by the beginning date, so the example above was started in 1997, on March 27, at 10:52:43 PM, GMT. The `not_terminated` in the file name means either that the file is still active or that `auditd` was unexpectedly interrupted. The name `patchwork` at the end is the host name whose audit data is being collected.

## Example of a Closed Audit File Name

The format of the name of a closed audit log file is shown below:

*YYYYMMDDHHMMSS.YYYYMMDDHHMMSS.hostname*

Here is an example:

The example above was started in 1997, on March 20, at 12:52:43 AM, GMT. The file was closed March 27, at 10:53:51 PM, GMT. The name `patchwork` at the end is the host name of the workstation whose audit data is being collected.

---

## Audit Files Management

Two commands, `praudit(1M)` and `auditreduce(1M)`, enable the audit reviewer to process the Trusted Solaris audit records. The `praudit` command makes the records readable, and the `auditreduce` command enables selecting particular audit records and merging the records into one audit trail.

---

**Note** - The `auditreduce` command can only find records that have been preselected by the security administrator. Events that are not recorded in the audit trail are unavailable to postselection tools.

---

## Merging the Audit Trail

The `auditreduce` command merges audit records from one or more input audit files to create a single, chronologically ordered output file. On a distributed system, the input audit files originate from different workstations. Therefore, when issued from the audit administration server, the `auditreduce` command treats the distributed system as if it were one workstation. This treatment simplifies audit administration. Coupled with backup audit partitions, the distributed system is robust in the face of workstation failures.

The `auditreduce` command also includes options for selecting sets of records to examine. For instance, records from the past 24 hours can be selected to generate a daily report; all records generated by a specific user can be selected to examine that user's activities; or all records caused by a specific event type can be selected to see how often that type occurs.

## Selecting Records from the Audit Trail

Options to the `auditreduce(1M)` command allow you to select audit records based on file characteristics and record characteristics, as shown in the following table.

**TABLE 3-1** Some Options to the `auditreduce` Command

Characteristic	Option(s)
Time, date (start, finish)	-d, -a, -f
Host (workstation) ID	-M, -h, -S
Audit class	-c
Audit event	-m
Audit session ID	
Audit User ID – AUID	-u
Effective and Real User ID – EUID, RUID	-e, -r
Effective and Real Group ID – EGID, RGID	-f, -g
Process ID – PID	-j
Sensitivity label	-s
Information label	-i
Filename	<i>filename</i>

Uppercase options select operations or parameters for *files*, and lowercase options select parameters for *records*. When piped through `praudit`, audit files processed by the `auditreduce` command are readable. Otherwise, they remain in binary format.

The merging and selecting functions of `auditreduce` are logically independent. The `auditreduce` command selects messages from the input files as the records are read, before the files are merged and written to disk.

## Using the `auditreduce` and `praudit` Commands

This section describes a few common uses of `auditreduce` and `praudit` to select and manage data. See the `auditreduce(1M)` man page for more examples.

Prerequisites for running the `auditreduce` and `praudit` commands:

- You are in an administrative role that includes the Audit Review profile. The role `admin` includes this profile by default.
- You are in an `admin_high` workspace of that role.

To create an `admin_high` workspace, see “To Create an Admin\_High Workspace” on page 71 in Chapter 2.

- You have launched a terminal window.

To access the audit trail for a distributed system:

- You issue the `auditreduce` command from the audit administration server.

## To Read a Closed Audit File

The `praudit` command enables you to display audit records interactively and create very basic reports. For multiple files, the input is piped from `auditreduce`.

- ◆ **Specify the audit file as the file argument to the `praudit` command.**

```
$ praudit 19970401000000.19970601000000.grebe
```

This displays audit token per line to standard output.

- ◆ **Specify the audit file as the file argument to the `praudit -l` command.**

```
$ praudit -l 19970401000000.19970601000000.grebe
```

This displays one audit record per line to standard output.

## To Read a Current Audit File

- ◆ **Use the `tail(1)` command to see what is currently being written to an active audit file.**

```
$ praudit | tail -40 19970401000000.not_terminated.grebe
```

This displays the latest 40 tokens that were recorded to standard output.

## To Display Several Audit Files as One Audit File

- ◆ **To display several audit files in chronological order in the terminal window, pipe the output of `auditreduce` into `praudit`.**

```
$ auditreduce 19970413000000.19970413235959.willet \
```

(continued)

```
19970413000000.19970413235959.grebe | praudit
```

- ◆ To display the whole audit trail in the terminal window, pipe the output of `auditreduce` into `praudit`.

```
$ auditreduce | praudit
```

---

**Note** - The `auditreduce` command without options does not disturb open audit files.

---

## To Print an Audit Log

- ◆ Use `praudit` with a pipe to `lp`, to send the output of one file to the printer.

```
$ praudit 19970413000000.19970413235959.audubon | lp
```

- ◆ Use `auditreduce` piped through `praudit` with a pipe to `lp`, to send the output of all closed audit files to the printer.

```
$ auditreduce | praudit | lp
```

---

**Note** - In the Trusted Solaris environment, the printer must be able to accept `admin_high` print jobs.

---

## To Display User Activity on a Selected Date

- ◆ Use the `-d` option to the `auditreduce` command to see audit information collected during a specified 24-hour period.

In the following example, the security administrator checks to see when a user named `doris` logged in and logged out on April 13, 1997, by requesting the `lp`



message class. The short-form date is in the form *yymmdd*. (The long form is described in the `auditreduce(1M)` man page.)

```
$ auditreduce -d 970413 --u doris -c lo | praudit
```

## To Print User Activity on a Selected Date

- ◆ Use the `auditreduce` command with a pipe through `praudit` to `lp`, to send selected output to a printer.

---

**Note** - In the Trusted Solaris environment, the printer must be able to accept `admin_high` print jobs.

---

```
$ auditreduce -d 970413 -u doris -c lo | praudit | lp
```

## To Copy Login/Logout Messages to a Single File

In this example, login/logout messages for a particular day are summarized in a file. The target file is written in a directory other than the normal audit root.

```
$ auditreduce -c lo -d 970413 \  
-O /usr/audit_summary/logins
```

The `-O` option creates an audit file in the `/usr/audit_summary` directory. The file name has 14-character timestamps for both start-time and end-time, and the suffix `logins`:

`/usr/audit_summary/19970413000000.19970413235959.logins`

## To Display Audit Records Created Before or After a Designated Date

The *date-time* options `-b` and `-a` allow specifying records before or after a particular day and time. A day begins at `yyyymmdd00:00:00` and ends at `yyyymmdd23:59:59`. The six parameters of a day are: year, month, day, hour, minute, and second. The digits (19) of the year are assumed and need not be specified.

The `auditreduce -a` command with the date shown in the following screen example sends all audit records created after midnight on July 15, 1997 through `praudit` to standard output.

```
$ auditreduce -a 97071500:00:00 | praudit
```

If `-a` is not specified, `auditreduce` defaults to 00:00:00, January 1, 1970.

The `auditreduce -b` command with the same date shown above sends all audit records created before midnight on July 15, 1997 through `praudit` to standard output.

```
$ auditreduce -b 97071500:00:00 | praudit
```

If `-b` is not specified, `auditreduce` defaults to the current time of day (GMT). The `-d` option selects a particular 24-hour period, as shown in “To Copy Login/Logout Messages to a Single File ” on page 89.

## To Find an Audit Event

- ◆ **Use the message type selection for `auditreduce` (`-m` option) to find a particular audit event.**

The `-m` option accepts either numeric message identifiers or `AUE_XXXXX` event names. The screen example below finds all kernel-level login events in the audit trail and displays them to standard output.

```
$ auditreduce -m AUE_LOGIN | praudit
```

The `auditreduce` command rejects an incorrect format, but does not describe the correct format.

## To Combine Selected Audit Files

Although `auditreduce` can do this type of combination and deletion automatically (see the `-C` and `-D` options in the `auditreduce(1M)` man page), it is often easier to select the files manually (perhaps with `find`) and use the `auditreduce` command to combine just the named set of files.

**1. List the audit files as arguments to the `auditreduce` command.**

In the following example, a recurring job that starts a bit before midnight merges the audit files from two days before. The final time on the file is the time the job ended, here just before midnight, Greenwich Mean Time (GMT).

```
$ auditreduce 19970413000000.19970413235959.grebe \  
19970413000000.19970413235959.willet \  
19970413000000.19970413235959.sora  
$ ls *audubon 19970413000000.19970414235959.audubon
```

**2. Delete the input files and move the output file to the audit root directory on the administration server.**

In this example, the `auditreduce(1M)` command was run on the audit administration server, `audubon`, and then placed in its audit root directory so that future calls to `auditreduce` locate the file.

```
$ rm /etc/security/tsol/grebe/files/19970413000000.19970413235959.grebe  
$ rm /etc/security/tsol/willet/files/19970413000000.19970413235959.willet  
$ rm /etc/security/tsol/sora/files/19970413000000.19970413235959.sora  
$ mv 19970413000000.19970414235959.audubon /etc/security/audit/audubon/files/
```

## To Reduce Audit Files

The `auditreduce` program can also reduce the number of records in its output file by eliminating the less interesting ones as it combines the input files.

You might use `auditreduce` to eliminate all except the login/logout events in audit files over a month old, assuming that if you needed to retrieve the complete audit trail you could recover it from backup tapes. The following example selects just the audit records from April 1997.

```
$ auditreduce -m AUE_LOGIN -a 19970401000000 \  
-b 19970501000000 \  
-O /usr/audit_summary/logins_april97
```

The output is a smaller file containing just the April 1997 login/logout records. Note that the end-time stamp is the date (in GMT) that the command was run (June 1, 1997), not the last date of the merged records. You specified the file suffix, `logins_april97`, on the command line with the directory name.

```
/usr/audit_summary/19970401000000.19970601000000.logins_april97
```

## To Change the `praudit` Field Separator to a Tab

When the `praudit` command displays an audit token, it separates the data fields with commas. However, if a field (such as a time stamp) contains a comma, this cannot be distinguished from a field-separating comma.

- ◆ **Press the Tab key as the value of the `-d` option to `praudit(1M)`.**

```
$ praudit -d"<press Tab key>" 19970413120429.19970413180433.grebe
```

There is no space between the `-d` option and the delimiter. Surround the delimiter with double quotes. The delimiter can be up to four characters long.

## To Change the `praudit` Token Separator to a Tab

Audit tokens are separated by newlines by default. When audit records are printed one per line using the `-l` option, the audit token separator is the same as the audit field separator. In the following screen example, the audit tokens are separated by tabs, as are the audit fields.

```
$ praudit -l -d"<press Tab key>" 19970413120429.19970413180433.grebe
```

## To Perform Selections Using a `praudit` Script

To accomplish more sophisticated display and reports, process the output from `praudit` with `sed` or `awk`, or write programs to interpret and process the binary audit records.

It is sometimes useful to manipulate `praudit` output as lines of text; for example to perform selections that cannot be done with `auditreduce`. A simple shell script can process the output of `praudit`. The following example is called `praudit_grep`:

```
#!/bin/sh
praudit | sed -e '1,2d' -e '$s/^file.*$//' -e 's/^header/^aheader/' \
| tr '\012\001' '\002\012' \
| grep "$1" \
| tr '\002' '\012'
```

The example script marks the header tokens by prefixing them with Control-A. (Note that the ^a is Control-A, not the two characters ^ and a. Prefixing is necessary to distinguish them from the string header that might appear as text.) The script then combines all the tokens for a record onto one line while preserving the line breaks as Control-A, runs the `grep` command, and restores the original newlines.

To run the script in the Trusted Solaris environment, the following conditions must be met:

- The script exists in an `admin_low` directory (to make it visible to the Profile Manager).
- The security administrator has added the script to the appropriate profile (such as Custom Admin Role), and given it the forced privileges:
- The security administrator has added any commands in the script that are not in the role's profile to the appropriate profile.
- The admin role runs the script in an `admin_high` profile shell in a directory where the admin role has write access.

---

## Audit Files Backup and Recovery

Audit files occupy disk space. The disk space needs to be freed up in order to make space for subsequent audit files. By default, the role `oper` handles audit file backup via the profile Media Backup and the role `admin` handles audit file restore via the profile Media Restore.

### ▼ To Back Up Audit Files

1. **As the role `oper` in an `admin_high` workspace, go to the workstation's audit files directory.**

```
$ cd /etc/security/audit/workstation_name[n]/files
```

2. **Allocate, at the label `admin_high`, the tape drive that you are going to use for backup.**

If you are unfamiliar with device allocation, see “To Allocate and Deallocate Devices” on page 65.

3. Use the `tar(1)` command to copy the completed audit files and their Trusted Solaris security attributes, such as the label, to the tape.

For example,

```
$ tar cvT \  
/etc/security/audit/workstation_name/files/19980413120429.19980413180433.grebe \  
/etc/security/audit/workstation_name/files/19980502120429.19980502180433.grebe \  
/etc/security/audit/workstation_name/files/19980513120429.19980513180433.grebe
```

4. Deallocate the tape drive when finished, remove the tape, and label it `admin_high`.
5. At the same time, in an `admin_low` workspace, back up system files that capture information about the users, labels, roles, and execution profiles on the workstation.

Store the audit tapes with the current system information tape(s).

6. As root, at label `admin_high`, remove the audit files that have been backed up.

For example,

```
$ rm \  
/etc/security/audit/workstation_name/files/19980413120429.19980413180433.grebe \  
/etc/security/audit/workstation_name/files/19980502120429.19980502180433.grebe \  
/etc/security/audit/workstation_name/files/19980513120429.19980513180433.grebe
```

## ▼ To Restore Audit Files

1. As role `admin`, in an `admin_high` workspace, go to the directory where the audit files are to be placed.

```
$ cd /etc/security/audit/workstation_name[n]/reports
```

2. **Allocate, at the label `admin_high`, the tape drive that you are going to use to restore the files.**

If you are unfamiliar with device allocation, see “To Allocate and Deallocate Devices” on page 65.

3. **Use the `tar(1)` command to copy the audit files and their Trusted Solaris security attributes, such as the label, from the tape.**

For example,

```
$ tar xvT \  
/etc/security/audit/workstation_name/files/19980513120429.19980513180433.grebe
```

4. **Deallocate the tape drive when finished and follow the Device Manager’s instructions.**

5. **Use the restored audit files.**

You may need to restore or refer to other system information from the audit backup’s associated system backup.

6. **As role `admin`, at label `admin_high`, remove the audit files when you are done.**

```
$ rm \  
/etc/security/audit/workstation_name/reports/19980513120429.19980513180433.grebe
```





## Troubleshooting Auditing

---

Another auditing task is to handle audit anomalies as they occur. Typical tasks that audit analysts and system administrators face are discussed below.

- “Preventing Audit Trail Overflow” on page 97
- “Cleaning up an Audit File Marked `not_terminated`” on page 99
- “Using the `sequence` Token for Debugging” on page 100
- “Starting the Audit Daemon Manually” on page 102
- “Workstations are Being Audited Differently” on page 102
- “Finding Failed Login Attempts” on page 103

---

## Preventing Audit Trail Overflow

When all audit file systems for a workstation fill up, the `audit_warn` script sends a message to the console that the hard limit has been exceeded on all audit file systems and also sends mail to the alias. By default, the audit daemon remains in a loop sleeping and checking for space until some space is freed. All auditable actions are suspended. The audit policy `ahlt` is in effect.

Site security policy may allow a different solution. There are other candidates: preventing overflow and keeping a count of dropped audit records.

If your security policy requires that overflow be prevented so that no audit data is ever lost, see “To Prevent Audit Trail Overflow by Planning Ahead” on page 98.

---

**Note** - The audit system can be configured to discard audit records upon overflow of the kernel audit buffer. Such a configuration does not constitute an evaluated configuration of the system, and the system should be configured to suspend upon overflow of the audit buffer.

---

If your security policy permits the loss of some audit data rather than suspending system activities due to audit trail overflow. In that case, you can set the `auditconfig` policy to drop or count records. See “To Handle an Audit Filesystem Overflow” on page 99 for how to drop or count records.

If your security policy requires you to handle filesystem overflow by halting the affected workstation, you must enter the workstation in single-user mode. This is not a secure practice. See “To Handle an Audit Filesystem Overflow” on page 99 for the procedure.

## ▼ To Prevent Audit Trail Overflow by Planning Ahead

If your security policy requires that all audit data be saved, do the following:

1. **Set up a schedule to regularly archive audit files and to delete the archived audit files from all audit file systems.**

The schedule must allow files to be deleted from the system before the hard limit of the system is reached. Scripts, including modified `audit_warn` scripts, can automatically move audit files to a separate disk before archiving.

2. **Manually archive audit files by backing them up on tape or moving them to an archive file system.**

3. **Store context-sensitive information that will be needed to interpret audit records along with the audit trail.**

For example, the current list of users and passwords, the directory listings on the workstations, and other volatile information should be saved.

4. **Keep records of what audit files are moved off line.**

5. **Store the archived tapes appropriately.**

6. **Reduce the volume of audit data you store by creating summary files.**

You can extract summary files from the audit trail using options to `auditreduce`, so that the summary files contain only records for certain specified types of audit events. An example of this would be a summary file containing only the audit records for all logins and logouts. See Chapter 3.

## ▼ To Handle an Audit Filesystem Overflow

- ◆ To set the audit policy that a count of audit records is kept when the audit file systems are full, as role `secadmin`, at label `admin_low`:

```
$ auditconfig -setpolicy +cnt
```



---

**Caution** - To run auditing in an evaluated configuration, you cannot have the `+cnt` policy turned on. It *must* be turned off.

---

- ◆ To set the audit policy that the workstation is shut down when its audit file systems are full:

```
$ auditconfig -setpolicy +ahlt
```

To set one of the above policies permanently, enter the command in the `audit_startup(1M)` script. See “To Set Audit Policy Permanently” on page 62 for how to edit the script.

---

**Note** - On a distributed system, the same audit policy should be applied to all workstations.

---

---

## Cleaning up an Audit File Marked `not_terminated`

Occasionally, if an audit daemon dies while its audit file is still open, or a server becomes inaccessible and forces the workstation to switch to a new server, an audit file remains in which the end-time in the file name remains the string `not_terminated`, even though the file is no longer used for audit records.

The `auditreduce(1M)` command processes files marked `not_terminated`, but because such files may contain incomplete records at the end, future processing may generate errors. To avoid errors, clean the incomplete file with the `-O` option of `auditreduce`. This creates a new file containing all the records that were in the old one, but with a proper file name time stamp. This operation loses the previous file pointer that's kept at the beginning of each audit file.

## ▼ To Clean Up a not\_terminated Audit File

1. As role **admin**, at label **admin\_high** check the `/etc/security/audit_data` file to determine the current process number of the audit daemon.  
If that process is still running, and if the file name in `audit_data(4)` is the same as the file in question, do not clean the file.
2. Issue the command `auditreduce` with the `-O` (capital o) option.
3. Provide the workstation name as the argument to `-O`, and the incomplete file name. To delete the original record, use the `-D` option.

```
$ auditreduce -O workstation 19970413120429.not_terminated.workstation
```

This creates a new audit file with the correct name, cleans up pointers to other files, and copies all the records to the new file. The end-time is the time when the command was executed; the correct suffix is *workstation*, explicitly specified.

4. If you did not use the `-D` option, verify that the new file contains the original file's records, then delete the original file.

```
$ ls -l 19970413120429*.workstation
$ rm 19970413120429.not_terminated*
```

---

## Using the sequence Token for Debugging

When an audit trail created from merging records from several workstations appears to have the records listed out of order, you can debug the audit trail discrepancies using the sequence token. Since the sequence token is not recorded by default, the security administrator adds it to the audit policy. The audit policy must be set identically on all workstations contributing to the audit trail.

When the audit trail has been debugged, the security administrator removes the token.

## ▼ To Add the sequence Token to the Audit Record

1. To add the `seq` audit policy dynamically, as role `secadmin`, at label `admin_low`, on the command line:

```
$ auditconfig -setpolicy +seq
$ auditconfig -getpolicy
slabel, seq
```

2. To add the `seq` audit policy permanently, as role `secadmin` at label `admin_low`, in the `audit_startup` file:

```
#!/bin/sh
auditconfig -setpolicy +slabel, seq
```

## ▼ To Prevent the sequence Token from Being Part of Audit Records

1. To remove the `seq` audit policy dynamically, on the command line, as role `secadmin` at label `admin_low`:

```
$ auditconfig -setpolicy -seq
$ auditconfig -getpolicy
slabel
```

2. To remove the `seq` audit policy from the `audit_startup` file, as role `secadmin` at label `admin_low`:

```
#!/bin/sh
```

```
auditconfig -setpolicy +slabel
```

---

## Starting the Audit Daemon Manually

On a distributed system, if many workstations have lost their audit daemon, bring up the audit daemons in order.

- ♦ **As role secadmin, execute the command `/usr/sbin/auditd` in an `admin_high` shell on the audit administration server, then on the audit servers, and finally on the audit clients.**

```
$ /usr/sbin/auditd
```

If you are unfamiliar with creating an `admin_high` shell, see “To Create an Admin\_High Workspace” on page 71.

---

## Workstations are Being Audited Differently

If you change audit configuration files on one workstation and fail to copy the files to the other workstations on the network, the workstations will be audited differently. Therefore,

1. **As role secadmin, at label `admin_low`, copy the audit configuration files from a central location to every workstation.**

Follow the procedure in “To Distribute Audit Configuration Files to a Network of Workstations” on page 63.

2. **Check that the audit class mappings for attributable and nonattributable events match the kernel cache.**

## ▼ To Set Audit Class Mappings for Attributable Events

1. First, as role `secadmin` at label `admin_low`, check to see if the kernel preselection mask matches the class mappings in the `flags:` field of the `audit_control(4)` file by issuing the command:

```
$ auditconfig -chkconf
```

2. If the runtime class mappings differ from the kernel cache, issue the command:

```
$ auditconfig -conf
```

## ▼ To Set Audit Class Mappings for Non-Attributable Audit Events

1. First, as role `secadmin` at label `admin_low`, check to see if the kernel preselection mask matches the nonattributable events in the `naflags:` field of the `audit_control(4)` file by issuing the command:

```
$ auditconfig -getkmask
```

- ♦ If they differ, issue the command:

```
$ auditconfig -setkmaskac
```

---

## Finding Failed Login Attempts

- ♦ As role `admin` at label `admin_high`, enter `-lo` as the value of the `-c` option to `auditreduce(1M)`.

```
$ auditreduce -c -lo -O /usr/audit_summary/logins_failed
```

The value “-lo” is the audit flag for failed (-) login (audit class lo) attempts. The command produces a binary file in the /usr/audit\_summary directory with all failed login attempts on the distributed system. The /usr/audit\_summary directory is labeled admin\_high.

/usr/audit\_summary/19970313120429.19970613120415.logins\_failed

---

**Note** - This command works only if the security administrator has preselected failed logins for the workstation, distributed system, or users.

---



## Event-to-Class Mappings

---

This appendix lists audit events by audit class. See the file `/etc/security/audit_event` for a list of events by audit event number.

---

## Audit Events Listed by Audit Class

The Trusted Solaris environment provides the audit classes listed alphabetically by Short Name in the following table. The classes are listed in the file `/etc/security/audit_class`.

**TABLE A-1** Trusted Solaris Audit Classes (Default)

Short Name	Long Name	Audit Mask	List of Events per Class
aa	Audit administration	0x00040000	Table A-2
ad	Administrative	0x000f0000	No predefined audit events.
ao	Other administration	0x00080000	Table A-3
all	All classes	0xffffffff	
ap	Application	0x00004000	Table A-4

**TABLE A-1** Trusted Solaris Audit Classes (Default) *(continued)*

Short Name	Long Name	Audit Mask	List of Events per Class
as	System-wide administration	0x00020000	Table A-22
ax	X server	0x00002000	Table A-24
cl	File close	0x00000040	Table A-5
fa	File attribute access	0x00000004	Table A-6
fc	File create	0x00000010	Table A-7
fd	File delete	0x00000020	Table A-8
fm	File attribute modify	0x00000008	Table A-9
fn	Fcntl	0x40000000	Table A-10
fr	File read	0x00000001	Table A-11
fw	File write	0x00000002	Table A-12
il	Information label	0x00010000	Obsolete.
io	Ioctl	0x20000000	Table A-13
ip	Ipc	0x00000200	Table A-14
lo	Login or logout	0x00001000	Table A-15
na	Non-attribute	0x00000400	Table A-16

**TABLE A-1** Trusted Solaris Audit Classes (Default) *(continued)*

Short Name	Long Name	Audit Mask	List of Events per Class
no	Invalid class	0x00000000	Table A-17
nt	Network	0x00000100	Table A-18
ot	Other	0x80000000	Table A-19
pc	Process	0x00300000	No predefined audit events.
pm	Process modify	0x00200000	Table A-20
ps	Process start/stop	0x00100000	Table A-21
ss	Change system state	0x00010000	Table A-23
xa	X - Allowed information flows	0x40000000	Table A-25
xc	X - Object create/destroy	0x20000000	Table A-26
xl	X - Client login/logout	0x08000000	Table A-27
xp	X - Privileged operations	0x10000000	Table A-28
xs	X - Operations that fail silently	0x80000000	Table A-29
xx	X - All X events	0xf8000000	See the individual X classes.

For more information about the classes, especially the X server classes, see the `audit_class(4)` man page.

## Events in Audit Class aa

The following table lists in alphabetical order the `audit` administration class of audit events provided in the Trusted Solaris 7 release.

**TABLE A-2** Audit Administration Audit Events (Default)

Audit Event Number and Event		Where Described
224	AUE_AUDITON_GETCAR	Table B-9
231	AUE_AUDITON_GETCLASS	Table B-10
229	AUE_AUDITON_GETCOND	Table B-11
223	AUE_AUDITON_GETCWD	Table B-12
221	AUE_AUDITON_GETKMASK	Table B-13
225	AUE_AUDITON_GETSTAT	Table B-14
141	AUE_AUDITON_GPOLICY	Table B-15
145	AUE_AUDITON_GQCTRL	Table B-16
139	AUE_AUDITON_GTERMID	No longer supported.
144	AUE_AUDITON_SESTATE	No longer supported.
232	AUE_AUDITON_SETCLASS	Table B-17
230	AUE_AUDITON_SETCOND	Table B-18
222	AUE_AUDITON_SETKMASK	Table B-19
228	AUE_AUDITON_SETSMASK	Table B-20

**TABLE A-2** Audit Administration Audit Events (Default) *(continued)*

Audit Event Number and Event		Where Described
226	AUE_AUDITON_SETSTAT	Table B-21
227	AUE_AUDITON_SETUMASK	Table B-22
142	AUE_AUDITON_SPOLICY	Table B-23
146	AUE_AUDITON_SQCTRL	Table B-24
140	AUE_AUDITON_STERMID	No longer supported.
529	AUE_AUDITPSA	Table B-25
150	AUE_AUDITSTAT	Table B-26
136	AUE_AUDITSVC	Table B-27
530	AUE_FAUDITPSA	Table B-40
132	AUE_GETAUDIT	Table B-51
130	AUE_GETAUID	Table B-52
147	AUE_GETKERNSTATE	No longer supported.
149	AUE_GETPORTAUDIT	Table B-62
134	AUE_GETUSERAUDIT	No longer supported.
133	AUE_SETAUDIT	Table B-130
131	AUE_SETAUID	Table B-131

**TABLE A-2** Audit Administration Audit Events (Default) *(continued)*

Audit Event Number and Event		Where Described
148	AUE_SETKERNSTATE	No longer supported.
135	AUE_SETUSERAUDIT	No longer supported.
9016	AUE_audit	Table B-225
9015	AUE_auditwrite	Table B-226

## Events in Audit Class ao

The following table lists in alphabetical order the other administration class of audit events provided in the Trusted Solaris 7 release.

**TABLE A-3** Administrative Other Audit Events (Default)

Audit Event Number and Event		Where Described
61	AUE_EXPORTFS	Table B-279
9	AUE_MKNOD	Table B-74
62	AUE_MOUNT	Table B-81
115	AUE_NFSSVC_EXIT	No longer supported.
58	AUE_NFS_GETFH	No longer supported.
53	AUE_NFS_SVC	No longer supported.
60	AUE_QUOTACTL	No longer supported.
12	AUE_UMOUNT	Table B-165

**TABLE A-3** Administrative Other Audit Events (Default) *(continued)*

Audit Event Number and Event		Where Described
56	AUE_UNMOUNT	No longer supported.
233	AUE_UTSSYS	Table B-168
6200	AUE_allocate_succ	Table B-218
6201	AUE_allocate_fail	Table B-219
6144	AUE_at_create	Table B-222
6145	AUE_at_delete	Table B-223
6146	AUE_at_perm	Table B-224
9034	AUE_automountd_mismatch	Table B-227
9033	AUE_automountd_mount	Table B-228
9029	AUE_chroot_cmd	Table B-229
6147	AUE_cron_invoke	Table B-232
6148	AUE_crontab_create	Table B-230
6149	AUE_crontab_delete	Table B-231
6150	AUE_crontab_perm	Table B-233
6202	AUE_deallocate_succ	Table B-235
6203	AUE_deallocate_fail	Table B-236

**TABLE A-3** Administrative Other Audit Events (Default) *(continued)*

<b>Audit Event Number and Event</b>		<b>Where Described</b>
9319	AUE_dm_add	Table B-234
9320	AUE_dm_del	
9321	AUE_dm_mod	
9031	AUE_fuser	Table B-240
9307	AUE_gm_add_grp	Table B-241
9308	AUE_gm_del_grp	
9309	AUE_gm_mod_grp	
9310	AUE_hm_add_host	Table B-243
9311	AUE_hm_del_host	
9312	AUE_hm_mod_host	
9313	AUE_hm_set_def	
6205	AUE_listdevice_succ	Table B-220
6206	AUE_listdevice_fail	Table B-221
9044	AUE_lp_cancel	Table B-253
9045	AUE_lp_status	
9009	AUE_pfsh_priv	Table B-259
9008	AUE_pfsh_trusted_nopriv	
9007	AUE_pfsh_trusted_priv	



**TABLE A-3** Administrative Other Audit Events (Default) *(continued)*

Audit Event Number and Event		Where Described
9316	AUE_pm_add_prn	Table B-262
9318	AUE_pm_del_prn	
9317	AUE_pm_mod_prn	
9306	AUE_pm_add_prof	Table B-263
9304	AUE_pm_del_prof	
9305	AUE_pm_mod_prof	
9014	AUE_sendmail_defer	Table B-275
9013	AUE_sendmail_deliver	Table B-275
9012	AUE_sendmail_upgrade	Table B-276
9315	AUE_sm_del_ser	Table B-277
9314	AUE_sm_mod_ser	
9017	AUE_uauth	Table B-284
		Table B-251
		Table B-252
9322	AUE_te_modsysfiles	Table B-217

**TABLE A-3** Administrative Other Audit Events (Default) *(continued)*

Audit Event Number and Event		Where Described
9302	AUE_um_add_user	Table B-285
9301	AUE_um_del_user	
9300	AUE_um_mod_user	
9303	AUE_um_set_def	
9024	AUE_uname_set	Table B-286

## Events in Audit Class ap

The following table lists in alphabetical order the `application` class of audit events provided in the Trusted Solaris 7 release.

**TABLE A-4** Application Audit Events (Default)

Audit Event Number and Event		Where Described
9010	AUE_pfsh_nopriv	Table B-259
9035	AUE_sl_change	Table B-280

## Events in Audit Class cl

The following table lists in alphabetical order the `file close` class of audit events provided in the Trusted Solaris 7 release.

**TABLE A-5** File Close Audit Events (Default)

Audit Event Number and Event	Where Described
112 AUE_CLOSE	Table B-34
213 AUE_MUNMAP	Table B-89

## Events in Audit Class fa

The following table lists in alphabetical order the file attribute access class of audit events provided in the Trusted Solaris 7 release.

**TABLE A-6** File Attribute Access Audit Events (Default)

Audit Event Number and Event	Where Described
14 AUE_ACCESS	Table B-5
220 AUE_AUDITSYS	Placeholder
66 AUE_BSMSYS	Placeholder
543 AUE_FGETCMWLABEL	Table B-54
55 AUE_FSTATFS	Table B-50
545 AUE_GETCMWFSRANGE	Table B-53
546 AUE_GETCMWLABEL	Table B-54
547 AUE_GETFILEPRIV	Table B-56
554 AUE_GETMLDADORN	Table B-57

**TABLE A-6** File Attribute Access Audit Events (Default) *(continued)*

Audit Event Number and Event		Where Described
555	AUE_GETSLDNAME	Table B-65
548	AUE_LGETCMWLABEL	Table B-54
17	AUE_LSTAT	Table B-70
236	AUE_LXSTAT	Table B-71
556	AUE_MLDLSTAT	Obsolete. Changed to library routines in Trusted Solaris 7.
557	AUE_MLDSTAT	
64	AUE_MSGSYS	Placeholder
3	AUE_OPEN	Placeholder
199	AUE_OSTAT	No longer supported.
71	AUE_PATHCONF	Table B-103
67	AUE_RFSSYS	Placeholder
63	AUE_SEMSYS	Placeholder
65	AUE_SHMSYS	Placeholder
16	AUE_STAT	Table B-154
54	AUE_STATFS	
234	AUE_STATVFS	

**TABLE A-6** File Attribute Access Audit Events (Default) *(continued)*

Audit Event Number and Event	Where Described
70 AUE_VPIXSYS	Placeholder
235 AUE_XSTAT	Table B-174

## Events in Audit Class fc

The following table lists in alphabetical order the `file create` class of audit events provided in the Trusted Solaris 7 release.

**TABLE A-7** File Create Audit Events (Default)

Audit Event Number and Event	Where Described
8 AUE_CHDIR	Table B-28
111 AUE_CORE	Table B-108
4 AUE_CREAT	Table B-35
532 AUE_FGETSLDNAME	Table B-46
5 AUE_LINK	Table B-69
47 AUE_MKDIR	Table B-73
73 AUE_OPEN_RC	Table B-92
75 AUE_OPEN_RTC	Table B-93
81 AUE_OPEN_RWC	Table B-96

**TABLE A-7** File Create Audit Events (Default) *(continued)*

Audit Event Number and Event	Where Described
83 AUE_OPEN_RWTC	Table B-97
77 AUE_OPEN_WC	Table B-100
79 AUE_OPEN_WTC	Table B-101
42 AUE_RENAME	Table B-114
48 AUE_RMDIR	Table B-115
21 AUE_SYMLINK	Table B-156
240 AUE_XMKNOD	Table B-173

## Events in Audit Class fd

The following table lists in alphabetical order the `file delete` class of audit events provided in the Trusted Solaris 7 release.

**TABLE A-8** File Delete Audit Events (Default)

Audit Event Number and Event	Where Described
44 AUE_FTRUNCATE	No longer supported.
74 AUE_OPEN_RT	Table B-94
75 AUE_OPEN_RTC	Table B-93
82 AUE_OPEN_RWT	Table B-98

**TABLE A-8** File Delete Audit Events (Default) *(continued)*

Audit Event Number and Event	Where Described
83 AUE_OPEN_RWTC	Table B-97
78 AUE_OPEN_WT	Table B-102
79 AUE_OPEN_WTC	Table B-101
42 AUE_RENAME	Table B-114
6 AUE_UNLINK	Table B-166

## Events in Audit Class fm

The following table lists in alphabetical order the file attribute modify class of audit events provided in the Trusted Solaris 7 release.

**TABLE A-9** File Attribute Modify Audit Events (Default)

Audit Event Number and Event	Where Described
251 AUE_ACLSET	Table B-129
11 AUE_CHOWN	Table B-30
252 AUE_FACLSET	Table B-129
39 AUE_FCHMOD	Table B-42
38 AUE_FCHOWN	Table B-43
45 AUE_FLOCK	Placeholder

**TABLE A-9** File Attribute Modify Audit Events (Default) *(continued)*

Audit Event Number and Event		Where Described
544	AUE_FSETCMWLABEL	Table B-133
523	AUE_FSETFATTRFLAG	Table B-49
158	AUE_IOCTL	Table B-66
237	AUE_LCHOWN	Table B-68
525	AUE_LSETCMWLABEL	Table B-133
19	AUE_MCTL	No longer supported.
524	AUE_MLDSETFATTRFLAG	Table B-75
542	AUE_SETCLEARANCE	Table B-132
549	AUE_SETCMWLABEL	Table B-133
541	AUE_SETCMWPLABEL	Table B-134
522	AUE_SETFATTRFLAG	Table B-137
550	AUE_SETFILEPRIV	Table B-138
551	AUE_SETPROCPRIV	Table B-142
202	AUE_UTIME	Table B-167
49	AUE_UTIMES	



**TABLE A-9** File Attribute Modify Audit Events (Default) *(continued)*

Audit Event Number and Event		Where Described
552	AUE_WRITE	Table B-172
553	AUE_WRITEV	
9037	AUE_dtfiler_copy	Table B-238
9038	AUE_dtfiler_move	Table B-238

## Events in Audit Class fn

The following table lists in alphabetical order the `fcntl` class of audit events provided in the Trusted Solaris 7 release.

**TABLE A-10** Fcntl Audit Events (Default)

Audit Event Number and Event		Where Described
30	AUE_FCNTL	Table B-45

## Events in Audit Class fr

The following table lists in alphabetical order the `file read` class of audit events provided in the Trusted Solaris 7 release.

**TABLE A-11** File Read Audit Events (Default)

Audit Event Number and Event		Where Described
72	AUE_OPEN_R	Table B-91
73	AUE_OPEN_RC	Table B-92
74	AUE_OPEN_RT	Table B-94
75	AUE_OPEN_RTC	Table B-93
80	AUE_OPEN_RW	Table B-95
81	AUE_OPEN_RWC	Table B-96
82	AUE_OPEN_RWT	Table B-98
83	AUE_OPEN_RWTC	Table B-97
22	AUE_READLINK	Table B-113

## Events in Audit Class fw

The following table lists in alphabetical order the `file read` class of audit events provided in the Trusted Solaris 7 release.

**TABLE A-12** File Write Audit Events (Default)

Audit Event Number and Event	Where Described
80 AUE_OPEN_RW	Table B-95
81 AUE_OPEN_RWC	Table B-96
82 AUE_OPEN_RWT	Table B-98
83 AUE_OPEN_RWTC	Table B-97
76 AUE_OPEN_W	Table B-99
77 AUE_OPEN_WC	Table B-100
78 AUE_OPEN_WT	Table B-102
79 AUE_OPEN_WTC	Table B-101

## Events in Audit Class io

The following table lists the audit event in the `ioctl` class provided in the Trusted Solaris 7 release.

**TABLE A-13** Ioctl Audit Events (Default)

Audit Event Number and Event	Where Described
158 AUE_IOCTL	Table B-66

# Events in Audit Class ip

The following table lists in alphabetical order the `ipc` class of audit events provided in the Trusted Solaris 7 release.

**TABLE A-14** IPC Audit Events (Default)

Audit Event Number and Event		Where Described
514	AUE_GETMSGQCMWLABEL	Table B-60
515	AUE_GETSEMCMWLABEL	Table B-63
516	AUE_GETSHMCMWLABEL	Table B-64
84	AUE_MSGCTL	Illegal command
85	AUE_MSGCTL_RMID	Table B-82
86	AUE_MSGCTL_SET	Table B-83
87	AUE_MSGCTL_STAT	Table B-84
88	AUE_MSGGET	Table B-85
174	AUE_MSGGETL	Table B-86
89	AUE_MSGRCV	Table B-87
175	AUE_MSGRCVL	Table B-87
90	AUE_MSGSND	Table B-88
176	AUE_MSGSNDL	Obsolete.
98	AUE_SEMCTL	Illegal command

**TABLE A-14** IPC Audit Events (Default) *(continued)*

Audit Event Number and Event	Where Described
105 AUE_SEMCTL_GETALL	Table B-116
102 AUE_SEMCTL_GETNCNT	Table B-117
103 AUE_SEMCTL_GETPID	Table B-118
104 AUE_SEMCTL_GETVAL	Table B-119
106 AUE_SEMCTL_GETZCNT	Table B-120
99 AUE_SEMCTL_RMID	Table B-121
100 AUE_SEMCTL_SET	Table B-122
108 AUE_SEMCTL_SETALL	Table B-123
107 AUE_SEMCTL_SETVAL	Table B-124
101 AUE_SEMCTL_STAT	Table B-125
109 AUE_SEMGET	Table B-126
177 AUE_SEMGETL	Table B-127
110 AUE_SEMOP	Table B-128
517 AUE_SEMOPL	Obsolete.
96 AUE_SHMAT	Table B-147
91 AUE_SHMCTL	Placeholder

**TABLE A-14** IPC Audit Events (Default) *(continued)*

Audit Event Number and Event		Where Described
92	AUE_SHMCTL_RMID	Table B-148
93	AUE_SHMCTL_SET	Table B-149
94	AUE_SHMCTL_STAT	Table B-150
97	AUE_SHMDT	Table B-151
95	AUE_SHMGET	Table B-152
178	AUE_SHMGETL	Table B-153

## Events in Audit Class lo

The following table lists in alphabetical order the login or logout class of audit events provided in the Trusted Solaris 7 release.

**TABLE A-15** Login or Logout Audit Events (Default)

Audit Event Number and Event		Where Described
6165	AUE_ftp	Table B-245
6152	AUE_login	Table B-247
6153	AUE_logout	Table B-250
6163	AUE_passwd	Table B-257

**TABLE A-15** Login or Logout Audit Events (Default) *(continued)*

Audit Event Number and Event	Where Described
6164 AUE_rexd	Table B-269
6162 AUE_rexecd	Table B-270
6155 AUE_rlogin	Table B-248
6158 AUE_rshd	Table B-271
6159 AUE_su	Table B-281
6154 AUE_telnet	Table B-249

## Events in Audit Class na

The following table lists in alphabetical order the non-attribute class of audit events provided in the Trusted Solaris 7 release.

**TABLE A-16** Non-attribute Audit Events (Default)

Audit Event Number and Event	Where Described
153 AUE_ENTERPROM	Table B-37
154 AUE_EXITPROM	
113 AUE_SYSTEMBOOT	Table B-158
6151 AUE_inetd_connect	Table B-244

**TABLE A-16** Non-attribute Audit Events (Default) *(continued)*

Audit Event Number and Event		Where Described
6156	AUE_mountd_mount	Table B-255
6157	AUE_mountd_umount	Table B-256

## Events in Audit Class no

The following table lists in alphabetical order the `invalid class` class of audit events provided in the Trusted Solaris 7 release.

**TABLE A-17** Invalid Class Audit Events (Default)

Audit Event Number and Event		Where Described
211	AUE_AUDIT	Table B-8
209	AUE_DUP2	No longer supported.
208	AUE_FSTAT	Table B-50
193	AUE_GETDENTS	Table B-55
13	AUE_JUNK	
194	AUE_LSEEK	Placeholder
518	AUE_MAC	No longer supported.
210	AUE_MMAP	Table B-76
242	AUE_MODCTL	Placeholder
197	AUE_NFS	Placeholder



**TABLE A-17** Invalid Class Audit Events (Default) *(continued)*

Audit Event Number and Event	Where Described
0 AUE_NULL	Indirect system call
185 AUE_PIPE	Table B-104
527 AUE_PREADL	Table B-105
528 AUE_PWRITEL	Table B-172
192 AUE_READ	Table B-112
558 AUE_READL	
198 AUE_READV	Placeholder
559 AUE_READVL	Table B-112
189 AUE_RECV	Placeholder
187 AUE_SEND	Placeholder
186 AUE_SOCKETPAIR	Placeholder
521 AUE_UPRIV	Table B-175
195 AUE_WRITE	Table B-171
196 AUE_WRITEV	Placeholder

## Events in Audit Class nt

The following table lists in alphabetical order the `network` class of audit events provided in the Trusted Solaris 7 release.

**TABLE A-18** Network Audit Events (Default)

Audit Event Number and Event		Where Described
33	AUE_ACCEPT	No longer supported.
34	AUE_BIND	No longer supported.
32	AUE_CONNECT	No longer supported.
217	AUE_GETMSG	Table B-58
219	AUE_GETPMSG	Table B-61
173	AUE_ONESIDE	No longer supported.
216	AUE_PUTMSG	Table B-109
218	AUE_PUTPMSG	Table B-111
191	AUE_RECVFROM	No longer supported.
190	AUE_RECVMSG	No longer supported.
188	AUE_SENDMSG	No longer supported.
184	AUE_SENDTO	No longer supported.
35	AUE_SETSOCKOPT	No longer supported.
247	AUE_SOCKACCEPT	Table B-59
248	AUE_SOCKCONNECT	Table B-110

**TABLE A-18** Network Audit Events (Default) *(continued)*

Audit Event Number and Event	Where Described
183 AUE_SOCKET	No longer supported.
250 AUE_SOCKRECEIVE	Table B-59
249 AUE_SOCKSEND	Table B-110
534 AUE_TNIF	Table B-159
535 AUE_TNRH	
536 AUE_TNRHTP	
537 AUE_TOKMAPPER	Table B-160

## Events in Audit Class ot

The following table lists in alphabetical order the `other` class of audit events provided in the Trusted Solaris 7 release.

**TABLE A-19** Other Audit Events (Default)

Audit Event Number and Event	Where Described
238 AUE_MEMCNTL	Table B-72

## Events in Audit Class pm

The following table lists in alphabetical order the `process modify` class of audit events provided in the Trusted Solaris 7 release.

**TABLE A-20** Process Modify Audit Events (Default)

Audit Event Number and Event		Where Described
24	AUE_CHROOT	Table B-31
1	AUE_EXIT	Table B-39
68	AUE_FCHDIR	Table B-41
69	AUE_FCHROOT	Table B-44
15	AUE_KILL	Table B-67
52	AUE_KILLPG	No longer supported.
203	AUE_NICE	Table B-90
204	AUE_OSETPGRP	No information.
200	AUE_OSETUID	No longer supported.
212	AUE_PRIOCNLSYS	Table B-106
214	AUE_SETEGID	Table B-135
215	AUE_SETEUID	Table B-136
526	AUE_SETPATTR	Table B-140
205	AUE_SETGID	Table B-135
26	AUE_SETGROUPS	Table B-139
27	AUE_SETPGRP	Table B-141
31	AUE_SETPRIORITY	No longer supported.

**TABLE A-20** Process Modify Audit Events (Default) *(continued)*

Audit Event Number and Event		Where Described
41	AUE_SETREGID	Table B-143
40	AUE_SETREUID	Table B-144
200	AUE_SETTUID -event name is AUE_OSETTUID	Table B-146
36	AUE_VTRACE	Table B-170

## Events in Audit Class ps

The following table lists in alphabetical order the `process start/stop` class of audit events provided in the Trusted Solaris 7 release.

**TABLE A-21** Process Start/Stop Audit Events (Default)

Audit Event Number and Event		Where Described
7	AUE_EXEC	Table B-38
23	AUE_EXECVE	
2	AUE_FORK	Table B-47
241	AUE_FORK1	
526	AUE_SETPATTR	Table B-140
25	AUE_VFORK	Table B-169
9027	AUE_psradm	Table B-266

## Events in Audit Class as

The following table lists in alphabetical order the system-wide administration class of audit events provided in the Trusted Solaris 7 release.

**TABLE A-22** System-wide Administration Audit Events (Default)

Audit Event Number and Event		Where Described
18	AUE_ACCT	Table B-6
50	AUE_ADJTIME	Table B-7
57	AUE_ASYNC_DAEMON	async_daemon(2) No information.
114	AUE_ASYNC_DAEMON_EXIT	async_daemon(2) No information.
538	AUE_CHSTATE	Table B-32
513	AUE_CLOCK_SETTIME	Table B-33
531	AUE_DRVPOLICY	Table B-36
246	AUE_MODADDMAN	Table B-77
245	AUE_MODCONFIG	Table B-78
243	AUE_MODLOAD	Table B-79
244	AUE_MODUNLOAD	Table B-80
533	AUE_PRIVENABLE	Table B-107
540	AUE_REMOUNT	Table B-163
59	AUE_SETDOMAINNAME	setdomainname(2) No information.

**TABLE A-22** System-wide Administration Audit Events (Default) *(continued)*

Audit Event Number and Event	Where Described
29 AUE_SETHOSTNAME	sethostname(2) No information.
51 AUE_SETRLIMIT	Table B-145
37 AUE_SETTIMEOFDAY	settimeofday(2) No information.
201 AUE_STIME	Table B-155
28 AUE_SWAPON	swapon(2) No information.
239 AUE_SYSINFO	Table B-157
9018 AUE_add_drv	Table B-216
9025 AUE_dispadmin	Table B-237
9032 AUE_eeprom	Table B-239
9042 AUE_installf	Table B-246
9020 AUE_modload	Table B-254
9021 AUE_modunload	Table B-254
9026 AUE_pbind	Table B-258
9040 AUE_pkginstall	Table B-260
9041 AUE_pkgremove	Table B-261
9027 AUE_psradm	Table B-266

**TABLE A-22** System-wide Administration Audit Events (Default) *(continued)*

Audit Event Number and Event		Where Described
9019	AUE_rem_drv	Table B-272
9043	AUE_removef	Table B-268
9022	AUE_setuname	Table B-278
9030	AUE_swap	Table B-282
9024	AUE_uname_set	Table B-286

## Events in Audit Class ss

The following table lists in alphabetical order the change system state class of audit events provided in the Trusted Solaris 7 release.

**TABLE A-23** Change System State Audit Events (Default)

Audit Event Number and Event		Where Described
539	AUE_FREEZE	Table B-161
561	AUE_REBOOT	Table B-162
560	AUE_SHUTDOWN	Table B-164
6160	AUE_halt_solaris	Table B-242
6161	AUE_reboot_solaris	Table B-267



**TABLE A-23** Change System State Audit Events (Default) *(continued)*

Audit Event Number and Event	Where Described
9028 AUE_run_level_change	Table B-273
9023 AUE_uadmin_cmd	Table B-283

## Events in Audit Class ax

The following table lists in alphabetical order the ax class of audit events provided in the Trusted Solaris 7 release.

**TABLE A-24** X Server Audit Events - Remainder (Default)

Audit Event Number and Event	Where Described
9039 AUE_sel_mgr_xfer	Table B-274

## Events in Audit Class xa

The following table lists in alphabetical order the xa class of audit events provided in the Trusted Solaris 7 release. This class contains X protocols that use "default" client privileges to succeed. These privileges are listed in the file `/usr/openwin/server/tsol/config.privs`. The security administrator can remove privileges from this file.

**TABLE A-25** X - Allowed Information Flows Audit Events (Default)

Audit Event Number and Event	Where Described
9194 AUE_ChangeHosts	Table B-212
9137 AUE_GrabServer	Table B-187

**TABLE A-25** X - Allowed Information Flows Audit Events (Default) *(continued)*

Audit Event Number and Event		Where Described
9183	AUE_InstallColormap	Table B-203
9146	AUE_SetFontPath	Table B-193
9138	AUE_UngrabServer	Table B-187

## Events in Audit Class xc

The following table lists in alphabetical order the `xc` class of audit events provided in the Trusted Solaris 7 release. This class contains audit events about the creation and destruction of X server objects.

**TABLE A-26** X - Object Create/Destroy Operations Audit Events (Default)

Audit Event Number and Event		Where Described
9176	AUE_AllocColor	Table B-202
9178	AUE_AllocColorCells	
9179	AUE_AllocColorPlanes	
9177	AUE_AllocNamedColor	
9120	AUE_ChangeProperty	Table B-179
9170	AUE_CreateColormap	Table B-201
9185	AUE_CreateCursor	Table B-205
9186	AUE_CreateGlyphCursor	Table B-206

**TABLE A-26** X - Object Create/Destroy Operations Audit Events (Default) *(continued)*

Audit Event Number and Event		Where Described
9103	AUE_CreateWindow	Table B-178
9121	AUE_DeleteProperty	Table B-179
9107	AUE_DestroySubwindows	Table B-178
9106	AUE_DestroyWindow	
9171	AUE_FreeColormap	Table B-203
9180	AUE_FreeColors	Table B-202
9187	AUE_FreeCursor	Table B-207
9152	AUE_FreeGC	Table B-196
9147	AUE_FreePixmap	Table B-208
9197	AUE_KillClient	Table B-212

## Events in Audit Class xl

The following table lists in alphabetical order the `xl` audit events provided in the Trusted Solaris 7 release.

**TABLE A-27** X - Client Login/Logout Audit Events (Default)

Audit Event Number and Event		Where Described
9101	AUE_ClientConnect	Table B-176
9102	AUE_ClientDisconnect	Table B-177

## Events in Audit Class xp

The following table lists in alphabetical order the `xp` audit events provided in the Trusted Solaris 7 release.

**TABLE A-28** X - Privileged Audit Events (Default)

Audit Event Number and Event		Where Described
9148	AUE_ChangeGc	Table B-194
9120	AUE_ChangeProperty	Table B-179
9108	AUE_ChangeSaveSet	Table B-178
9104	AUE_ChangeWindowAttributes	
9115	AUE_CirculateWindow	
9114	AUE_ConfigureWindow	
9172	AUE_CopyColormapAndFree	Table B-203
9149	AUE_CopyGC	Table B-195
9161	AUE_FillPolygon	Table B-199

**TABLE A-28** X - Privileged Audit Events (Default) *(continued)*

Audit Event Number and Event		Where Described
9199	AUE_ForceScreenSaver	Table B-210
9116	AUE_GetGeometry	Table B-178
9140	AUE_GetMotionEvents	Table B-189
9122	AUE_GetProperty	Table B-179
9105	AUE_GetWindowAttributes	Table B-178
9130	AUE_GrabButton	Table B-182
9135	AUE_GrabKey	Table B-185
9133	AUE_GrabKeyboard	Table B-186
9128	AUE_GrabPointer	Table B-183
9168	AUE_ImageText8	Table B-200
9169	AUE_ImageText16	
9173	AUE_InstallColormap	Table B-203
9123	AUE_ListProperties	Table B-179
9184	AUE_LookupColor	Table B-204
9111	AUE_MapSubwindows	Table B-178
9110	AUE_MapWindow	

**TABLE A-28** X - Privileged Audit Events (Default) *(continued)*

Audit Event Number and Event		Where Described
9160	AUE_PolyArc	Table B-199
9163	AUE_PolyFillArc	
9162	AUE_PolyFillRectangle	
9157	AUE_PolyLine	
9156	AUE_PolyPoint	
9158	AUE_PolySegment	Table B-200
9166	AUE_PolyText8	
9167	AUE_PolyText16	
9164	AUE_PutImage	
9183	AUE_QueryColors	
9145	AUE_QueryKeymap	Table B-192
9139	AUE_QueryPointer	Table B-188
9117	AUE_QueryTree	Table B-178
9188	AUE_RecolorCursor	Table B-207
9109	AUE_ReparentWindow	Table B-178
9198	AUE_RotateProperties	Table B-213
9195	AUE_SetAccessControl	Table B-212

**TABLE A-28** X - Privileged Audit Events (Default) *(continued)*

Audit Event Number and Event		Where Described
9151	AUE_SetClipRectangles	Table B-196
9150	AUE_SetDashes	
9193	AUE_SetScreenSaver	Table B-210
9124	AUE_SetSelectionOwner	Table B-181
9181	AUE_StoreColors	Table B-204
9182	AUE_StoreNamedColor	
9141	AUE_TranslateCoords	Table B-190
9136	AUE_UngrabKey	Table B-186
9174	AUE_UninstallColormap	Table B-203
9113	AUE_UnmapSubwindows	Table B-178
9112	AUE_UnmapWindow	
9202	AUE_XExtensions	Table B-215

## Events in Audit Class xs

The following table lists in alphabetical order the `xs` audit events provided in the Trusted Solaris 7 release.

---

**Note** - These events should be audited for success only, not for failure.

---

**TABLE A-29** X - Fail Silently Audit Events (Default)

Audit Event Number and Event		Where Described
9193	AUE_Bell	Table B-209
9132	AUE_ChangeActivePointerGrab	Table B-184
9190	AUE_ChangeKeyboardControl	Table B-209
9189	AUE_ChangeKeyboardMapping	
9192	AUE_ChangePointerControl	
9126	AUE_ConvertSelection	Table B-181
9154	AUE_CopyArea	Table B-198
9155	AUE_CopyPlane	
9119	AUE_GetAtomName	Table B-180
9165	AUE_GetImage	Table B-200
9144	AUE_GetInputFocus	Table B-191
9125	AUE_GetSelectionOwner	Table B-181
9128	AUE_GrabPointer	Table B-183
9118	AUE_InternAtom	Table B-180
9175	AUE_ListInstalledColormap	Table B-203
9159	AUE_PolyRectangle	Table B-199



**TABLE A-29** X - Fail Silently Audit Events (Default) *(continued)*

Audit Event Number and Event		Where Described
9127	AUE_SendEvent	Table B-189
9196	AUE_SetCloseDownMode	Table B-211
9143	AUE_SetInputFocus	Table B-191
9201	AUE_SetModifierMapping	Table B-214
9200	AUE_SetPointerMapping	
9124	AUE_SetSelectionOwner	Table B-181
9134	AUE_UngrabKeyboard	Table B-185
9129	AUE_UngrabPointer	Table B-183
9131	AUE_UngrabButton	
9142	AUE_WarpPointer	Table B-190



## Audit Record Descriptions

---

This appendix has two parts. The first part describes each part of an audit record structure and each audit token structure. The second part defines all of the audit records generated in Trusted Solaris 7 software by event description.

- “Audit Record Structure” on page 147
- “Audit Token Structure” on page 148
- “Kernel-Level Generated Audit Records” on page 176
- “Kernel-Level Pseudo-Events” on page 257
- “X Server Protocol Audit Records” on page 257
- “User-Level Generated Audit Records” on page 280

---

## Audit Record Structure

An audit record is a sequence of audit tokens. Each token contains event information such as user ID, time, and date. A header token begins an audit record, and an optional trailer concludes the record. Other audit tokens contain audit-relevant information. The following figure shows a typical audit record.

header token
subject token
slabel token
return token

Figure B-1 Typical Audit Record

## Audit Token Structure

Logically, each token has a token type identifier followed by data specific to the token. Each token type has its own format and structure. The audit tokens are shown in the table below. Those marked TS in the TS7 column are in Trusted Solaris 2.5.1 and Trusted Solaris 7 only. Those not marked TS are modified versions of audit tokens from the Solaris Basic Security Module. The token scheme can be extended.

TABLE B-1 Trusted Solaris Audit Tokens

Token Name	Description	TS7
acl	Access Control List information	TS
arbitrary	Data with format and type information	
arg	System call argument value	
attr	File attributes	
clearance	Clearance information	TS
exec_args	Exec system call arguments	
exec_env	Exec system call environment variables	
exit	Program exit information	
file	Audit file information	
groups	Process groups information (obsolete)	

**TABLE B-1** Trusted Solaris Audit Tokens *(continued)*

<b>Token Name</b>	<b>Description</b>	<b>TS7</b>
header	Indicates start of record	
host	Indicates the host where the audit record was collected	TS
ilabel	Information label information (obsolete in Trusted Solaris 7)	TS
in_addr	Internet address	
ip	IP header information	
ipc	System V IPC information	
ipc_perm	System V IPC object tokens	
ipport	Internet port address	
liaison	Liaison information for Trusted Networking	TS
newgroups	Process groups information	
opaque	Unstructured data (unspecified format)	
path	Path information (path)	
priv	Use of privilege information	TS
privilege	Privilege set information	TS
process	Process token information	
return	Status of system call	
seq	Sequence number token	
slabel	sensitivity label information	TS
socket	Socket type and addresses	
socket-inet	Socket port and address	

**TABLE B-1** Trusted Solaris Audit Tokens *(continued)*

Token Name	Description	TS7
<code>subject</code>	Subject information (same structure as process token)	
<code>text</code>	character string	
<code>trailer</code>	Indicates end of record	
<code>xatom</code>	X window atom identification	TS
<code>xclient</code>	X client identification	TS
<code>xcolormap</code>	X window color information	TS
<code>xcursor</code>	X window cursor information	TS
<code>xfont</code>	X window font information	TS
<code>xgc</code>	X window graphical context information	TS
<code>xpixmap</code>	Xwindow pixel mapping information	TS
<code>xproperty</code>	X window property information	TS
<code>xselect</code>	X window data information	TS
<code>xwindow</code>	X window window information	TS

An audit record always contains a `header` token and may contain a `trailer` token. The `header` token indicates where the audit record begins in the audit trail. The optional `trailer` token allows backward seeks of the audit trail. Every audit record contains a `subject` token, except for audit records from some non-attributable events. In the case of attributable events, these two tokens refer to the values of the process that caused the event. In the case of asynchronous events, the `process` tokens refer to the system. For an example of how to read an audit record, go to “Reading an Audit Record” on page 81.

## acl Token

The `acl` token provides information about any access control lists in place on an object. If there is no current `acl`, this token is not written to the audit record. Also, unless required by an audit record format, this token is normally recorded only when the appropriate auditing policy is set. The fields are:

- A token ID
- The object type of an array element
- The user/group id of an array element
- The permissions given to the subject

The following figure shows the token format.

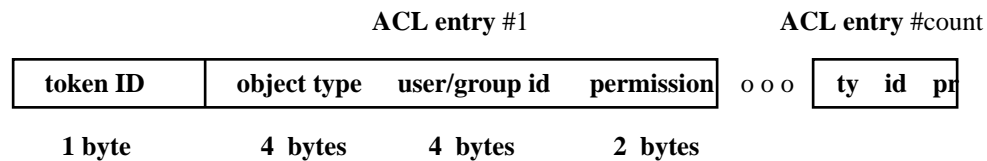


Figure B-2 `acl` Token Format

A list of `acl` tokens is displayed by `praudit(1M)` as follows:

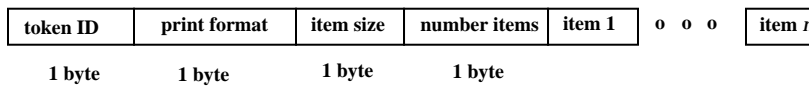
```
acl,user_obj,,rwx
acl,user,bin,—
acl,group_obj,,r-x
acl,class_obj,,r-
acl,other_obj,,r-x
```

## arbitrary Token

The `arbitrary` token encapsulates data for the audit trail. It consists of four fixed fields and an array of data. The item array may have a number of items. The fields are:

- A token ID
- A suggested format, such as decimal
- A size of encapsulated data, such as int
- A count of the data array items
- An item array

The following figure shows the token format.



*Figure B-3* arbitrary Token Format

The print format field can take the values shown in Table B-2.

**TABLE B-2** arbitrary Token Print Format Field Values

Value	Action
AUP_BINARY	Print date in binary
AUP_OCTAL	Print date in octal
AUP_DECIMAL	Print date in decimal
AUP_HEX	Print date in hex
AUP_STRING	Print date as a string

The item size field can take the values shown in Table B-3.

**TABLE B-3** arbitrary Token Item Size Field Values

Value	Action
AUR_BYTE	Data is in units of bytes (1 byte)
AUR_SHORT	Data is in units of shorts (2 bytes)
AUR_LONG	Data is in units of longs (4 bytes)
AUR_LONGLONG	Data is in units of longlongs (8 bytes)

An arbitrary token is displayed by `praudit` as follows:

```
arbitrary,decimal,int,1
42
```



## arg Token

The `arg` token contains system call argument information. A 32-bit integer system call argument is allowed in an audit record. The fields are:

- A token ID
- An argument ID of the relevant system call argument
- The argument value
- The length of an optional descriptive text string (does not show)
- An optional text string

The following figure shows the token format.

token ID	argument #	argument value	text length	text
1 byte	1 byte	4 bytes	2 bytes	<i>n</i> bytes

*Figure B-4* arg Token Format

An `arg` token is displayed by `praudit` as follows:

```
argument,2,0x3,cmd
```

## attr Token

The `attr` token contains file attribute information from the kernel's internal representation of a file or folder. This token usually accompanies a `path` token and is produced during path searches. In the event of a path-search error, this token is not included as part of the audit record since the file attribute information is not available. The fields are:

- A token ID
- The file access mode and type
- The owner user ID
- The owner group ID
- The file system ID
- The inode ID
- The device ID that the file might represent

See the `statvfs(2)` man page for further information about the file system ID and the device ID. The following figure shows the token format.

token ID	file mod	owner UID	owner GID	file system I	file inode ID	device ID
1 byte	4 bytes	4 bytes	4 bytes	4 bytes	4 bytes	4 bytes

*Figure B-5 attr Token Format*

An `attr` token is displayed by `praudit` as follows:

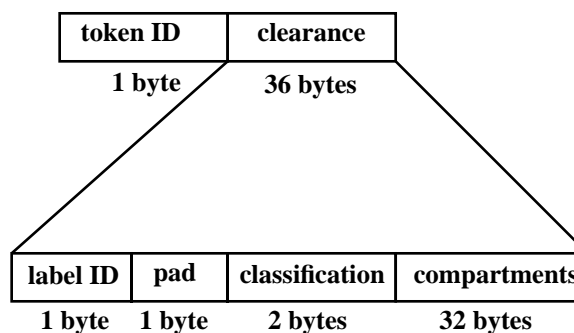
```
attribute,100555,root,root,1805,13871,-4288
```

## clearance Token

The `clearance` token contains Trusted Solaris clearance information. The fields are:

- A token ID
- The CMW clearance, containing
  - A pad ID identifying the label type
  - The clearance's classifications
  - The clearance's compartments

The following figure shows the token format.



*Figure B-6 clearance Token Format*

A `clearance` token is displayed by `praudit` as follows:

```
clearance, TOP SECRET
```

## exec\_args Token

The `exec_args` token records the arguments to an `exec( )` system call. The fields are:

- A token ID
- A count that represents the number of arguments passed to the `exec` call
- Zero or more null-terminated strings, the arguments of the `exec` call

The following figure shows an `exec_args` token.

token ID	count	exec_args
1 byte	4 bytes	count null-terminated strings

Figure B-7 `exec_args` Token Format

---

**Note** - The `exec_args` token is output only when the audit policy `argv` is active. See “Dynamic Procedures” on page 70 for more information.

---

An `exec_args` token is displayed by `praudit` as follows:

`exec_args,`

## exec\_env Token

The `exec_env` token records the current environment variables to an `exec( )` system call. The fields are:

- A token ID
- A count of the current environment variables in the `exec` call
- Zero or more null-terminated strings, the variables of the `exec` call

The following figure shows an `exec_env` token.

token ID	count	env_vars
1 byte	4 bytes	count null-terminated strings

Figure B-8 `exec_env` Token Format

---

**Note** - The `exec_env` token is output only when the audit policy `argv` is active. See “Dynamic Procedures” on page 70 for more information.

---

An `exec_env` token is displayed by `praudit` as follows:

## exec\_env exit Token

The `exit` token records the exit status of a program and a return value. The fields are:

- A token ID
- A program exit status as passed to the `exit()` system call
- A return value that describes the exit status or indicates a system error number

The following figure shows an `exit` token.

token ID	status	return value
1 byte	4 bytes	4 bytes

Figure B-9 exit Token Format

An `exit` token is displayed by `praudit` as follows:

```
exit>Error 0,0
```

## file Token

The `file` token is a special token generated by the audit daemon to mark the beginning of a new audit trail file and the end of an old file as it is deactivated. The audit daemon builds a special audit record containing this token to link together successive audit files into one audit trail. The fields are:

- A token ID
- A time and date stamp that identifies the time the file was created or closed
- A byte count of the file name including a null terminator (does not show)
- The file null-terminated name

The following figure shows the token format.

token ID	date & time	name length	previous/next file name
1 byte	8 bytes	2 bytes	<i>n</i> bytes

Figure B-10 file Token Format

A `file` token is displayed by `praudit` as follows:

## groups Token (Obsolete)

This token has been replaced by the `newgroups` token, which provides the same type of information but requires less space. A description of the `groups` token is provided here for completeness, but the application designer should use the `newgroups` token. Note that `praudit` does not distinguish between the two tokens as both token IDs are labelled `groups` when character output is displayed.

The `groups` token records the `groups` entries from the process's credential. The fields are:

- A token ID
- An array of `groups` entries of size `NGROUPS_MAX` (16)

The following figure shows a `groups` token.

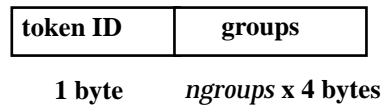


Figure B-11 `groups` Token Format

A `groups` token is displayed by `praudit` as follows:

```
group,staff,wheel,daemon,kmem,bin,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1
```

---

**Note** - The `groups` token is output only when the audit policy `group` is active. See “The `auditconfig` Command” on page 34 for more information.

---

## header Token

The `header` token is special in that it marks the beginning of an audit record and combines with the `trailer` token to bracket all the other tokens in the record. The fields are:

- A token ID
- The record length in bytes, including the `header` and `trailer` tokens
- An audit record structure version number

- An event ID identifying the type of audit event
- An event ID modifier with descriptive information about the event type
- The time and date the record was created

The following figure shows a `header` token.

token ID	byte count	version #	event ID	ID modifier	date and time
1 byte	4 bytes	1 byte	2 bytes	2 bytes	8 bytes

Figure B-12 header Token Format

The event modifier field has the following flags defined:

Value	Constant Name	Description
0x0001	PAD_MACUSE	MAC decision was successful
0x0002	PAD_MACREAD	MAC read failure
0x0004	PAD_MACWRITE	MAC write failure
0x0008	PAD_MACSEARCH	MAC search failure
0x0010	PAD_MACKILL	MAC signal failure
0x0020	PAD_MACTRACE	MAC trace failure
0x0040	PAD_MACIOCTL	MAC ioctl failure
0x0080	PAD_SPRIVUSE	Successful use of privilege
0x0100	PAD_FPRIVUSE	Failed use of privilege
0x4000	PAD_NONATTR	Nonattributable event
0x8000	PAD_FAILURE	Failed audit event

A `header` token is displayed by `praudit` as follows:

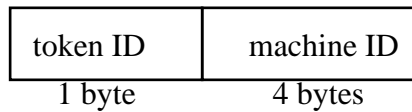
```
header,449,3,pfsh(1M),,Mon May
```

## host Token

The `host` token contains the machine ID for the workstation which generated this audit record. The fields are:

- A token ID
- The workstation ID of the host that generated the audit record

The following figure shows the token format.



*Figure B-13* host Token Format

A host token is displayed by `praudit` as follows:

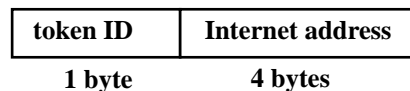
```
host,patchwork
```

## in\_addr Token

The `in_addr` token contains an Internet address. This 4-byte value is an Internet Protocol address. The fields are:

- A token ID
- An Internet address

The following figure shows the token format.



*Figure B-14* in\_addr Token Format

An `in_addr` token is displayed by `praudit` as follows:

```
ip addr,129.150.110.3
```

## ip Token

The `ip` token contains a copy of an Internet Protocol header but does not include any IP options. The IP options may be added by including more of the IP header in the token. The IP header structure is defined in `/usr/include/netinet/ip.h`. The fields are:

- A token ID
- A 20-byte copy of an IP header (all 20 bytes)

The following figure shows the token format.

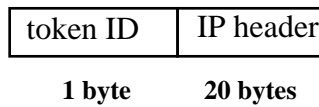


Figure B-15 ip Token Format

An `ip` token is displayed by `praudit` as follows:

```
ip,0.0.0.0
```

## ipc Token

The `ipc` token contains the System V IPC message/semaphore/shared-memory handle used by the caller to identify a particular IPC object. The fields are:

- A token ID
- An IPC object type identifier
- The IPC object handle

The following figure shows the token format.

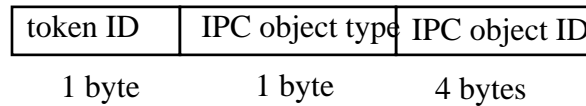


Figure B-16 ipc Token Format

An `ipc` token is displayed by `praudit` as follows:

```
IPC,msg,3
```

---

**Note** - The IPC object identifiers violate the context-free nature of the Solaris CMW audit tokens. No global “name” uniquely identifies IPC objects; instead, they are identified by their handles, which are valid only during the time the IPC objects are active. The identification should not be a problem since the System V IPC mechanisms are seldom used and they all share the same audit class.

---

The IPC object type field may have the values shown in Table B-4. The values are defined in `</usr/include/bsm/audit.h>`.



**TABLE B-4** IPC Object Type Field

Name	Value	Description
AU_IPC_MSG	1	IPC message object
AU_IPC_SEM	2	IPC semaphore object
AU_IPC_SHM	3	IPC shared memory object

## ipc\_perm Token

The `ipc_perm` token contains a copy of the System V IPC access information. Audit records for shared memory, semaphore, and message IPCs have this token added. The fields are:

- A token ID
- The IPC owner's user ID
- The IPC owner's group ID
- The IPC creator's user ID
- The IPC creator's group ID
- The IPC access modes
- The IPC sequence number
- The IPC key value

The values are taken from the `ipc_perm` structure associated with the IPC object. The following figure shows the token format.

token ID	owner uid	owner gid	creator uid	creator gid	ipc mode	sequence ID	IPC key
1 byte	4 bytes	4 bytes	4 bytes	4 bytes	4 bytes	4 bytes	4 bytes

*Figure B-17* ipc\_perm Token Format

An `ipc_perm` token is displayed by `praudit` as follows:

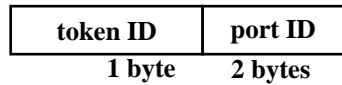
```
IPC perm,root,wheel,root,wheel,0,0,0x00000000
```

## ipport Token

The `ipport` token contains the TCP (or UDP) port address. The fields are:

- A token ID
- A TCP/UDP address

The following figure shows the token format.



*Figure B-18* ipport Token Format

An `ipport` token is displayed by `praudit` as follows:

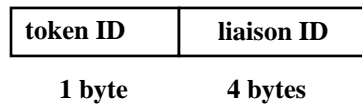
```
ipport,0xf6d6
```

## liaison Token

The `liaison` token contains a liaison ID used by the Trusted Networking software. The fields are:

- A token ID
- The liaison ID

The following figure shows the token format.



*Figure B-19* liaison Token Format

A `liaisontoken` is displayed by `praudit` as follows:

```
liaison,17
```

## newgroups Token

This token is the replacement for the `groups` token. Note that `praudit` does not distinguish between the two tokens as both token IDs are labelled `groups` when character output is displayed.

The `newgroups` token records the groups entries from the process's credential. The fields are:

- A token ID field
- A count of the number of groups contained in this audit record.
- Zero or more group entries.

The following figure shows the token format.

token ID	count	groups
1 byte	2 bytes	count * 4 bytes

Figure B-20 newgroups Token Format

---

**Note** - The `newgroups` token is output only when the audit policy `group` is active. See “The `auditconfig` Command” on page 34 for more information.

---

A `newgroups` token is displayed by `praudit` as follows:

```
newgroups,1,analysts
```

## opaque Token

The `opaque` token contains unformatted data as a sequence of bytes. The fields are:

- A token ID
- A byte count of the data array
- An array of byte data

The following figure shows the token format.

token ID	data length	data bytes
1 byte	2 bytes	<i>n</i> bytes

Figure B-21 opaque Token Format

An `opaque` token is displayed by `praudit` as follows:

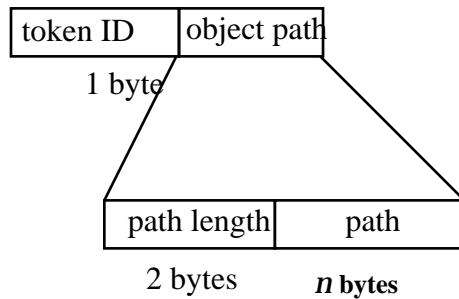
```
opaque,12,0x4f5041515545204441544100
```

## path Token

The `path` token contains access path information for an object. The fields are:

- A token ID
- A byte count of the path length (does not show)
- An absolute path to the object based on the real root of the system

The following figure shows the token format.



*Figure B-22* path Token Format

A path token is displayed by `praudit` as follows:

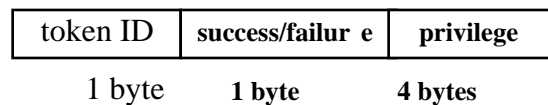
```
path,/etc/security/audit/patchwork
```

## priv Token

The `priv` token contains use of privilege information. The fields are:

- A token ID
- A success/failure field indicating whether the use of privilege was successful (1 success, 0 failure)
- The privilege being tested

The following figure shows a `priv` token.



*Figure B-23* priv Token Format

A `priv` token is displayed by `praudit` as follows:

`useofpriv,failed use of priv,win_mac_write`

## privilege Token

The `privilege` token contains privilege information for an object or a subject. The fields are:

- A token ID
- The type of privilege
- The privilege set

where `type` is one of the following:

Value	Type
0	Unknown or Undefined
1	Forced
2	Allowed
3	Effective
4	Inheritable
5	Permitted
6	Saved

The following figure shows the token format.

token ID	type	privileges
1 byte	1 byte	16 bytes

*Figure B-24* privilege Token Format

A `privilege` token is displayed by `praudit` as follows:

`privilege,1,proc_tcb_audit`

## process Token

The `process` token contains information describing a process as an object such as the recipient of a signal. The fields are:

- A token ID
- The user audit ID
- The effective user ID
- The effective group ID
- The real user ID
- The real group ID
- The process ID
- The session ID
- A terminal ID made up of
  - A device ID
  - A workstation ID

The following figure shows the token format.

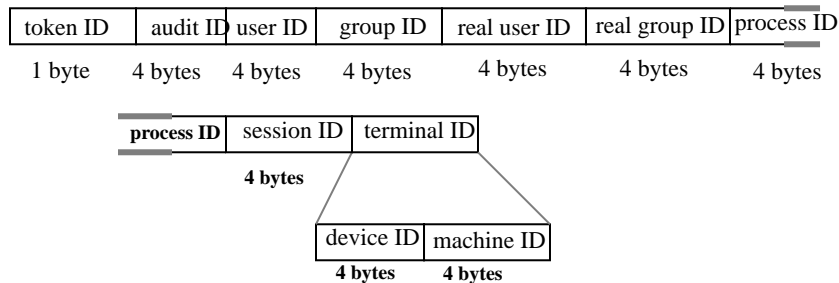


Figure B-25 Format for process and subject Tokens

The audit ID, user ID, group ID, process ID, and session ID are long instead of short.

---

**Note** - The `process` token fields for the session ID, the real user ID, or the real group ID may be unavailable. The entry is then set to -1.

---

A `process` token is displayed by `praudit` as follows:

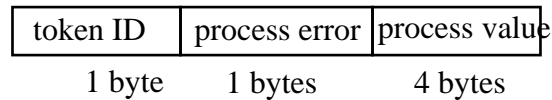
```
process,root,root,wheel,root,wheel,0,0,0,0,0,0,0
```

## return Token

The return token contains the return status of the system call (`u_error`) and the process return value (`u_rval1`). The token indicates exit status and other return values in application auditing. This token is always returned as part of kernel-generated audit records for system calls. The fields are:

- A token ID
- The system call error status
- The system call return value

The following figure shows the token format.



*Figure B-26* return Token Format

A return token is displayed by `praudit` as follows:

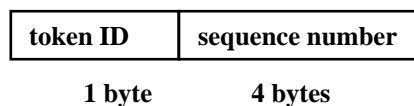
```
return,failure: No such file or directory,-1
```

## seq Token

The seq token (sequence token) is an optional token that contains an increasing sequence number. This token is for debugging. The token is added to each audit record when the `AUDIT_SEQ` policy is active. The fields are:

- A token ID
- A 32-bit unsigned long-sequence number

The sequence number is incremented every time an audit record is generated and put onto the audit trail. The following figure shows the token format.



*Figure B-27* seq Token Format

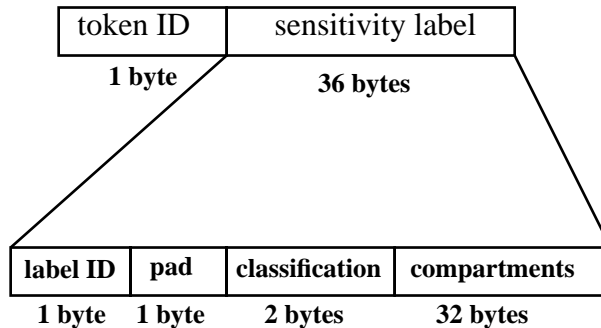
A seq token is displayed by `praudit` as follows:

## slabel Token

The `slabel` token contains a sensitivity label. The fields are:

- A token ID
- A sensitivity label

The following figure shows the token format.



*Figure B-28* slabel Token Format

An `slabel` token is displayed by `praudit` as follows:

```
slabel,ADMIN_LOW
```

## socket Token

The `socket` token contains information describing an Internet socket. The fields are:

- A token ID
- A socket type field (TCP/UDP/UNIX)
- The local port address
- The local Internet address
- The remote port address
- The remote Internet address

The socket type is taken from the designated socket and the port and Internet addresses are taken from the socket's `inpcb` control structure. The following figure shows the token format.



Token ID	socket type	local port	local Internet address	remote port	remote Internet address
1 byte	2 bytes	2 bytes	4 bytes	2 bytes	4 bytes

**Figure B-29** socket Token Format

A socket token is displayed by `praudit` as follows:

```
socket,0x0000,0x0000,0.0.0.0,0x0000,0.0.0.0
```

```
socket,0x0002,0x8008,patchwork
```

## socket-inet Token

The `socket-inet` token describes a socket connection to a local port, which is used to represent the socket information in the Internet namespace. The fields are:

- A token ID
- A socket family field that indicates the Internet family (`AF_INET`, `AF_OSI`, and so on)
- The local port address
- The socket address

The following figure shows the token format.

Token ID	socket family	local port	socket address
1 byte	2 bytes	2 bytes	4 bytes

**Figure B-30** socket-inet Token Format

A `socket-inet` token is displayed by `praudit` as follows:

```
socket,0x0002,0x8008,patchwork
```

## subject Token

The `subject` token describes a subject (process). The structure is the same as the `process` token:

- A token ID
- The user audit ID
- The effective user ID
- The effective group ID

- The real user ID
- The real group ID
- The process ID
- The session ID
- A terminal ID made up of
  - A device ID
  - A workstation ID

This token is always returned as part of kernel-generated audit records for system calls. The audit ID, user ID, group ID, process ID, and session ID are long instead of short. Figure B-25 shows the token format.

---

**Note** - The `subject` token fields for the session ID, the real user ID, or the real group ID may be unavailable. The entry is then set to `-1`.

---

A `subject` token is displayed by `praudit` as follows:

```
subject,root,root,staff,root,staff,552,552,24 3 patchwork
```

## text Token

The `text` token contains a text string. The fields are:

- A token ID
- The length of the text string (does not show)
- A text string

The following figure shows the token format.

token ID	text length	text string
1 bytes	2 bytes	<i>n</i> bytes

*Figure B-31* text Token Format

A `text` token is displayed by `praudit` as follows:

```
text,emily
```

## trailer Token

A `trailer` token marks the end of an audit record to support backward seeks of the audit trail. It is an optional token that is added as the last token of each record only when the `AUDIT_TRAIL` audit policy has been set. The fields are:

- A token ID
- A pad number that marks the end of the record (does not show)
- The total number of audit record characters including the header and trailer tokens

The following figure shows the token format.

token ID	pad number	byte count
1 byte	2 bytes	4 bytes

*Figure B-32* trailer Token Format

A `trailer` token is displayed by `praudit` as follows:

```
trailer,136
```

## xatom Token

The `xatom` token contains information concerning an X atom. The fields are:

- A token ID
- The string length
- A text string identifying the atom

The following figure shows the token format.

token ID	string length	atom string
1 byte	2 bytes	N bytes

*Figure B-33* xatom Token Format

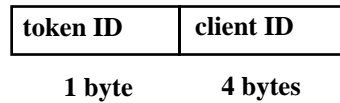
An `xatom` token is displayed by `praudit` as follows:

## xclient Token

The `xclient` token contains information concerning the X client. The fields are:

- A token ID
- The client ID

The following figure shows the token format.



*Figure B-34* xclient Token Format

An `xclient` token is displayed by `praudit` as follows:

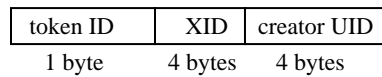
```
xclient,15
```

## xcolormap Token

The `xcolormap` token contains information about the colormaps. The fields are:

- A token ID
- The X server identifier
- The creator's user ID

The following figure shows the token format.



*Figure B-35* Format for `xcolormap`, `xcursor`, `xfont`, `xgc`, `xpixmap`, and `xwindow` Tokens

An `xcolormap` token is displayed by `praudit` as follows:

`xcolormap,0x08c00005,svr`

## xcursor Token

The `xcursor` token contains information about the cursors. The fields are:

- A token ID
- The X server identifier
- The creator's user ID

Figure B–35 shows the token format.

An `xcursor` token is displayed by `praudit` as follows:

`xcursor,0x0f400006,svr`

## xfont Token

The `xfont` token contains information about the fonts. The fields are:

- A token ID
- The X server identifier
- The creator's user ID

Figure B–35 shows the token format.

An `xfont` token is displayed by `praudit` as follows:

`xfont,0x08c00001,svr`

## xgc Token

The `xgc` token contains information about the `xgc`. The fields are:

- A token ID
- The X server identifier
- The creator's user ID

Figure B–35 shows the token format.

An `xgc` token is displayed by `praudit` as follows:

xgc,0x002f2ca0,src

## xpixmap Token

The `xpixmap` token contains information about the pixel mappings. The fields are:

- A token ID
- The X server identifier
- The creator's user ID

Figure B-35 shows the token format.

An `xpixmap` token is displayed by `praudit` as follows:

xpixmap,0x08c00005,src

## xproperty Token

The `xproperty` token contains information about various properties of a window. The fields are:

- A token ID
- The X server identifier
- The creator's user ID
- A string length
- A string (atom name)

The following figure shows an `xproperty` token format.

token ID	XID	creator UID	strlen	string (atom name)
1 byte	4 bytes	4 bytes	2 bytes	N bytes

Figure B-36 xproperty Token Format

An `xproperty` token is displayed by `praudit` as follows:

xproperty,0x000075d5,root,\_MOTIF\_DEFAULT\_BINDINGS

## xselect Token

The `xselect` token contains the data moved between windows. This data is a byte stream with no assumed internal structure, and a property string. The fields are:

- A token ID

- The length of the property string
- The property string
- A length for the property type
- The property type string
- A length field that gives the number of bytes of data
- A byte string containing the data

The following figure shows the token format.

token ID	property length	property string	prop type len	prop type	data length	window data
1 byte	2 bytes	N bytes	2 bytes	N bytes	2 bytes	N bytes

*Figure B-37 xselect Token Format*

An `xselect` token is displayed by `praudit` as follows:

```
xselect,
```

## xwindow Token

The `xwindow` token contains information about a window. The fields are:

- A token ID
- The X server identifier
- The creator's user ID

Figure B-35 shows the token format.

An `xwindow` token is displayed by `praudit` as follows:

```
xwindow,0x07400001,gww
```

---

# Audit Records

## General Audit Record Structure

The audit records produced by Trusted Solaris 7 have a sequence of tokens. Certain tokens are optional within an audit record, according to the current audit policy. The

group, sequence, and trailer tokens fall into this category. The administrator can determine if these are included in an audit record with the `auditconfig` command `-getpolicy` option.

## Kernel-Level Generated Audit Records

These audit records are created by system calls which are used by the kernel. The records are sorted alphabetically by system call. The description of each record includes:

- The name of the system call
- A man page reference (if appropriate)
- The audit event number
- The audit event name
- The audit event class
- The mask for the event class
- The audit record structure

**TABLE B-5** `access(2)`

Event Name	Event ID	Event Class	Mask
AUE_ACCESS	14	fa	0x00000004

Format:  
*header-token*  
*path-token* [*attr-token*]  
[*slabel-token*] (object)  
[*priv-token*] (if privilege used or required)  
*subject-token*  
*slabel-token* (subject)  
*return-token*



TABLE B-6 acct(2)

Event Name	Event ID	Event Class	Mask
AUE_ACCT	18	as	0x00020000
Format (zero path): <i>header-token</i> <i>argument-token</i> (1, "accounting off", 0) <i>[priv-token]</i> (if privilege used or required) <i>subject-token</i> <i>return-token</i> Format (non-zero path): <i>header-token</i> <i>path-token</i> [ <i>attr-token</i> ] <i>subject-token</i> <i>return-token</i>			

TABLE B-7 adjtime(2)

Event Name	Event ID	Event Class	Mask
AUE_ADJTIME	50	as	0x00000800
Format: <i>header-token</i> <i>[priv-token]</i> (if privilege used or required) <i>subject-token</i> <i>return-token</i>			

TABLE B-8 audit(2)

Event Name	Event ID	Event Class	Mask
AUE_AUDIT	211	no	0x00000000
Format: <i>header-token</i> <i>[priv-token]</i> (if privilege used or required) <i>subject-token</i> <i>return-token</i>			

**TABLE B-9**    `auditon(2)` — get current active root

Event Name	Event ID	Event Class	Mask
AUE_AUDITON_GETCAR	224	aa	0x00040000
Format: <i>header-token</i> <i>[priv-token]</i> (if privilege used or required) <i>subject-token</i> <i>return-token</i>			

**TABLE B-10**    `auditon(2)` — get event class

Event Name	Event ID	Event Class	Mask
AUE_AUDITON_GETCLASS	231	aa	0x00040000
Format: <i>header-token</i> <i>[priv-token]</i> (if privilege used or required) <i>subject-token</i> <i>return-token</i>			

**TABLE B-11**    `auditon(2)` — get audit state

Event Name	Event ID	Event Class	Mask
AUE_AUDITON_GETCOND	229	aa	0x00040000
Format: <i>header-token</i> <i>[priv-token]</i> (if privilege used or required) <i>subject-token</i> <i>return-token</i>			

**TABLE B-12** auditon(2) — get current working directory

Event Name	Event ID	Event Class	Mask
AUE_AUDITON_GETCWD	223	aa	0x00040000
Format: <i>header-token</i> <i>[priv-token]</i> (if privilege used or required) <i>subject-token</i> <i>return-token</i>			

**TABLE B-13** auditon(2) — get kernel mask

Event Name	Event ID	Event Class	Mask
AUE_AUDITON_GETKMASK	221	aa	0x00040000
Format: <i>header-token</i> <i>[priv-token]</i> (if privilege used or required) <i>subject-token</i> <i>return-token</i>			

**TABLE B-14** auditon(2) — get audit statistics

Event Name	Event ID	Event Class	Mask
AUE_AUDITON_GETSTAT	225	aa	0x00040000
Format: <i>header-token</i> <i>[priv-token]</i> (if privilege used or required) <i>subject-token</i> <i>return-token</i>			

**TABLE B-15** `auditon(2)` — GETPOLICY command

Event Name	Event ID	Event Class	Mask
AUE_AUDITON_GPOLICY	114	aa	0x00040000
Format: <i>header-token</i> <i>[priv-token]</i> (if privilege used or required) <i>subject-token</i> <i>return-token</i>			

**TABLE B-16** `auditon(2)` — get audit queue control parameters

Event Name	Event ID	Event Class	Mask
AUE_AUDITON_GQCTRL	145	aa	0x00040000
Format: <i>header-token</i> <i>[priv-token]</i> (if privilege used or required) <i>subject-token</i> <i>return-token</i>			

**TABLE B-17** `auditon(2)` — set event class

Event Name	Event ID	Event Class	Mask
AUE_AUDITON_SETCLASS	232	aa	0x00040000
Format: <i>header-token</i> <i>[argument-token]</i> (2, "setclass:ec_event", event number) <i>[argument-token]</i> (3, "setclass:ec_class", class mask) <i>[priv-token]</i> (if privilege used or required) <i>subject-token</i> <i>return-token</i>			

**TABLE B-18**    auditon(2) — set audit state

Event Name	Event ID	Event Class	Mask
AUE_AUDITON_SETCOND	230	aa	0x00040000

Format:  
*header-token*  
*[argument-token]* (3, "setcond", audit state)  
*[priv-token]* (if privilege used or required)  
*subject-token*  
*return-token*

**TABLE B-19**    auditon(2) — set kernel mask

Event Name	Event ID	Event Class	Mask
AUE_AUDITON_SETKMASK	222	aa	0x00040000

Format:  
*header-token*  
*[argument-token]* (2, "setkmask:as\_success", kernel mask)  
*[argument-token]* (2, "setkmask:as\_failure", kernel mask)  
*[priv-token]* (if privilege used or required)  
*subject-token*  
*return-token*

**TABLE B-20**    auditon(2) — set mask per session ID

Event Name	Event ID	Event Class	Mask
AUE_AUDITON_SETSMASK	228	aa	0x00040000

Format:  
*header-token*  
*[argument-token]* (3, "setsmask:as\_success", session ID mask)  
*[argument-token]* (3, "setsmask:as\_failure", session ID mask)  
*[priv-token]* (if privilege used or required)  
*subject-token*  
*return-token*

**TABLE B-21** `auditon(2)` — reset audit statistics

Event Name	Event ID	Event Class	Mask
AUE_AUDITON_SETSTAT	226	aa	0x00040000
Format: <i>header-token</i> <i>[priv-token]</i> (if privilege used or required) <i>subject-token</i> <i>return-token</i>			

**TABLE B-22** `auditon(2)` — set mask per uid

Event Name	Event ID	Event Class	Mask
AUE_AUDITON_SETUMASK	227	aa	0x00040000
Format: <i>header-token</i> <i>[argument-token]</i> (3, "setumask:as_success", audit ID mask) <i>[argument-token]</i> (3, "setumask:as_failure", audit ID mask) <i>[priv-token]</i> (if privilege used or required) <i>subject-token</i> <i>return-token</i>			

**TABLE B-23** `auditon(2)` — SETPOLICY command

Event Name	Event ID	Event Class	Mask
AUE_AUDITON_SPOLICY	147	aa	0x00040000
Format: <i>header-token</i> <i>[argument-token]</i> (1, "policy", audit policy flags) <i>[priv-token]</i> (if privilege used or required) <i>subject-token</i> <i>return-token</i>			

**TABLE B-23** auditon(2) — SETPOLICY command *(continued)*

**TABLE B-24** auditon(2) — set audit queue control parameters

Event Name	Event ID	Event Class	Mask
AUE_AUDITON_SQCTRL	146	aa	0x00040000
Format: <i>header-token</i> <i>[argument-token]</i> (3,"setqctrl:aq_hiwater",queue control param.) <i>[argument-token]</i> (3,"setqctrl:aq_lowwater",queue control param.) <i>[argument-token]</i> (3,"setqctrl:aq_bufsz",queue control param.) <i>[argument-token]</i> (3,"setqctrl:aq_delay",queue control param.) <i>[priv-token]</i> (if privilege used or required) <i>subject-token</i> <i>return-token</i>			

**TABLE B-25** auditpsa(2)

Event Name	Event ID	Event Class	Mask
AUE_AUDITPSA	529	aa	0x00040000
Format (valid file descriptor): <i>header-token</i> <i>argument-token</i> (1,"op", state) <i>in_addr-token</i> <i>[priv-token]</i> (if privilege used or required) <i>subject-token</i> <i>slabel-token</i> <i>return-token</i>			

TABLE B-26 auditstat(2)

Event Name	Event ID	Event Class	Mask
AUE_AUDITSTAT	150	aa	0x00040000
Format: <i>header-token</i> <i>[argument-token]</i> <i>[priv-token]</i> (if privilege used or required) <i>subject-token</i> <i>return-token</i>			

TABLE B-27 auditsvc(2)

Event Name	Event ID	Event Class	Mask
AUE_AUDITSVC	136	aa	0x00040000
Format (valid file descriptor): <i>header-token</i> <i>[path-token]</i> <i>[attr-token]</i> <i>[priv-token]</i> (if privilege used or required) <i>subject-token</i> <i>return-token</i> Format (invalid file descriptor): <i>header-token</i> <i>argument-token</i> (1, "no path: fd", file descriptor) <i>[priv-token]</i> (if privilege used or required) <i>subject-token</i> <i>return-token</i>			



TABLE B-28 chdir(2)

Event Name	Event ID	Event Class	Mask
AUE_CHDIR	8	pc	0x00300000
Format: <i>header-token</i> <i>path-token</i> [ <i>attr-token</i> ] [ <i>slabel-token</i> ] (object) [ <i>priv-token</i> ] (if privilege used or required) <i>subject-token</i> <i>slabel-token</i> (subject) <i>return-token</i>			

TABLE B-29 chmod(2)

Event Name	Event ID	Event Class	Mask
AUE_CHMOD	10	fm	0x00000008
Format: <i>header-token</i> <i>argument-token</i> (2, "new file mode", mode) <i>path-token</i> [ <i>attr-token</i> ] [ <i>slabel-token</i> ] (object) [ <i>priv-token</i> ] (if privilege used or required) <i>subject-token</i> <i>slabel-token</i> (subject) <i>return-token</i>			

**TABLE B-30** chown(2)

Event Name	Event ID	Event Class	Mask
AUE_CHOWN	11	fm	0x00000008
Format: <i>header-token</i> <i>argument-token</i> (2, "new file uid", uid) <i>argument-token</i> (3, "new file gid", gid) <i>path-token</i> <i>[attr-token]</i> <i>[slabel-token]</i> (object) <i>[priv-token]</i> (if privilege used or required) <i>subject-token</i> <i>slabel-token</i> (subject) <i>return-token</i>			

**TABLE B-31** chroot(2)

Event Name	Event ID	Event Class	Mask
AUE_CHROOT	24	pm	0x00200000
Format: <i>header-token</i> <i>path-token</i> <i>[attr-token]</i> <i>[slabel-token]</i> (object) <i>[priv-token]</i> (if privilege used or required) <i>subject-token</i> <i>slabel-token</i> (subject) <i>return-token</i>			

**TABLE B-32** chstate(2)

Event Name	Event ID	Event Class	Mask
AUE_CHSTATE	538	as	0x00000800
Format: <i>header-token</i> <i>[priv-token]</i> (if privilege used or required) <i>subject-token</i> <i>slabel-token</i> <i>return-token</i>			

**TABLE B-33** clock\_settime(3R)

Event Name	Event ID	Event Class	Mask
AUE_CLOCK_SETTIME	513	as	0x00000800
Format: <i>header-token</i> <i>slabel-token</i> <i>return-token</i>			

**TABLE B-34** `close(2)`

Event Name	Event ID	Event Class	Mask
AUE_CLOSE	112	c1	0x00000040

Format:

<file system object>

*header-token*

*argument-token* (1, "fd", file descriptor)

[*path-token*]

[*attr-token*]

[*slabel-token*] (object)

[*priv-token*] (if privilege used or required)

*subject-token*

*slabel-token* (subject)

*return-token*

Also for files closed on process termination. The *argument-token* is only present with the `close()` system call. It may be removed in future releases. The *path-token* is present only with valid file descriptors.

**TABLE B-35** `creat(2)`

Event Name	Event ID	Event Class	Mask
AUE_CREAT	4	fc	0x00000010

Format

*header-token*

*path-token*

[*attr-token*]

[*slabel-token*] (object)

[*priv-token*] (if privilege used or required)

*subject-token*

*slabel-token* (subject)

*return-token*

**TABLE B-36** devpolicy(2)

Event Name	Event ID	Event Class	Mask
AUE_DRVPOLICY	531	as	0x00000800
Format: <i>header-token</i> <i>[priv-token]</i> (if privilege used or required) <i>subject-token</i> <i>slabel-token</i> <i>return-token</i>			

**TABLE B-37** enter prom, exit prom

Event Name	Event ID	Event Class	Mask
AUE_ENTERPROM	153	na	0x00000400
AUE_EXITPROM	154	na	0x00000400
Format: <i>header-token</i> <i>text-token</i> (addr, "monitor PROM"   "kadb") <i>[priv-token]</i> (if privilege used or required) <i>subject-token</i> <i>return-token</i>			

**TABLE B-38** `exec(2)`, `execve(2)`

Event Name	Event ID	Event Class	Mask
AUE_EXEC	7	ps	0x00100000
AUE_EXECVE	23	ps	0x00100000

Format:  
*header-token*  
*path-token*  
*[attr-token]*  
*[slabel-token]* (object)  
*[priv-token]* (if privilege used or required)  
*subject-token*  
*slabel-token* (subject)  
*return-token*

**TABLE B-39** `exit(2)`

Event Name	Event ID	Event Class	Mask
AUE_EXIT	1	pm	0x00200000

Format:  
*header-token*  
*[priv-token]* (if privilege used or required)  
*subject-token*  
*return-token*

TABLE B-40 fauditpsa(2)

Event Name	Event ID	Event Class	Mask
AUE_FAUDITPSA	530	aa	0x00040000
Format: <i>header-token</i> <i>[priv-token]</i> (if privilege used or required) <i>subject-token</i> <i>slabel-token</i> <i>return-token</i>			

TABLE B-41 fchdir(2)

Event Name	Event ID	Event Class	Mask
AUE_FCHDIR	68	pc	0x00300000
Format: <i>header-token</i> <i>[path-token]</i> <i>[attr-token]</i> <i>[slabel-token]</i> (object) <i>[priv-token]</i> (if privilege used or required) <i>subject-token</i> <i>slabel-token</i> (subject) <i>return-token</i>			

**TABLE B-42** fchmod(2)

Event Name	Event ID	Event Class	Mask
AUE_FCHMOD	39	fm	0x00000008
<p>Format (valid file descriptor):</p> <p><i>header-token</i></p> <p><i>argument-token</i> (2, "new file mode", mode)</p> <p>[<i>path-token</i>]</p> <p>[<i>attr-token</i>]</p> <p>[<i>slabel-token</i>] (object)</p> <p>[<i>priv-token</i>] (if privilege used or required)</p> <p><i>subject-token</i></p> <p><i>slabel-token</i> (subject)</p> <p><i>return-token</i></p> <p>Format (invalid file descriptor):</p> <p><i>header-token</i></p> <p><i>argument-token</i> (2, "new file mode", mode)</p> <p><i>argument-token</i> (1, "no path: fd", file descriptor)</p> <p>[<i>priv-token</i>] (if privilege used or required)</p> <p><i>subject-token</i></p> <p><i>slabel-token</i> (subject)</p> <p><i>return-token</i></p>			



TABLE B–43 fchown(2)

Event Name	Event ID	Event Class	Mask
AUE_FCHOWN	38	fm	0x00000008
Format (valid file descriptor): <i>header-token</i> <i>argument-token</i> (2, "new file uid", uid) <i>argument-token</i> (3, "new file gid", gid) <i>[path-token]</i> <i>[attr-token]</i> <i>[slabel-token]</i> (object) <i>[priv-token]</i> (if privilege used or required) <i>subject-token</i> <i>slabel-token</i> (subject) <i>return-token</i> Format (non-file descriptor): <i>header-token</i> <i>argument-token</i> (2, "new file uid", uid) <i>argument-token</i> (3, "new file gid", gid) <i>argument-token</i> (1, "no path: fd", file descriptor) <i>[priv-token]</i> (if privilege used or required) <i>subject-token</i> <i>slabel-token</i> (subject) <i>return-token</i>			

TABLE B–44 fchroot(2)

Event Name	Event ID	Event Class	Mask
AUE_FCHROOT	69	pm	0x00200000
Format: <i>header-token</i> <i>[path-token]</i> <i>[attr-token]</i> <i>[slabel-token]</i> (object) <i>[priv-token]</i> (if privilege used or required) <i>subject-token</i> <i>slabel-token</i> (subject) <i>return-token</i>			

TABLE B-45 fcntl(2)

Event Name	Event ID	Event Class	Mask
AUE_FCNTL (cmd=F_GETLK, F_SETLK, F_SETLKW)	30	fn	0x40000000
Format (file descriptor): <i>header-token</i> <i>argument-token</i> (2, "cmd", cmd) <i>path-token</i> <i>attr-token</i> <i>[priv-token]</i> (if privilege used or required) <i>subject-token</i> <i>return-token</i> Format (bad file descriptor): <i>header-token</i> <i>argument-token</i> (2, "cmd", cmd) <i>argument-token</i> (1, "no path: fd", file descriptor) <i>[priv-token]</i> (if privilege used or required) <i>subject-token</i> <i>return-token</i>			

TABLE B-46 fgetslldname(2)

Event Name	Event ID	Event Class	Mask
AUE_FGETSLDNAME	532	fc	0x00000010
Format: <i>header-token</i> <i>[priv-token]</i> (if privilege used or required) <i>subject-token</i> <i>slabel-token</i> <i>return-token</i>			

**TABLE B–47**    fork(2), fork1(2)

Event Name	Event ID	Event Class	Mask
AUE_FORK	2	ps	0x00100000
AUE_FORK1	241	ps	0x00100000

Format:

*header-token*

*[argument-token]* (0, "child PID", pid)

*[priv-token]* (if privilege used or required)

*subject-token*

*slabel-token* (subject)

*return-token*

The `fork()` and `fork1()` return values are undefined since each audit record is produced at the point that the child process is spawned.

**TABLE B–48**    fsetcmwlabel(2)

Event Name	Event ID	Event Class	Mask
AUE_FSETCMWLABEL	544	fm	0x00000008

Format:

*header-token*

*argument-token* (3, "flag", which parts of label to set)

*[slabel-token]* (if slabel is being set)

*path-token*

*[attr-token]*

*[slabel-token]*

*[priv-token]* (if privilege used or required)

*subject-token*

*slabel-token* (subject)

*return-token*

**TABLE B-49** fsetfattrflag(2)

Event Name	Event ID	Event Class	Mask
AUE_FSETFATTRFLAG	523	fm	0x00000008
Format: <i>header-token</i> <i>argument-token</i> (2, "which", which flags to set) <i>argument-token</i> (3, "attrs", flag values) <i>path-token</i> <i>[attr-token]</i> <i>[slabel-token]</i> <i>[priv-token]</i> (if privilege used or required) <i>subject-token</i> <i>slabel-token</i> <i>return-token</i>			

**TABLE B-50** fstatfs(2)

Event Name	Event ID	Event Class	Mask
AUE_FSTATFS	55	fa	0x00000004
Format (file descriptor): <i>header-token</i> <i>[path-token]</i> <i>[attr-token]</i> <i>[slabel-token]</i> (object) <i>[priv-token]</i> (if privilege used or required) <i>subject-token</i> <i>slabel-token</i> (subject) <i>return-token</i> Format (non-file descriptor): <i>header-token</i> <i>argument-token</i> (1, "no path: fd", file descriptor) <i>[priv-token]</i> (if privilege used or required) <i>subject-token</i> <i>slabel-token</i> (subject) <i>return-token</i>			

TABLE B-51 getaudit(2)

Event Name	Event ID	Event Class	Mask
AUE_GETAUDIT	132	aa	0x00040000
Format: <i>header-token</i> <i>[priv-token]</i> (if privilege used or required) <i>subject-token</i> <i>return-token</i>			

TABLE B-52 getauid(2)

Event Name	Event ID	Event Class	Mask
AUE_GETAUID	130	aa	0x00040000
Format: <i>header-token</i> <i>[priv-token]</i> (if privilege used or required) <i>subject-token</i> <i>return-token</i>			

TABLE B-53 getcmwfsrange(2)

Event Name	Event ID	Event Class	Mask
AUE_GETCMWFSRANGE	545	fa	0x00000004
Format: <i>header-token</i> <i>path-token</i> <i>[attr-token]</i> <i>[slabel-token]</i> <i>[priv-token]</i> (if privilege used or required) <i>subject-token</i> <i>slabel-token</i> <i>return-token</i>			

**TABLE B-54** getcmwlabel(2), fgetcmwlabel(2), lgetcmwlabel(2)

Event Name	Event ID	Event Class	Mask
AUE_GETCMWLABEL	546	fa	0x00000004
AUE_FGETCMWLABEL	118	fa	0x00000004
AUE_LGETCMWLABEL	548	fa	0x00000004
Format: <i>header-token</i> <i>path-token</i> <i>[attr-token]</i> <i>[slabel-token]</i> <i>[priv-token]</i> (if privilege used or required) <i>subject-token</i> <i>slabel-token</i> <i>return-token</i>			

**TABLE B-55** getdents(2)

Event Name	Event ID	Event Class	Mask
AUE_GETDENTS	193	no	0x00000000
Format: <i>header-token</i> <i>path-token</i> <i>[attr-token]</i> <i>[priv-token]</i> (if privilege used or required) <i>subject-token</i> <i>return-token</i>			

TABLE B-56 getfpriv(2)

Event Name	Event ID	Event Class	Mask
AUE_GETFILEPRIV	547	fa	0x00000004
Format: <i>header-token</i> <i>path-token</i> <i>[attr-token]</i> <i>[slabel-token]</i> <i>[priv-token]</i> (if privilege used or required) <i>subject-token</i> <i>slabel-token</i> <i>return-token</i>			

TABLE B-57 getmldadorn(2)

Event Name	Event ID	Event Class	Mask
AUE_GETMLDADORN	554	fa	0x00000004
Format: <i>header-token</i> <i>path-token</i> <i>[attr-token]</i> <i>[slabel-token]</i> <i>[priv-token]</i> (if privilege used or required) <i>subject-token</i> <i>slabel-token</i> <i>return-token</i>			

**TABLE B-58** getmsg(2)

Event Name	Event ID	Event Class	Mask
AUE_GETMSG	217	nt	0x00000100
Format: <i>header-token</i> <i>argument-token</i> (1, "fd", file descriptor) <i>argument-token</i> (4, "pri", priority) <i>[priv-token]</i> (if privilege used or required) <i>subject-token</i> <i>return-token</i>			

**TABLE B-59** getmsg(2) — accept, receive

Event Name	Event ID	Event Class	Mask
AUE_SOCKETACCEPT	247	nt	0x00000100
AUE_SOCKETRECEIVE	250	nt	0x00000100
Format: <i>header-token</i> <i>socket-inet-token</i> <i>argument-token</i> (1, "fd", file descriptor) <i>argument-token</i> (4, "pri", priority) <i>[priv-token]</i> (if privilege used or required) <i>subject-token</i> <i>return-token</i>			



TABLE B-60 getmsgqcmwlabel(2)

Event Name	Event ID	Event Class	Mask
AUE_GETMSGQCMWLABEL	514	ip	0x00000200

Format:  
*header-token*  
*argument-token* (1, "msg ID", message ID)  
*[argument-token]*  
*[ipc\_perm-token]* (of the IPC)  
*[slabel-token]*  
*[priv-token]* (if privilege used or required)  
*subject-token*  
*slabel-token*  
*return-token*

The *ipc*, *ipc\_perm*, and the *slabel* of the ipc tokens are not included if the msg ID is invalid.

TABLE B-61 getpmsg(2)

Event Name	Event ID	Event Class	Mask
AUE_GETPMSG	219	nt	0x00000100

Format:  
*header-token*  
*argument-token* (1, "fd", file descriptor)  
*[priv-token]* (if privilege used or required)  
*subject-token*  
*return-token*

TABLE B-62 getportaudit(2)

Event Name	Event ID	Event Class	Mask
AUE_GETPORTAUDIT	149	aa	0x00040000
Format: <i>header-token</i> <i>[priv-token]</i> (if privilege used or required) <i>subject-token</i> <i>return-token</i>			

TABLE B-63 getsemcmwlabel(2)

Event Name	Event ID	Event Class	Mask
AUE_GETSEMCMWLABEL	515	ip	0x00000200
Format: <i>header-token</i> <i>argument-token</i> (1, "sem ID", semaphore ID) <i>[argument-token]</i> <i>[ipc_perm-token]</i> (of the IPC) <i>[slabel-token]</i> <i>[priv-token]</i> (if privilege used or required) <i>subject-token</i> <i>slabel-token</i> <i>return-token</i>			
The <i>ipc</i> , <i>ipc_perm</i> , and the <i>slabel</i> of the ipc tokens are not included if the sem ID is invalid.			

TABLE B-64 getshmcmwlabel(2)

Event Name	Event ID	Event Class	Mask
AUE_GETSHMCMWLABEL	516	ip	0x00000200

Format:  
*header-token*  
*argument-token* (1, "shm ID", semaphore ID)  
*[argument-token]*  
*[ipc\_perm-token]* (of the IPC)  
*[slabel-token]*  
*[priv-token]* (if privilege used or required)  
*subject-token*  
*slabel-token*  
*return-token*

The *ipc*, *ipc\_perm*, and the *slabel* of the ipc tokens are not included if the shm ID is invalid.

TABLE B-65 getslldname(2)

Event Name	Event ID	Event Class	Mask
AUE_GETSLDNAME	555	fa	0x00000004

Format:  
*header-token*  
*path-token*  
*[attr-token]*  
*[slabel-token]*  
*[priv-token]* (if privilege used or required)  
*subject-token*  
*slabel-token*  
*return-token*

TABLE B-66    `ioctl(2)`

Event Name	Event ID	Event Class	Mask
AUE_IOCTL	158	io	0x20000000
Format (good file descriptor): <i>header-token</i> <i>path-token</i> <i>[attr-token]</i> <i>argument-token</i> (2, "cmd" <code>ioctl</code> cmd) <i>argument-token</i> (3, "arg" <code>ioctl</code> arg) <i>[priv-token]</i> (if privilege used or required) <i>subject-token</i> <i>return-token</i> Format (socket): <i>header-token</i> <i>[socket-token]</i> <i>argument-token</i> (2, "cmd" <code>ioctl</code> cmd) <i>argument-token</i> (3, "arg" <code>ioctl</code> arg) <i>[priv-token]</i> (if privilege used or required) <i>subject-token</i> <i>return-token</i> Format (non-file file descriptor): <i>header-token</i> <i>argument-token</i> (1, "fd", file descriptor) <i>argument-token</i> (2, "cmd" <code>ioctl</code> cmd) <i>argument-token</i> (3, "arg" <code>ioctl</code> arg) <i>[priv-token]</i> (if privilege used or required) <i>subject-token</i> <i>return-token</i> Format (bad file name): <i>header-token</i> <i>argument-token</i> (1, "no path: fd", file descriptor) <i>argument-token</i> (2, "cmd" <code>ioctl</code> cmd) <i>argument-token</i> (3, "arg" <code>ioctl</code> arg) <i>[priv-token]</i> (if privilege used or required) <i>subject-token</i> <i>return-token</i>			

TABLE B-67 kill(2)

Event Name	Event ID	Event Class	Mask
AUE_KILL	15	pm	0x00200000
Format (valid process): <i>header-token</i> <i>argument-token</i> (2, "signal", signo) <i>[process-token]</i> <i>[slabel-token]</i> (process) <i>[priv-token]</i> (if privilege used or required) <i>subject-token</i> <i>slabel-token</i> (subject) <i>return-token</i>			
Format (zero or negative process): <i>header-token</i> <i>argument-token</i> (2, "signal", signo) <i>argument-token</i> (1, "process", pid) <i>[priv-token]</i> (if privilege used or required) <i>subject-token</i> <i>slabel-token</i> (subject) <i>return-token</i>			

TABLE B-68 lchown(2)

Event Name	Event ID	Event Class	Mask
AUE_LCHOWN	237	fm	0x00000008
Format: <i>header-token</i> <i>argument-token</i> (2, "new file uid", uid) <i>argument-token</i> (3, "new file gid", gid) <i>path-token</i> <i>[attr-token]</i> <i>[priv-token]</i> (if privilege used or required) <i>subject-token</i> <i>return-token</i>			

TABLE B-69 link(2)

Event Name	Event ID	Event Class	Mask
AUE_LINK	5	fc	0x00000010
Format: <i>header-token</i> <i>path-token</i> (from path) <i>[attr-token]</i> (from path) <i>[slabel-token]</i> (from path) <i>path-token</i> (to path) <i>[attr-token]</i> (to path) <i>[slabel-token]</i> (to path) <i>[priv-token]</i> (if privilege used or required) <i>subject-token</i> <i>slabel-token</i> (subject) <i>return-token</i>			

TABLE B-70 lstat(2)

Event Name	Event ID	Event Class	Mask
AUE_LSTAT	17	fa	0x00000004
Format: <i>header-token</i> <i>path-token</i> <i>[attr-token]</i> <i>[priv-token]</i> (if privilege used or required) <i>subject-token</i> <i>return-token</i>			

TABLE B-71 lxstat(2)

Event Name	Event ID	Event Class	Mask
AUE_LXSTAT	236	fa	0x00000004
Format: <i>header-token</i> <i>path-token</i> <i>[attr-token]</i> <i>[slabel-token]</i> (object) <i>[priv-token]</i> (if privilege used or required) <i>subject-token</i> <i>slabel-token</i> (subject) <i>return-token</i>			

TABLE B-72 memcntl(2)

Event Name	Event ID	Event Class	Mask
AUE_MEMCNTL	238	ot	0x80000000
Format: <i>header-token</i> <i>argument-token</i> (1, "base", base address) <i>argument-token</i> (2, "len", length) <i>argument-token</i> (3, "cmd", command) <i>argument-token</i> (4, "arg", command args) <i>argument-token</i> (5, "attr", command attributes) <i>argument-token</i> (6, "mask", 0) <i>[priv-token]</i> (if privilege used or required) <i>subject-token</i> <i>return-token</i>			

**TABLE B-73**    `mkdir(2)`

Event Name	Event ID	Event Class	Mask
AUE_MKDIR	47	fc	0x00000010
Format: <i>header-token</i> <i>argument-token</i> (2, "mode", mode) <i>path-token</i> <i>[attr-token]</i> <i>[slabel-token]</i> (object) <i>[priv-token]</i> (if privilege used or required) <i>subject-token</i> <i>slabel-token</i> (subject) <i>return-token</i>			

**TABLE B-74**    `mknod(2)`

Event Name	Event ID	Event Class	Mask
AUE_MKNOD	9	fc	0x00000010
Format: <i>header-token</i> <i>argument-token</i> (2, "mode", mode) <i>argument-token</i> (3, "dev", dev) <i>path-token</i> <i>[attr-token]</i> <i>[slabel-token]</i> (object) <i>[priv-token]</i> (if privilege used or required) <i>subject-token</i> <i>slabel-token</i> (subject) <i>return-token</i>			



TABLE B-75 mldsetfattrflag(2)

Event Name	Event ID	Event Class	Mask
AUE_MLDSETFATTRFLAG	524	fm	0x00000008
Format: <i>header-token</i> <i>argument-token</i> (2, "which", which flags to set) <i>argument-token</i> (3, "attrs", flag values) <i>[priv-token]</i> (if privilege used or required) <i>subject-token</i> <i>slabel-token</i> <i>return-token</i>			

TABLE B-76 mmap(2)

Event Name	Event ID	Event Class	Mask
AUE_MMMap	210	no	0x00000000
Format (valid file descriptor): <i>header-token</i> <i>argument-token</i> (1, "addr", segment address) <i>argument-token</i> (2, "len", segment length) <i>[path-token]</i> <i>[attr-token]</i> <i>[priv-token]</i> (if privilege used or required) <i>subject-token</i> <i>return-token</i> Format (invalid file descriptor): <i>header-token</i> <i>argument-token</i> (1, "addr", segment address) <i>argument-token</i> (2, "len", segment length) <i>argument-token</i> (1, "no path: fd", file descriptor) <i>[priv-token]</i> (if privilege used or required) <i>subject-token</i> <i>return-token</i>			

**TABLE B-77** modctl(2) — bind module

Event Name	Event ID	Event Class	Mask
AUE_MODADDMAJ	246	as	0x00000800
Format: <i>header-token</i> <i>[text-token]</i> (driver major number) <i>[text-token]</i> (driver name) <i>text-token</i> (root dir.   "no rootdir") <i>text-token</i> (driver major number   "no drvname") <i>argument-token</i> (5, "", number of aliases) (0..n) <i>[text-token]</i> (aliases) <i>[priv-token]</i> (if privilege used or required) <i>subject-token</i> <i>return-token</i>			

**TABLE B-78** modctl(2) — configure module

Event Name	Event ID	Event Class	Mask
AUE_MODCONFIG	245	as	0x00000800
Format: <i>header-token</i> <i>text-token</i> (root dir.   "no rootdir") <i>text-token</i> (driver major number   "no drvname") <i>[priv-token]</i> (if privilege used or required) <i>subject-token</i> <i>return-token</i>			

**TABLE B-79** modctl(2) — load module

Event Name	Event ID	Event Class	Mask
AUE_MODLOAD	243	as	0x00020000
Format: <i>header-token</i> <i>[text-token]</i> (default path) <i>text-token</i> (filename path) <i>[priv-token]</i> (if privilege used or required) <i>subject-token</i> <i>return-token</i>			

**TABLE B-80** modctl(2) — unload module

Event Name	Event ID	Event Class	Mask
AUE_MODUNLOAD	244	as	0x00020000
Format: <i>header-token</i> <i>argument-token</i> (1, "id", module ID) <i>[priv-token]</i> (if privilege used or required) <i>subject-token</i> <i>return-token</i>			

TABLE B-81 mount(2)

Event Name	Event ID	Event Class	Mask
AUE_MOUNT	62	ao	0x00080000
Format (UNIX file system): <i>header-token</i> <i>argument-token</i> (3, "flags", flags) <i>text-token</i> (filesystem type) <i>path-token</i> [attr-token] [slabel-token] (object) [priv-token] (if privilege used or required) <i>subject-token</i> <i>slabel-token</i> (subject) <i>return-token</i> Format (NFS file system): <i>header-token</i> <i>argument-token</i> (3, "flags", flags) <i>text-token</i> (filesystem type) <i>text-token</i> (host name) <i>argument-token</i> (3, "internal flags", flags) <i>path-token</i> [attr-token] [slabel-token] (object) [priv-token] (if privilege used or required) <i>subject-token</i> <i>slabel-token</i> (subject) <i>return-token</i>			

TABLE B-82 msgctl(2) — IPC\_RMID command

Event Name	Event ID	Event Class	Mask
AUE_MSGCTL_RMID	85	ip	0x00000200
Format: <i>header-token</i> <i>argument-token</i> (1, "msg ID", message ID) [argument-token] [ipc_perm-token] [slabel-token] [priv-token] (if privilege used or required) <i>subject-token</i> <i>slabel-token</i> <i>return-token</i> The <i>ipc</i> , <i>ipc_perm</i> , and the <i>slabel</i> of the ipc tokens are not included if the msg ID is invalid.			

**TABLE B–82** msgctl(2) — IPC\_RMID command *(continued)*

**TABLE B–83** msgctl(2) — IPC\_SET command

Event Name	Event ID	Event Class	Mask
AUE_MSGCTL_SET	86	ip	0x00000200

Format:

*header-token*

*argument-token* (1, "msg ID", message ID)

[*argument-token*]

[*ipc\_perm-token*] (of the IPC's old values)

[*slabel-token*]

[*ipc\_perm-token*] (of the IPC's new values)

[*slabel-token*]

[*priv-token*] (if privilege used or required)

*subject-token*

*slabel-token*

*subject-token*

*return-token*

The *ipc*, *ipc\_perm*, and the *slabel* of the ipc tokens are not included if the msg ID is invalid.

**TABLE B–84** msgctl(2) — IPC\_STAT command

Event Name	Event ID	Event Class	Mask
AUE_MSGCTL_STAT	87	ip	0x00000200

Format:

*header-token*

*argument-token* (1, "msg ID", message ID)

[*argument-token*]

[*ipc\_perm-token*] (of the IPC)

[*slabel-token*]

[*priv-token*] (if privilege used or required)

*subject-token*

*slabel-token*

*return-token*

The *ipc*, *ipc\_perm*, and the *slabel* of the ipc tokens are not included if the msg ID is invalid.

**TABLE B-85** msgget(2)

Event Name	Event ID	Event Class	Mask
AUE_MSGGET	88	ip	0x00000200

Format:

*header-token*  
*argument-token* (1, "msg key", message key)  
*argument-token* (2, "msg flag", message flags)  
*[ipc\_perm-token]* (of the IPC object)  
*[slabel-token]*  
*[argument-token]*  
*[priv-token]* (if privilege used or required)  
*subject-token*  
*slabel-token*  
*return-token*

The *ipc*, *ipc\_perm*, and the *slabel* of the ipc tokens are not included if the msg ID is invalid.

**TABLE B-86** msggetl(2)

Event Name	Event ID	Event Class	Mask
AUE_MSGGETL	174	ip	0x00000200

Format:

*header-token*  
*argument-token* (1, "msg key", message key)  
*argument-token* (2, "msg flag", message flags)  
*slabel-token* (desired SL)  
*[ipc\_perm-token]* (of the IPC object)  
*[slabel-token]*  
*[argument-token]*  
*[priv-token]* (if privilege used or required)  
*subject-token*  
*slabel-token*  
*return-token*

The *ipc*, *ipc\_perm*, and the *slabel* of the ipc tokens are not included if the msg ID is invalid.

TABLE B-87 msgrcv(2)

Event Name	Event ID	Event Class	Mask
AUE_MSGRCV	89	ip	0x00000200
AUE_MSGRCVL	175	ip	0x00000200

Format:

*header-token*

*argument-token* (1, "msg ID", message ID)

[*argument-token*]

[*ipc\_perm-token*]

[*slabel-token*]

[*priv-token*] (if privilege used or required)

*subject-token*

*slabel-token*

*return-token*

The *ipc*, *ipc\_perm*, and the *slabel* of the ipc tokens are not included if the msg ID is invalid.

TABLE B-88 msgsnd(2)

Event Name	Event ID	Event Class	Mask
AUE_MSGSND	90	ip	0x00000200

Format:

*header-token*

*argument-token* (1, "msg ID", message ID)

[*argument-token*]

[*ipc\_perm-token*] (of the IPC's new values)

[*slabel-token*]

[*priv-token*] (if privilege used or required)

*subject-token*

*slabel-token*

*return-token*

The *ipc*, *ipc\_perm*, and the *slabel* of the ipc tokens are not included if the msg ID is invalid.

**TABLE B-89**    `munmap(2)`

Event Name	Event ID	Event Class	Mask
AUE_MUNMAP	214	cl	0x00000040
Format: <i>header-token</i> <i>argument-token</i> (1, "addr", address of memory) <i>argument-token</i> (2, "len", memory segment size) <i>[priv-token]</i> (if privilege used or required) <i>subject-token</i> <i>return-token</i>			

**TABLE B-90**    `old nice(2)`

Event Name	Event ID	Event Class	Mask
AUE_NICE	203	pc	0x00300000
Format: <i>header-token</i> <i>[priv-token]</i> (if privilege used or required) <i>subject-token</i> <i>slabel-token</i> (subject) <i>return-token</i>			

**TABLE B-91**    `open(2) — read`

Event Name	Event ID	Event Class	Mask
AUE_OPEN_R	72	fr	0x00000001
Format: <i>header-token</i> <i>path-token</i> <i>[attr-token]</i> <i>[slabel-token]</i> (object) <i>[priv-token]</i> (if privilege used or required) <i>subject-token</i> <i>slabel-token</i> (subject) <i>return-token</i>			



TABLE B-91 open(2) — read (continued)

TABLE B-92 open(2) — read,creat

Event Name	Event ID	Event Class	Mask
AUE_OPEN_RC	73	fc,fr	0x00000011
Format: <i>header-token</i> <i>path-token</i> [attr-token] [slabel-token] (object) [priv-token] (if privilege used or required) <i>subject-token</i> slabel-token (subject) return-token			

TABLE B-93 open(2) — read,trunc,creat

Event Name	Event ID	Event Class	Mask
AUE_OPEN_RTC	75	fc,fd,fr	0x00000031
Format: <i>header-token</i> <i>path-token</i> [attr-token] [slabel-token] (object) [priv-token] (if privilege used or required) <i>subject-token</i> slabel-token (subject) return-token			

**TABLE B-94** open(2) — read,trunc

Event Name	Event ID	Event Class	Mask
AUE_OPEN_RT	74	fd, fr	0x00000021
Format: <i>header-token</i> <i>path-token</i> <i>[attr-token]</i> <i>[slabel-token]</i> (object) <i>[priv-token]</i> (if privilege used or required) <i>subject-token</i> <i>slabel-token</i> (subject) <i>return-token</i>			

**TABLE B-95** open(2) — read,write

Event Name	Event ID	Event Class	Mask
AUE_OPEN_RW	80	fr, fw	0x00000003
Format: <i>header-token</i> <i>path-token</i> <i>[attr-token]</i> <i>[slabel-token]</i> (object) <i>[priv-token]</i> (if privilege used or required) <i>subject-token</i> <i>slabel-token</i> (subject) <i>return-token</i>			

TABLE B-96 open(2) — read,write,creat

Event Name	Event ID	Event Class	Mask
AUE_OPEN_RWC	81	<i>fr</i> , <i>fw</i> , <i>fc</i>	0x00000013
Format: <i>header-token</i> <i>path-token</i> [ <i>attr-token</i> ] [ <i>slabel-token</i> ] (object) [ <i>priv-token</i> ] (if privilege used or required) <i>subject-token</i> <i>slabel-token</i> (subject) <i>return-token</i>			

TABLE B-97 open(2) — read,write,trunc,creat

Event Name	Event ID	Event Class	Mask
AUE_OPEN_RWTC	83	<i>fr</i> , <i>fw</i> , <i>fc</i> , <i>fd</i>	0x00000033
Format: <i>header-token</i> <i>path-token</i> [ <i>attr-token</i> ] [ <i>slabel-token</i> ] (object) [ <i>priv-token</i> ] (if privilege used or required) <i>subject-token</i> <i>slabel-token</i> (subject) <i>return-token</i>			

**TABLE B-98** open(2) — read,write,trunc

Event Name	Event ID	Event Class	Mask
AUE_OPEN_RWT	82	fr, fw, fd	0x00000023
Format: <i>header-token</i> <i>path-token</i> [ <i>attr-token</i> ] [ <i>slabel-token</i> ] (object) [ <i>priv-token</i> ] (if privilege used or required) <i>subject-token</i> <i>slabel-token</i> (subject) <i>return-token</i>			

**TABLE B-99** open(2) — write

Event Name	Event ID	Event Class	Mask
AUE_OPEN_W	76	fw	0x00000002
Format: <i>header-token</i> <i>path-token</i> [ <i>attr-token</i> ] [ <i>slabel-token</i> ] (object) [ <i>priv-token</i> ] (if privilege used or required) <i>subject-token</i> <i>slabel-token</i> (subject) <i>return-token</i>			

**TABLE B-100**    `open(2)` — write,creat

Event Name	Event ID	Event Class	Mask
AUE_OPEN_WC	77	<code>fw,fc</code>	0x00000012
Format: <i>header-token</i> <i>path-token</i> [ <i>attr-token</i> ] [ <i>slabel-token</i> ] (object) [ <i>priv-token</i> ] (if privilege used or required) <i>subject-token</i> <i>slabel-token</i> (subject) <i>return-token</i>			

**TABLE B-101**    `open(2)` — write,trunc,creat

Event Name	Event ID	Event Class	Mask
AUE_OPEN_WTC	79	<code>fw,fc,fd</code>	0x00000032
Format: <i>header-token</i> <i>path-token</i> [ <i>attr-token</i> ] [ <i>slabel-token</i> ] (object) [ <i>priv-token</i> ] (if privilege used or required) <i>subject-token</i> <i>slabel-token</i> (subject) <i>return-token</i>			

**TABLE B-102** open(2) — write,trunc

Event Name	Event ID	Event Class	Mask
AUE_OPEN_WT	78	fw,fd	0x00000022
Format: <i>header-token</i> <i>path-token</i> <i>[attr-token]</i> <i>[slabel-token]</i> (object) <i>[priv-token]</i> (if privilege used or required) <i>subject-token</i> <i>slabel-token</i> (subject) <i>return-token</i>			

**TABLE B-103** pathconf(2)

Event Name	Event ID	Event Class	Mask
AUE_PATHCONF	71	fa	0x00000004
Format: <i>header-token</i> <i>path-token</i> <i>[attr-token]</i> <i>[slabel-token]</i> (object) <i>[priv-token]</i> (if privilege used or required) <i>subject-token</i> <i>slabel-token</i> (subject) <i>return-token</i>			

TABLE B-104 pipe(2)

Event Name	Event ID	Event Class	Mask
AUE_PIPE	185	no	0x00000000
Format: <i>header-token</i> <i>[priv-token]</i> (if privilege used or required) <i>subject-token</i> <i>slabel-token</i> (subject) <i>return-token</i>			

TABLE B-105 preadl(2)

Event Name	Event ID	Event Class	Mask
AUE_PREADL	527	no	0x00000000
Format: <i>header-token</i> <i>path-token</i> <i>[attr-token]</i> <i>[slabel-token]</i> <i>[priv-token]</i> (if privilege used or required) <i>subject-token</i> <i>slabel-token</i> <i>return-token</i>			

TABLE B-106 priocntl(2)

Event Name	Event ID	Event Class	Mask
AUE_PRIOCNTLSYS	212	pm	0x00200000
Format: <i>header-token</i> <i>argument-token</i> (1, "pc_version", priocntl version num.) <i>argument-token</i> (3,"cmd", command) <i>[priv-token]</i> (if privilege used or required) <i>subject-token</i> <i>slabel-token</i> (subject) <i>return-token</i>			

TABLE B-107 privilege enable

Event Name	Event ID	Event Class	Mask
AUE_PRIVENABLE	533	as	0x00020000
Format: <i>header-token</i> <i>[priv-token]</i> (if privilege used or required) <i>subject-token</i> <i>slabel-token</i> <i>return-token</i>			

TABLE B-108 process dumped core

Event Name	Event ID	Event Class	Mask
AUE_CORE	111	fc	0x0000010
Format: <i>header-token</i> <i>path-token</i> <i>[attr-token]</i> <i>argument-token</i> (1, "signal", signal) <i>[priv-token]</i> (if privilege used or required) <i>subject-token</i> <i>return-token</i>			



TABLE B-108 process dumped core (continued)

TABLE B-109 putmsg(2)

Event Name	Event ID	Event Class	Mask
AUE_PUTMSG	216	nt	0x00000100
Format: <i>header-token</i> <i>argument-token</i> (1, "fd", file descriptor) <i>argument-token</i> (4, "pri", priority) <i>[priv-token]</i> (if privilege used or required) <i>subject-token</i> <i>return-token</i>			

TABLE B-110 putmsg(2) - connect, send

Event Name	Event ID	Event Class	Mask
AUE_SOCKETCONNECT	248	nt	0x00000100
AUE_SOCKETSEND	249	nt	0x00000100
Format: <i>header-token</i> <i>socket-inet-token</i> <i>argument-token</i> (1, "fd", file descriptor) <i>argument-token</i> (4, "pri", priority) <i>[priv-token]</i> (if privilege used or required) <i>subject-token</i> <i>return-token</i>			

**TABLE B-111** putpmsg(2)

Event Name	Event ID	Event Class	Mask
AUE_PUTPMSG	218	nt	0x00000100
Format: <i>header-token</i> <i>argument-token</i> (1, "fd", file descriptor) <i>[priv-token]</i> (if privilege used or required) <i>subject-token</i> <i>return-token</i>			

**TABLE B-112** read(2), readl(2), readvl(2)

Event Name	Event ID	Event Class	Mask
AUE_READ	192	no	0x00000000
AUE_READL	558		
AUE_READVL	559		
Format: <i>header-token</i> <i>path-token</i> <i>[attr-token]</i> <i>[slabel-token]</i> <i>[priv-token]</i> (if privilege used or required) <i>subject-token</i> <i>slabel-token</i> <i>return-token</i>			

TABLE B-113 readlink(2)

Event Name	Event ID	Event Class	Mask
AUE_READLINK	22	fr	0x00000001
Format: <i>header-token</i> <i>path-token</i> [ <i>attr-token</i> ] [ <i>slabel-token</i> ] (object) [ <i>priv-token</i> ] (if privilege used or required) <i>subject-token</i> <i>slabel-token</i> (subject) <i>return-token</i>			

TABLE B-114 rename(2)

Event Name	Event ID	Event Class	Mask
AUE_RENAME	42	fc, fd	0x00000030
Format: <i>header-token</i> <i>path-token</i> (from name) [ <i>attr-token</i> ] (from name) [ <i>slabel-token</i> ] (from name) [ <i>path-token</i> ] (to name) [ <i>priv-token</i> ] (if privilege used or required) <i>subject-token</i> <i>slabel-token</i> (subject) <i>return-token</i>			

TABLE B-115 `rmdir(2)`

Event Name	Event ID	Event Class	Mask
AUE_RMDIR	48	fd	0x00000020
Format: <i>header-token</i> <i>path-token</i> <i>[attr-token]</i> <i>[slabel-token]</i> (object) <i>[priv-token]</i> (if privilege used or required) <i>subject-token</i> <i>slabel-token</i> (subject) <i>return-token</i>			

TABLE B-116 `semctl(2)` — `getall`

Event Name	Event ID	Event Class	Mask
AUE_SEMCTL_GETALL	105	ip	0x00000200
Format: <i>header-token</i> <i>argument-token</i> (1, "sem ID", semaphore ID) <i>[argument-token]</i> <i>[ipc_perm-token]</i> <i>[slabel-token]</i> <i>[priv-token]</i> (if privilege used or required) <i>subject-token</i> <i>slabel-token</i> <i>return-token</i>  The <i>ipc</i> , <i>ipc_perm</i> , and the <i>slabel</i> of the ipc tokens are not included if the semaphore ID is invalid.			

**TABLE B-117** semctl(2) — GETNCNT command

Event Name	Event ID	Event Class	Mask
AUE_SEMCTL_GETNCNT	102	ip	0x00000200

Format:  
*header-token*  
*argument-token* (1, "sem ID", semaphore ID)  
*[argument-token]*  
*[ipc\_perm-token]*  
*[slabel-token]*  
*[priv-token]* (if privilege used or required)  
*subject-token*  
*slabel-token*  
*return-token*

The *ipc*, *ipc\_perm*, and the *slabel* of the ipc tokens are not included if the semaphore ID is invalid.

**TABLE B-118** semctl(2) — GETPID command

Event Name	Event ID	Event Class	Mask
AUE_SEMCTL_GETPID	103	ip	0x00000200

Format:  
*header-token*  
*argument-token* (1, "sem ID", semaphore ID)  
*[argument-token]*  
*[ipc\_perm-token]*  
*[slabel-token]*  
*[priv-token]* (if privilege used or required)  
*subject-token*  
*slabel-token*  
*return-token*

The *ipc*, *ipc\_perm*, and the *slabel* of the ipc tokens are not included if the semaphore ID is invalid.

**TABLE B-119** semctl(2) — GETVAL command

Event Name	Event ID	Event Class	Mask
AUE_SEMCTL_GETVAL	104	ip	0x00000200

Format:

*header-token*  
*argument-token* (1, "sem ID", semaphore ID)  
*[argument-token]*  
*[ipc\_perm-token]*  
*[slabel-token]*  
*[priv-token]* (if privilege used or required)  
*subject-token*  
*slabel-token*  
*return-token*

The *ipc*, *ipc\_perm*, and the *slabel* of the ipc tokens are not included if the semaphore ID is invalid.

**TABLE B-120** semctl(2) — GETZCNT command

Event Name	Event ID	Event Class	Mask
AUE_SEMCTL_GETZCNT	106	ip	0x00000200

Format:

*header-token*  
*argument-token* (1, "sem ID", semaphore ID)  
*[argument-token]*  
*[ipc\_perm-token]*  
*[slabel-token]*  
*[priv-token]* (if privilege used or required)  
*subject-token*  
*slabel-token*  
*return-token*

The *ipc*, *ipc\_perm*, and the *slabel* of the ipc tokens are not included if the semaphore ID is invalid.

TABLE B-121 `semctl(2)` — IPC\_RMID command

Event Name	Event ID	Event Class	Mask
AUE_SEMCTL_RMID	99	ip	0x00000200

Format:

*header-token*

*argument-token* (1, "sem ID", semaphore ID)

[*argument-token*]

[*ipc\_perm-token*]

[*slabel-token*]

[*priv-token*] (if privilege used or required)

*subject-token*

*slabel-token*

*return-token*

The *ipc*, *ipc\_perm*, and the *slabel* of the ipc tokens are not included if the semaphore ID is invalid.

TABLE B-122 `semctl(2)` — IPC\_SET command

Event Name	Event ID	Event Class	Mask
AUE_SEMCTL_SET	100	ip	0x00000200

Format:

*header-token*

*argument-token* (1, "sem ID", semaphore ID)

[*argument-token*]

[*ipc\_perm-token*] (of the IPC's old values)

[*slabel-token*]

[*ipc\_perm-token*] (of the IPC's new values)

[*slabel-token*]

[*priv-token*] (if privilege used or required)

*subject-token*

*slabel-token*

*return-token*

The *ipc*, *ipc\_perm*, and the *slabel* of the ipc tokens are not included if the semaphore ID is invalid.

**TABLE B-123** semctl(2) — SETALL command

Event Name	Event ID	Event Class	Mask
AUE_SEMCTL_SETALL	108	ip	0x00000200

Format:

*header-token*  
*argument-token* (1, "sem ID", semaphore ID)  
*[argument-token]*  
*[ipc\_perm-token]*  
*[slabel-token]*  
*[priv-token]* (if privilege used or required)  
*subject-token*  
*slabel-token*  
*return-token*

The *ipc*, *ipc\_perm*, and the *slabel* of the ipc tokens are not included if the semaphore ID is invalid.

**TABLE B-124** semctl(2) — SETVAL command

Event Name	Event ID	Event Class	Mask
AUE_SEMCTL_SETVAL	107	ip	0x00000200

Format:

*header-token*  
*argument-token* (1, "sem ID", semaphore ID)  
*[argument-token]*  
*[ipc\_perm-token]*  
*[slabel-token]*  
*[priv-token]* (if privilege used or required)  
*subject-token*  
*slabel-token*  
*return-token*

The *ipc*, *ipc\_perm*, and the *slabel* of the ipc tokens are not included if the semaphore ID is invalid.



TABLE B-125 semctl(2) — IPC\_STAT command

Event Name	Event ID	Event Class	Mask
AUE_SEMCTL_STAT	101	ip	0x00000200
Format: <i>header-token</i> <i>argument-token</i> (1, "sem ID", semaphore ID) <i>[argument-token]</i> <i>[ipc_perm-token]</i> <i>[slabel-token]</i> <i>[priv-token]</i> (if privilege used or required) <i>subject-token</i> <i>slabel-token</i> <i>return-token</i>			

TABLE B-126 semget(2)

Event Name	Event ID	Event Class	Mask
AUE_SEMGET	109	ip	0x00000200
Format: <i>header-token</i> <i>argument-token</i> (1, "sem key", semaphore key) <i>argument-token</i> (3, "sem flags", semaphore flags) <i>[ipc_perm-token]</i> <i>[slabel-token]</i> <i>[argument-token]</i> <i>[priv-token]</i> (if privilege used or required) <i>subject-token</i> <i>slabel-token</i> <i>return-token</i>			
The <i>ipc</i> , <i>ipc_perm</i> , and the <i>slabel</i> of the ipc tokens are not included if the semaphore ID is invalid.			

**TABLE B-127** semgetl(2)

Event Name	Event ID	Event Class	Mask
AUE_SEMGETL	177	ip	0x00000200

Format:

*header-token*

*argument-token* (1, "sem key", semaphore key)

*argument-token* (3, "sem flags", semaphore flags)

*slabel-token*

[*ipc\_perm-token*]

[*slabel-token*]

[*argument-token*]

[*priv-token*] (if privilege used or required)

*subject-token*

*slabel-token*

*return-token*

The *ipc*, *ipc\_perm*, and the *slabel* of the ipc tokens are not included if the system call failed.

**TABLE B-128** semop(2)

Event Name	Event ID	Event Class	Mask
AUE_SEMOP	110	ip	0x00000200

Format:

*header-token*

*argument-token* (1, "sem ID", semaphore ID)

[*argument-token*]

[*ipc\_perm-token*]

[*slabel-token*]

[*priv-token*] (if privilege used or required)

*subject-token*

*slabel-token*

*return-token*

The *ipc*, *ipc\_perm*, and the *slabel* of the ipc tokens are not included if the semaphore ID is invalid.

**TABLE B-129** setacl(1), setfac1(1)

Event Name	Event ID	Event Class	Mask
AUE_ACLSET	251	fm	0x00000008
AUE_FACLSET	252	fm	0x00000008

Format:

*header-token*

*argument-token* (2, "cmd", command)

*argument-token* (3, "n\_entries", number of acl entries)

*acl-token* ... (token repeated "n\_entries" times)

*path-token*

*[attr-token]*

*[priv-token]* (if privilege used or required)

*subject-token*

*return-token*

**TABLE B-130** setaudit(2)

Event Name	Event ID	Event Class	Mask
AUE_SETAUDIT	133	aa	0x00040000

Format (valid program stack address):

*header-token*

*argument-token* (1, "setaudit:aud", audit user ID)

*argument-token* (1, "setaudit:port", terminal ID)

*argument-token* (1, "setaudit:machine", terminal ID)

*argument-token* (1, "setaudit:as\_success", preselection mask)

*argument-token* (1, "setaudit:as\_failure", preselection mask)

*argument-token* (1, "setaudit:asid", audit session ID)

*[priv-token]* (if privilege used or required)

*subject-token*

*return-token*

Format (invalid program stack address):

*header-token*

*subject-token*

*return-token*

**TABLE B-131** setauid(2)

Event Name	Event ID	Event Class	Mask
AUE_SETAUID	131	aa	0x00040000
Format: <i>header-token</i> <i>argument-token</i> (2, "setauid", audit user ID) <i>[priv-token]</i> (if privilege used or required) <i>subject-token</i> <i>return-token</i>			

**TABLE B-132** setclearance(2)

Event Name	Event ID	Event Class	Mask
AUE_SETCLEARANCE	542	fm	0x00000008
Format: <i>header-token</i> <i>clearance-token</i> (specified) <i>clearance-token</i> (old) <i>clearance-token</i> (new) <i>[priv-token]</i> (if privilege used or required) <i>subject-token</i> <i>slabel-token</i> (subject) <i>return-token</i>			

**TABLE B-133** setcmwlabel(2), lsetcmwlabel(2)

Event Name	Event ID	Event Class	Mask
AUE_SETCMWLABEL	549	fm	0x00000008
AUE_LSETCMWLABEL	525	fm	0x00000008

Format:

*header-token*

*argument-token* (3, "flag", which parts of label to set)

[*slabel-token*] (if slabel is being set)

[*priv-token*] (if privilege used or required)

*subject-token*

*slabel-token*

*return-token*

**TABLE B-134** setcmwplabel(2)

Event Name	Event ID	Event Class	Mask
AUE_SETCMWPLABEL	541	fm	0x00000008
<p>Format (setting flag == SETCL_ALL):</p> <p><i>header-token</i></p> <p><i>slabel-token</i> (SL from input argument)</p> <p><i>slabel-token</i> (original SL)</p> <p><i>argument-token</i> (2, "flag", value)</p> <p><i>slabel-token</i> (new SL)</p> <p>[<i>priv-token</i>] (if privilege used or required)</p> <p><i>subject-token</i></p> <p><i>slabel-token</i> (subject)</p> <p><i>return-token</i></p> <p>Format (setting flag == SETCL_SL):</p> <p><i>header-token</i></p> <p><i>slabel-token</i> (SL from input argument)</p> <p><i>slabel-token</i> (SL of subject before)</p> <p><i>argument-token</i> (2, "flag", value)</p> <p><i>slabel-token</i> (SL of subject after)</p> <p>[<i>priv-token</i>] (if privilege used or required)</p> <p><i>subject-token</i></p> <p><i>slabel-token</i> (subject)</p> <p><i>return-token</i></p> <p>Format (setting flag == SETCL_IL):</p> <p><i>header-token</i></p> <p><i>argument-token</i> (2, "flag", value)</p> <p>[<i>priv-token</i>] (if privilege used or required)</p> <p><i>subject-token</i></p> <p><i>slabel-token</i> (subject)</p> <p><i>return-token</i></p>			

**TABLE B-135** setegid(2), old setgid(2)

Event Name	Event ID	Event Class	Mask
AUE_SETEGID	214	pm	0x00200000
AUE_SETGID	205	pm	0x00200000

Format:  
*header-token*  
*argument-token* (1, "gid", group ID)  
*[priv-token]* (if privilege used or required)  
*subject-token*  
*slabel-token* (subject)  
*return-token*

**TABLE B-136** seteuid(2)

Event Name	Event ID	Event Class	Mask
AUE_SETEUID	215	pm	0x00200000

Format:  
*header-token*  
*argument-token* (1, "gid", user ID)  
*[priv-token]* (if privilege used or required)  
*subject-token*  
*slabel-token* (subject)  
*return-token*

**TABLE B-137** setfattrflag(2)

Event Name	Event ID	Event Class	Mask
AUE_SETFATTRFLAG	522	fm	0x00000008
Format: <i>header-token</i> <i>argument-token</i> (2, "which", which flags to set) <i>argument-token</i> (3, "attrs", flag values) <i>[priv-token]</i> (if privilege used or required) <i>subject-token</i> <i>slabel-token</i> <i>return-token</i>			

**TABLE B-138** setfpriv(2)

Event Name	Event ID	Event Class	Mask
AUE_SETFILEPRIV	550	fm	0x00000008
Format: <i>header-token</i> <i>argument-token</i> (4, "privilege type", privilege set type) <i>privilege-token</i> <i>[priv-token]</i> (if privilege used or required) <i>subject-token</i> <i>slabel-token</i> <i>return-token</i>			



TABLE B-139 setgroups(2)

Event Name	Event ID	Event Class	Mask
AUE_SETGROUPS	26	pm	0x00200000
Format: <i>header-token</i> <i>[argument-token]</i> (1, "setgroups", group ID) <i>[priv-token]</i> (if privilege used or required) <i>subject-token</i> <i>slabel-token</i> (subject) <i>return-token</i> One <i>argument-token</i> for each group set.			

TABLE B-140 setpattr(2)

Event Name	Event ID	Event Class	Mask
AUE_SETPATTR	526	ps	0x00100000
Format: <i>header-token</i> <i>argument-token</i> (1, "type", type of attribute to set) <i>argument-token</i> (2, "value", value of attribute) <i>[priv-token]</i> (if privilege used or required) <i>subject-token</i> <i>slabel-token</i> <i>return-token</i>			

TABLE B-141 setpgrp(2)

Event Name	Event ID	Event Class	Mask
AUE_SETPGRP	27	pm	0x00200000
Format: <i>header-token</i> <i>[priv-token]</i> (if privilege used or required) <i>subject-token</i> <i>slabel-token</i> <i>return-token</i>			

TABLE B-141 setpgrp(2) (continued)

TABLE B-142 setppriv(2)

Event Name	Event ID	Event Class	Mask
AUE_SETPROCPRIV	127	fm	0x00000008
Format: <i>header-token</i> <i>argument-token</i> (3, "type", privilege set type) <i>argument-token</i> (4, "op", operation to perform) <i>privilege-token</i> (specified) <i>privilege-token</i> (old) <i>[priv-token]</i> (if privilege used or required) <i>subject-token</i> <i>slabel-token</i> <i>return-token</i>			

TABLE B-143 setregid(2)

Event Name	Event ID	Event Class	Mask
AUE_SETREGID	41	pm	0x00200000
Format: <i>header-token</i> <i>argument-token</i> (1, "rgid", real group ID) <i>argument-token</i> (1, "egid", effective group ID) <i>[priv-token]</i> (if privilege used or required) <i>subject-token</i> <i>return-token</i>			

TABLE B-144    setreuid(2)

Event Name	Event ID	Event Class	Mask
AUE_SETREUID	40	pm	0x00200000
Format: <i>header-token</i> <i>argument-token</i> (1, "ruid", real user ID) <i>argument-token</i> (1, "euid", effective user ID) <i>[priv-token]</i> (if privilege used or required) <i>subject-token</i> <i>return-token</i>			

TABLE B-145    setrlimit(2)

Event Name	Event ID	Event Class	Mask
AUE_SETRLIMIT	51	as	0x00020000
Format: <i>header-token</i> <i>[priv-token]</i> (if privilege used or required) <i>subject-token</i> <i>slabel-token</i> (subject) <i>return-token</i>			

TABLE B-146 `old setuid(2)`

Event Name	Event ID	Event Class	Mask
AUE_OSETUID	200	pm	0x00200000

Format:

*header-token*

*argument-token* (1, "uid", user ID)

[*priv-token*] (if privilege used or required)

*subject-token*

*slabel-token* (subject)

*return-token*

Due to a current bug in the audit software, this token is reported as AUE\_OSETUID.

TABLE B-147 `shmat(2)`

Event Name	Event ID	Event Class	Mask
AUE_SHMAT	96	ip	0x00000200

Format:

*header-token*

*argument-token* (1, "shm ID", shared memory ID)

*argument-token* (2, "shm adr", shared mem addr)

[*argument-token*]

[*ipc\_perm-token*]

[*slabel-token*]

[*priv-token*] (if privilege used or required)

*subject-token*

*slabel-token*

*return-token*

The *ipc*, *ipc\_perm*, and *slabel* tokens are not included if the shared memory segment ID is invalid.

TABLE B-148 shmctl(2) — IPC\_RMID command

Event Name	Event ID	Event Class	Mask
AUE_SHMCTL_RMID	92	ip	0x00000200

Format:

*header-token*

*argument-token* (1, "shm ID", shared memory ID)

[*argument-token*]

[*ipc\_perm-token*]

[*slabel-token*]

[*priv-token*] (if privilege used or required)

*subject-token*

*slabel-token*

*return-token*

The *ipc*, *ipc\_perm*, and *slabel* tokens are not included if the shared memory segment ID is invalid.

TABLE B-149 shmctl(2) — IPC\_SET command

Event Name	Event ID	Event Class	Mask
AUE_SHMCTL_SET	93	ip	0x00000200

Format:

*header-token*

*argument-token* (1, "shm ID", shared memory ID)

[*argument-token*]

[*ipc\_perm-token*] (of the IPC's old values)

[*slabel-token*]

[*ipc\_perm-token*] (of the IPC's new values)

[*slabel-token*]

[*priv-token*] (if privilege used or required)

*subject-token*

*slabel-token*

*return-token*

The *ipc*, *ipc\_perm*, and *slabel* tokens are not included if the shared memory segment ID is invalid.

**TABLE B-150** shmctl(2) — IPC\_STAT command

Event Name	Event ID	Event Class	Mask
AUE_SHMCTL_STAT	94	ip	0x00000200

Format:

*header-token*

*argument-token* (1, "shm ID", shared memory ID)

[*argument-token*]

[*ipc\_perm-token*]

[*slabel-token*]

[*priv-token*] (if privilege used or required)

*subject-token*

*slabel-token*

*return-token*

The *ipc*, *ipc\_perm*, and *slabel* tokens are not included if the shared memory segment ID is invalid.

**TABLE B-151** shmdt(2)

Event Name	Event ID	Event Class	Mask
AUE_SHMDT	97	ip	0x00000200

Format:

*header-token*

*argument-token* (1, "shm adr", shared mem addr)

[*priv-token*] (if privilege used or required)

*subject-token*

*slabel-token*

*return-token*

TABLE B-152 shmget(2)

Event Name	Event ID	EventClass	Mask
AUE_SHMGET	95	ip	0x00000200

Format:

*header-token*

*argument-token* (1, "shm ID", shared memory ID)

*argument-token* (3, "shm flag", shared memory flags)

*[argument-token]*

*[slabel-token]*

*[ipc\_perm-token]* (of the IPC's old values)

*[slabel-token]*

*[ipc\_perm-token]* (of the IPC's new values)

*[slabel-token]*

*[priv-token]* (if privilege used or required)

*subject-token*

*slabel-token*

*subject-token*

The *ipc*, *ipc\_perm*, and *slabel* tokens are not included for failed events.

TABLE B-153 shmget1(2)

Event Name	Event ID	EventClass	Mask
AUE_SHMGETL	178	ip	0x00000200

Format:

*header-token*

*argument-token* (1, "shm ID", shared memory ID)

*argument-token* (3, "shm flag", shared memory flags)

*slabel-token*

*[ipc\_perm-token]* (of the IPC's old values)

*[slabel-token]*

*[ipc\_perm-token]* (of the IPC's new values)

*[slabel-token]*

*[priv-token]* (if privilege used or required)

*subject-token*

*slabel-token*

*subject-token*

The *ipc*, *ipc\_perm*, and *slabel* tokens are not included for failed events.

**TABLE B-154** stat(2), statfs(2), statvfs(2)

Event Name	Event ID	Event Class	Mask
AUE_STAT	16	fa	0x00000004
AUE_STATFS	54	fa	0x00000004
AUE_STATVFS	234	fa	0x00000004
Format: <i>header-token</i> <i>path-token</i> <i>[attr-token]</i> <i>[slabel-token]</i> (object) <i>[priv-token]</i> (if privilege used or required) <i>subject-token</i> <i>slabel-token</i> (subject) <i>return-token</i>			

**TABLE B-155** stime(2)

Event Name	Event ID	Event Class	Mask
AUE_STIME	201	as	0x00020000
Format: <i>header-token</i> <i>[priv-token]</i> (if privilege used or required) <i>subject-token</i> <i>return-token</i>			



TABLE B-156 symlink(2)

Event Name	Event ID	Event Class	Mask
AUE_SYMLINK	21	fc	0x00000010
Format: <i>header-token</i> <i>text-token</i> (symbolic link string) <i>path-token</i> <i>[attr-token]</i> <i>[slabel-token]</i> (object) <i>[priv-token]</i> (if privilege used or required) <i>subject-token</i> <i>slabel-token</i> (subject) <i>return-token</i>			

TABLE B-157 sysinfo(2)

Event Name	Event ID	Event Class	Mask
AUE_SYSINFO	39	as	0x00020000
Format: <i>header-token</i> <i>argument-token</i> (1, "cmd", command) <i>text-token</i> (name) <i>[priv-token]</i> (if privilege used or required) <i>subject-token</i> <i>return-token</i>			

TABLE B-158 system booted

Event Name	Event ID	Event Class	Mask
AUE_SYSTEMBOOT	113	na	0x00000400
Format: <i>header-token</i> <i>text-token</i> ("booting kernel") <i>return-token</i>			

**TABLE B-158** system booted (continued)

**TABLE B-159** tnif(2), tnrh(2), tnrhtp(2)

Event Name	Event ID	Event Class	Mask
AUE_TNIF	534	nt	0x00000100
AUE_TNRH	535		
AUE_TNRHTP	536		
Format: <i>header-token</i> <i>[priv-token]</i> (if privilege used or required) <i>subject-token</i> <i>slabel-token</i> <i>return-token</i>			

**TABLE B-160** tokmapper(2)

Event Name	Event ID	Event Class	Mask
AUE_TOKMAPPER	537	nt	0x00000100
Format: <i>header-token</i> <i>argument-token</i> (1, "op", state) <i>in_addr-token</i> <i>[priv-token]</i> (if privilege used or required) <i>subject-token</i> <i>slabel-token</i> <i>return-token</i>			

**TABLE B-161** uadmin(2) - system freeze

Event Name	Event ID	Event Class	Mask
AUE_FREEZE	539	ss	0x00010000
Format: <i>header-token</i> <i>[priv-token]</i> (if privilege used or required) <i>subject-token</i> <i>slabel-token</i> <i>return-token</i>			

**TABLE B-162** uadmin(2) - system reboot

Event Name	Event ID	Event Class	Mask
AUE_REBOOT	561	ss	0x00010000
Format: <i>header-token</i> <i>[priv-token]</i> (if privilege used or required) <i>subject-token</i> <i>slabel-token</i> <i>return-token</i>			

**TABLE B-163** uadmin(2) - system remount

Event Name	Event ID	Event Class	Mask
AUE_REMOUNT	540	as	0x00020000
Format: <i>header-token</i> <i>[priv-token]</i> (if privilege used or required) <i>subject-token</i> <i>slabel-token</i> <i>return-token</i>			

**TABLE B-164** uadmin(2) - system shutdown

Event Name	Event ID	Event Class	Mask
AUE_SHUTDOWN	560	ss	0x00010000
Format: <i>header-token</i> <i>[priv-token]</i> (if privilege used or required) <i>subject-token</i> <i>slabel-token</i> <i>return-token</i>			

**TABLE B-165** umount(2) — old version

Event Name	Event ID	Event Class	Mask
AUE_UMOUNT	12	ao	0x00080000
Format: <i>header-token</i> <i>path-token</i> <i>[attr-token]</i> <i>[slabel-token]</i> (object) <i>[priv-token]</i> (if privilege used or required) <i>subject-token</i> <i>slabel-token</i> (subject) <i>return-token</i>			

TABLE B-166 unlink(2)

Event Name	Event ID	Event Class	Mask
AUE_UNLINK	6	fd	0x00000020
Format: <i>header-token</i> <i>path-token</i> [ <i>attr-token</i> ] [ <i>slabel-token</i> ] (object) [ <i>priv-token</i> ] (if privilege used or required) <i>subject-token</i> <i>slabel-token</i> (subject) <i>return-token</i>			

TABLE B-167 old utime(2), utimes(2)

Event Name	Event ID	Event Class	Mask
AUE_UTIME	202	fm	0x00000008
AUE_UTIMES	49	fm	0x00000008
Format: <i>header-token</i> <i>path-token</i> [ <i>attr-token</i> ] [ <i>priv-token</i> ] (if privilege used or required) <i>subject-token</i> <i>return-token</i>			

**TABLE B-168** utssys(2) — fusers

Event Name	Event ID	Event Class	Mask
AUE_UTSSYS	233	ao	0x00080000
Format: <i>header-token</i> <i>path-token</i> <i>[attr-token]</i> <i>[priv-token]</i> (if privilege used or required) <i>subject-token</i> <i>return-token</i>			

**TABLE B-169** vfork(2)

Event Name	Event ID	Event Class	Mask
AUE_VFORK	25	ps	0x00100000
Format: <i>header-token</i> <i>argument-token</i> (0, "child PID", pid) <i>[priv-token]</i> (if privilege used or required) <i>subject-token</i> <i>slabel-token</i> (subject) <i>return-token</i>			
The <code>fork</code> return values are undefined since the audit record is produced at the point that the child process is spawned.			

TABLE B-170 vtrace(2)

Event Name	Event ID	Event Class	Mask
AUE_VTRACE	36	pm	0x00200000
Format: <i>header-token</i> <i>[priv-token]</i> (if privilege used or required) <i>subject-token</i> <i>return-token</i>			

TABLE B-171 write(2)

Event Name	Event ID	Event Class	Mask
AUE_WRITE	195	no	0x00000000
Format: <i>header-token</i> <i>slabel-token</i> (from label specified in syscall args) <i>path-token</i> <i>[attr-token]</i> <i>[slabel-token]</i> <i>[priv-token]</i> (if privilege used or required) <i>subject-token</i> <i>slabel-token</i> <i>return-token</i>			

TABLE B-172 writel(2), pwrite(2), writev(2)

Event Name	Event ID	Event Class	Mask
AUE_PWRITE	528	no	0x00000000
AUE_WRITE	552	fm	0x00000008

**TABLE B-172** writel(2), pwritel(2), writev(2) (continued)

Event Name	Event ID	Event Class	Mask
AUE_WRITEVL	553	fm	0x00000008
Format: <i>header-token</i> <i>slabel-token</i> (from label specified in syscall args) <i>path-token</i> <i>[attr-token]</i> <i>[slabel-token]</i> <i>[priv-token]</i> (if privilege used or required) <i>subject-token</i> <i>slabel-token</i> <i>return-token</i>			

**TABLE B-173** xmknod(2)

Event Name	Event ID	Event Class	Mask
AUE_XMKNOD	240	fc	0x00000010
Format: <i>header-token</i> <i>path-token</i> <i>[attr-token]</i> <i>[slabel-token]</i> (object) <i>[priv-token]</i> (if privilege used or required) <i>subject-token</i> <i>slabel-token</i> (subject) <i>return-token</i>			



TABLE B-174 xstat(2)

Event Name	Event ID	Event Class	Mask
AUE_XSTAT	235	fa	0x00000004

Format:  
*header-token*  
*path-token*  
*[attr-token]*  
*[slabel-token]* (object)  
*[priv-token]* (if privilege used or required)  
*subject-token*  
*slabel-token* (subject)  
*return-token*

## Kernel-Level Pseudo-Events

Pseudo-events do have their own audit record structure. They create audit records for the event that uses privilege. When the pseudo-event AUE\_UPRIV is in a class that is being audited, any use of privilege will be audited, including uses of privilege for events that are otherwise not being audited.

TABLE B-175 Use of privilege

Event Name	Event ID	Event Class	Mask
AUE_UPRIV	521	no	0x00000000

## X Server Protocol Audit Records

These audit records are created by X windows calls and use of the X server. The records are sorted alphabetically by protocol; where possible, records with identical structure are listed together. The description of each record includes:

- The name of the protocol
- The audit event number
- The audit event name

■ The audit record structure

**TABLE B-176** XClientConnect

Event Name	Message	Event ID	Event Class	Mask
AUE_ClientConnect	Client connection to Xserver	9101	x1	0x08000000
Format: <i>header-token</i> <i>subject-token</i> <i>newgroups-token</i> <i>slabel-token</i> <i>xclient-token</i> <i>inaddr-token</i> (IP address of client) <i>ipport-token</i> (port on server) <i>return-token</i>				

**TABLE B-177** XClientDisconnect

Event Name	Message	Event ID	Event Class	Mask
AUE_ClientDisconnect	Client logout from Xserver	9102	x1	0x08000000
Format: <i>header-token</i> <i>subject-token</i> <i>newgroups-token</i> <i>slabel-token</i> <i>xclient-token</i> <i>return-token</i>				

**TABLE B-178** X Server Protocols - window operations

Event Name	Message	Event ID	Event Class	Mask
AUE_ChangeSaveSet	Change the saved set	9108	xp	0x10000000
AUE_ChangeWindowAttributes	Change window attributes	9104		
AUE_CirculateWindow	Circulate the window	9115		
AUE_ConfigureWindow	Configure the window	9114		
AUE_CreateWindow	Create window	9103		
AUE_DestroySubwindows	Destroy subwindows	9107		
AUE_DestroyWindow	Destroy window	9106		
AUE_GetGeometry	Get window geometry	9116		
AUE_GetWindowAttributes	Get window attributes	9105		
AUE_MapSubwindows	Map the subwindows	9111		
AUE_MapWindow	Map the window	9110		
AUE_QueryTree	Query window tree	9117		
AUE_ReparentWindow	Reparent the window	9109		
AUE_UnmapSubwindows	Unmap the subwindows	9113		

**TABLE B-178** X Server Protocols - window operations *(continued)*

Event Name	Message	Event ID	Event Class	Mask
AUE_UnmapWindow	Unmap the window	9112		
Format: <i>header-token</i> <i>subject-token</i> <i>newgroups-token</i> <i>slabel-token</i> <i>[priv-token]</i> (if privilege used or required) <i>xwindow-token</i> <i>return-token</i>				

**TABLE B-179** X Server Protocols - window properties

Event Name	Message	Event ID	Event Class	Mask
AUE_ChangeProperty	Change window property	9120	xc	0x20000000
AUE_DeleteProperty	Delete window property	9121	xc	0x20000000
AUE_GetProperty	Get window property	9122	xp	0x10000000
AUE_ListProperties	List window properties	9123	xp	0x10000000
Format: <i>header-token</i> <i>subject-token</i> <i>newgroups-token</i> <i>slabel-token</i> <i>[priv-token]</i> (if privilege used or required) <i>xwindow-token</i> <i>xproperty-token</i> <i>return-token</i>				

**TABLE B-180** XGetAtomName, XInternAtom

Event Name	Message	Event ID	Event Class	Mask
AUE_GetAtomName	Get atom name	9119	xs	0x80000000
AUE_InternAtom	Fetch atom	9118	xs	0x80000000
Format: <i>header-token</i> <i>subject-token</i> <i>newgroups-token</i> <i>slabel-token</i> <i>[priv-token]</i> (if privilege used or required) <i>xatom-token</i> (atom string) <i>return-token</i>				

**TABLE B-181** XConvertSelection, XGetSelectionOwner, XSetSelectionOwner

Event Name	Message	Event ID	Event Class	Mask
AUE_ConvertSelection	Convert selection	9126	xs	0x80000000
AUE_GetSelectionOwner	Get selection owner	9125	xs	0x80000000
AUE_SetSelectionOwner	Set selection owner	9124	xp	0x10000000
Format: <i>header-token</i> <i>subject-token</i> <i>newgroups-token</i> <i>slabel-token</i> <i>[priv-token]</i> (if privilege used or required) <i>xclient-token</i> <i>return-token</i>				

TABLE B-182 XGrabButton

Event Name	Message	Event ID	Event Class	Mask
AUE_GrabButton	Grab window button	9130	xp	0x10000000
Format: <i>header-token</i> <i>subject-token</i> <i>newgroups-token</i> <i>slabel-token</i> <i>[priv-token]</i> (if privilege used or required) <i>xwindow-token</i> (grabbing window id) <i>xwindow-token</i> (current device focus) <i>xcursor-token</i> <i>return-token</i>				

TABLE B-183 XGrabPointer, XUngrabPointer, XUngrabButton

Event Name	Message	Event ID	Event Class	Mask
AUE_GrabPointer	Grab pointer	9128	xs	0x80000000
AUE_UngrabButton	Release window button	9131	xs	0x80000000
AUE_UngrabPointer	Release pointer	9129	xs	0x80000000
Format: <i>header-token</i> <i>subject-token</i> <i>newgroups-token</i> <i>slabel-token</i> <i>[priv-token]</i> (if privilege used or required) <i>xwindow-token</i> (grabbing window id) <i>xwindow-token</i> (current device focus) <i>xcursor-token</i> <i>return-token</i>				

TABLE B-184 XChangeActivePointerGrab

Event Name	Message	Event ID	Event Class	Mask
AUE_ChangeActivePointerGrab	Change active pointer grab	9132	xs	0x80000000
Format: <i>header-token</i> <i>subject-token</i> <i>newgroups-token</i> <i>slabel-token</i> <i>[priv-token]</i> (if privilege used or required) <i>xcursor-token</i> <i>return-token</i>				

TABLE B-185 XGrabKey, XUngrabKeyboard

Event Name	Message	Event ID	Event Class	Mask
AUE_GrabKey	Grab key	9135	xs	0x80000000
AUE_UngrabKeyboard	Release keyboard	9134	xs	0x80000000
Format: <i>header-token</i> <i>subject-token</i> <i>newgroups-token</i> <i>slabel-token</i> <i>[priv-token]</i> (if privilege used or required) <i>xwindow-token</i> <i>return-token</i>				

**TABLE B-186** XGrabKeyboard, XUngrabKey

Event Name	Message	Event ID	Event Class	Mask
AUE_GrabKeyboard	Grab keyboard	9133	xp	0x10000000
AUE_UngrabKey	Release key	9135	xp	0x10000000
Format: <i>header-token</i> <i>subject-token</i> <i>newgroups-token</i> <i>slabel-token</i> <i>[priv-token]</i> (if privilege used or required) <i>xwindow-token</i> <i>return-token</i>				

**TABLE B-187** XGrabServer, XUngrabServer

Event Name	Message	Event ID	Event Class	Mask
AUE_GrabServer	Grab the server	9137	xa	0x40000000
AUE_UngrabServer	Release the server	9138	xa	0x40000000
Format: <i>header-token</i> <i>subject-token</i> <i>newgroups-token</i> <i>slabel-token</i> <i>[priv-token]</i> (if privilege used or required) <i>xclient-token</i> <i>return-token</i>				



TABLE B-188 XQueryPointer

Event Name	Message	Event ID	Event Class	Mask
AUE_QueryPointer	Query pointer	9139	xp	0x10000000
Format: <i>header-token</i> <i>subject-token</i> <i>newgroups-token</i> <i>slabel-token</i> <i>[priv-token]</i> (if privilege used or required) <i>xwindow-token</i> (querying window id) <i>xwindow-token</i> (pointer's window id) <i>return-token</i>				

TABLE B-189 XGetMotionEvents, XSendEvent

Event Name	Message	Event ID	Event Class	Mask
AUE_GetMotionEvents	Get motion events	9140	xp	0x10000000
AUE_SendEvent	Send window event	9127	xs	0x80000000
Format: <i>header-token</i> <i>subject-token</i> <i>newgroups-token</i> <i>slabel-token</i> <i>[priv-token]</i> (if privilege used or required) <i>xwindow-token</i> <i>return-token</i>				

**TABLE B-190** XTranslateCoords, XWarpPointer

Event Name	Message	Event ID	Event Class	Mask
AUE_TranslateCoords	Translate coordinates	9141	xp	0x10000000
AUE_WarpPointer	Warp the pointer	9142	xs	0x80000000
Format: <i>header-token</i> <i>subject-token</i> <i>newgroups-token</i> <i>slabel-token</i> <i>[priv-token]</i> (if privilege used or required) <i>xwindow-token</i> (source window id) <i>xwindow-token</i> (destination window id) <i>return-token</i>				

**TABLE B-191** XGetInputFocus, XSetInputFocus

Event Name	Message	Event ID	Event Class	Mask
AUE_GetInputFocus	Get input focus	9144	xs	0x80000000
AUE_SetInputFocus	Set input focus	9143	xs	0x80000000
Format: <i>header-token</i> <i>subject-token</i> <i>newgroups-token</i> <i>slabel-token</i> <i>[priv-token]</i> (if privilege used or required) <i>xwindow-token</i> <i>return-token</i>				

TABLE B-192 XQueryKeymap

Event Name	Message	Event ID	Event Class	Mask
AUE_QueryKeymap	Query keymap	9145	xp	0x10000000
Format: <i>header-token</i> <i>subject-token</i> <i>newgroups-token</i> <i>slabel-token</i> <i>[priv-token]</i> (if privilege used or required) <i>xclient-token</i> <i>return-token</i>				

TABLE B-193 XSetFontPath

Event Name	Message	Event ID	Event Class	Mask
AUE_SetFontPath	Set font path	9146	xa	0x40000000
Format: <i>header-token</i> <i>subject-token</i> <i>newgroups-token</i> <i>slabel-token</i> <i>[priv-token]</i> (if privilege used or required) <i>xwindow-token</i> <i>xfont-token</i> <i>return-token</i>				

TABLE B-194 XChangeGC

Event Name	Message	Event ID	Event Class	Mask
AUE_ChangeGC	Change graphical context	9148	xp	0x10000000
Format: <i>header-token</i> <i>subject-token</i> <i>newgroups-token</i> <i>slabel-token</i> <i>[priv-token]</i> (if privilege used or required) <i>xfont-token</i> <i>xpixmap-token</i> <i>xgc-token</i> <i>return-token</i>				

TABLE B-195 XCopyGC

Event Name	Message	Event ID	Event Class	Mask
AUE_CopyGC	Copy graphical context	9149	xp	0x10000000
Format: <i>header-token</i> <i>subject-token</i> <i>newgroups-token</i> <i>slabel-token</i> <i>[priv-token]</i> (if privilege used or required) <i>xgc-token</i> (source gc ID) <i>xgc-token</i> (destination gc ID) <i>return-token</i>				

**TABLE B-196** XFreeGC, XSetClipRectangles, XSetDashes

Event Name	Message	Event ID	Event Class	Mask
AUE_FreeGC	Free graphical context	9152	xc	0x20000000
AUE_SetClipRectangles	Set clip rectangles	9151	xp	0x10000000
AUE_SetDashes	Set dashes	9150	xp	0x10000000

Format:  
*header-token*  
*subject-token*  
*newgroups-token*  
*slabel-token*  
*[priv-token]* (if privilege used or required)  
*xgc-token*  
*return-token*

**TABLE B-197** XClearArea

Event Name	Message	Event ID	Event Class	Mask
AUE_ClearArea	Clear area	9153	xp	0x10000000

Format:  
*header-token*  
*subject-token*  
*newgroups-token*  
*slabel-token*  
*[priv-token]* (if privilege used or required)  
*xwindow-token*  
*return-token*

**TABLE B-198** XCopyArea, XCopyPlane

Event Name	Message	Event ID	Event Class	Mask
AUE_CopyArea	Copy area	9154	xs	0x80000000
AUE_CopyPlane	Copy plane	9155	xs	0x80000000

Format:  
*header-token*  
*subject-token*  
*newgroups-token*  
*slabel-token*  
*[priv-token]* (if privilege used or required)  
*xpixmap-token* (source pixmap ID)  
*xpixmap-token* (destination pixmap ID)  
*xgc-token*  
*return-token*

**TABLE B-199** XFillPolygon, XPolyArc, XPolyFillArc, XPolyFillRectangle, XPolyLine, XPolyPoint, XPolyRectangle, XPolySegment

Event Name	Message	Event ID	Event Class	Mask
AUE_FillPolygon	Fill polygon	9161	xp	0x10000000
AUE_PolyArc	Polyarc	9160	xp	0x10000000
AUE_PolyFillArc	Fill polyarc	9163	xp	0x10000000
AUE_PolyFillRectangle	Fill polyrectangle	9162	xp	0x10000000
AUE_PolyLine	Polyline	9157	xp	0x10000000
AUE_PolyPoint	Polypoint	9156	xp	0x10000000
AUE_PolyRectangle	Polyrectangle	9159	xs	0x80000000

**TABLE B-199** XFillPolygon, XPolyArc, XPolyFillArc, XPolyFillRectangle, XPolyLine, XPolyPoint, XPolyRectangle, XPolySegment *(continued)*

Event Name	Message	Event ID	Event Class	Mask
AUE_PolySegment	Polysegment	9158	xp	0x10000000
Format: <i>header-token</i> <i>subject-token</i> <i>newgroups-token</i> <i>slabel-token</i> <i>[priv-token]</i> (if privilege used or required) <i>xwindow-token</i> <i>xpixmap-token</i> <i>xgc-token</i> <i>return-token</i>				

**TABLE B-200** XGetImage, XImageText8, XImageText16, XPolyText8, XPolyText16, XPutImage

Event Name	Message	Event ID	Event Class	Mask
AUE_GetImage	Get image	9165	xs	0x80000000
AUE_ImageText8	Imagetext (8-bit)	9168	xp	0x10000000
AUE_ImageText16	Imagetext (16-bit)	9169	xp	0x10000000
AUE_PolyText8	Polytext (8-bit)	9166	xp	0x10000000
AUE_PolyText16	Polytext (16-bit)	9167	xp	0x10000000

**TABLE B-200** XGetImage, XImageText8, XImageText16, XPolyText8, XPolyText16, XPutImage (continued)

Event Name	Message	Event ID	Event Class	Mask
AUE_PutImage	Put image	9164	xp	0x10000000
Format: <i>header-token</i> <i>subject-token</i> <i>newgroups-token</i> <i>slabel-token</i> <i>[priv-token]</i> (if privilege used or required) <i>xwindow-token</i> <i>xpixmap-token</i> <i>xgc-token</i> <i>return-token</i>				

**TABLE B-201** XCreateColormap

Event Name	Message	Event ID	Event Class	Mask
AUE_CreateColormap	Create colormap	9170	xc	0x20000000
Format: <i>header-token</i> <i>subject-token</i> <i>newgroups-token</i> <i>slabel-token</i> <i>[priv-token]</i> (if privilege used or required) <i>xwindow-token</i> <i>return-token</i>				

**TABLE B-202** XAllocColor, XAllocColorCells, XAllocColorPlanes, XAllocNamedColor,



**TABLE B-202** XAllocColor, XAllocColorCells, XAllocColorPlanes, XAllocNamedColor, XFreeColors *(continued)*

XFreeColors

Event Name	Message	Event ID	Event Class	Mask
AUE_AllocColor	Allocate color	9176	xc	0x20000000
AUE_AllocColorCells	Allocate color cells	9178		
AUE_AllocColorPlanes	Allocate color planes	9179		
AUE_AllocNamedColor	Allocate named color	9177		
AUE_FreeColors	Free colors	9180		
Format: <i>header-token</i> <i>subject-token</i> <i>newgroups-token</i> <i>slabel-token</i> <i>[priv-token]</i> (if privilege used or required) <i>xcolormap-token</i> <i>return-token</i>				

**TABLE B-203** XCopyColormapAndFree, XFreeColormap, XInstallColormap, XListInstalledColormap, XUninstallColormap

Event Name	Message	Event ID	Event Class	Mask
AUE_CopyColormapAndFree	Copy and free colormap	9172	xp	0x10000000
AUE_FreeColormap	Free colormap	9171	xp	0x10000000
AUE_InstallColormap	Install colormap	9173	xa	0x40000000
AUE_ListInstalledColormap	List installed colormap	9175	xs	0x80000000

**TABLE B-203** XCopyColormapAndFree, XFreeColormap, XInstallColormap, XListInstalledColormap, XUninstallColormap (continued)

Event Name	Message	Event ID	Event Class	Mask
AUE_UninstallColormap	Uninstall colormap	9174	xp	0x10000000
Format: <i>header-token</i> <i>subject-token</i> <i>newgroups-token</i> <i>slabel-token</i> <i>[priv-token]</i> (if privilege used or required) <i>xcolormap-token</i> <i>return-token</i>				

**TABLE B-204** XLookupColor, XQueryColors, XStoreColors, XStoreNamedColor

Event Name	Message	Event ID	Event Class	Mask
AUE_LookupColor	Look up colors	9184	xp	0x10000000
AUE_QueryColors	Query colors	9183	xp	0x10000000
AUE_StoreColors	Store colors	9181	xp	0x10000000
AUE_StoreNamedColor	Store named colors	9182	xp	0x10000000
Format: <i>header-token</i> <i>subject-token</i> <i>newgroups-token</i> <i>slabel-token</i> <i>[priv-token]</i> (if privilege used or required) <i>xcolormap-token</i> <i>return-token</i>				

TABLE B-205 XCreateCursor

Event Name	Message	Event ID	Event Class	Mask
AUE_CreateCursor	Create cursor	9185	xc	0x20000000
Format: <i>header-token</i> <i>subject-token</i> <i>newgroups-token</i> <i>slabel-token</i> <i>[priv-token]</i> (if privilege used or required) <i>xpixmap-token</i> (source pixmap ID) <i>xpixmap-token</i> (mask pixmap ID) <i>xcursor-token</i> <i>return-token</i>				

TABLE B-206 XCreateGlyphCursor

Event Name	Message	Event ID	Event Class	Mask
AUE_CreateGlyphCursor	Create glyph cursor	9186	xc	0x20000000
Format: <i>header-token</i> <i>subject-token</i> <i>newgroups-token</i> <i>slabel-token</i> <i>[priv-token]</i> (if privilege used or required) <i>xfont-token</i> (source font ID) <i>xfont-token</i> (mask font ID) <i>xcursor-token</i> <i>return-token</i>				

**TABLE B-207** XFreeCursor, XRecolorCursor

Event Name	Message	Event ID	Event Class	Mask
AUE_FreeCursor	Free cursor	9187	xc	0x20000000
AUE_RecolorCursor	Recolor cursor	9188	xp	0x10000000

Format:  
*header-token*  
*subject-token*  
*newgroups-token*  
*slabel-token*  
*[priv-token]* (if privilege used or required)  
*xcursor-token*  
*return-token*

**TABLE B-208** XFreePixmap

Event Name	Message	Event ID	Event Class	Mask
AUE_FreePixmap	Free pixmap	9147	xc	0x20000000

Format:  
*header-token*  
*subject-token*  
*newgroups-token*  
*slabel-token*  
*[priv-token]* (if privilege used or required)  
*xpixmap-token*  
*return-token*

**TABLE B-209** XBell, XChangeKeyboardControl, XChangeKeyboardMapping,

**TABLE B-209** XBell, XChangeKeyboardControl, XChangeKeyboardMapping, XChangePointerControl *(continued)*

XChangePointerControl

Event Name	Message	Event ID	Event Class	Mask
AUE_Bell	Bell	9193	xs	0x80000000
AUE_ChangeKeyboardControl	Change keyboard control	9190		
AUE_ChangeKeyboardMapping	Change keyboard mapping	9189		
AUE_ChangePointerControl	Change pointer control	9192		
Format: <i>header-token</i> <i>subject-token</i> <i>newgroups-token</i> <i>slabel-token</i> <i>[priv-token]</i> (if privilege used or required) <i>xclient-token</i> <i>return-token</i>				

**TABLE B-210** XForceScreenSaver, XSetScreenSaver

Event Name	Message	Event ID	Event Class	Mask
AUE_ForceScreenSaver	Cover screen	9199	xp	0x10000000
AUE_SetScreenSaver	Set screensaver	9193		
Format: <i>header-token</i> <i>subject-token</i> <i>newgroups-token</i> <i>slabel-token</i> <i>[priv-token]</i> (if privilege used or required) <i>xclient-token</i> <i>return-token</i>				

**TABLE B-211** XSetCloseDownMode

Event Name	Message	Event ID	Event Class	Mask
AUE_SetCloseDownMode	Set closedown mode	9196	xs	0x80000000
Format: <i>header-token</i> <i>subject-token</i> <i>newgroups-token</i> <i>slabel-token</i> <i>[priv-token]</i> (if privilege used or required) <i>xclient-token</i> <i>return-token</i>				

**TABLE B-212** XChangeHosts, XKillClient, XSetAccessControl

Event Name	Message	Event ID	Event Class	Mask
AUE_ChangeHosts	Change hosts	9194	xa	0x40000000
AUE_KillClient	Kill client	9197	xc	0x20000000
AUE_SetAccessControl	Set access control	9195	xp	0x10000000
Format: <i>header-token</i> <i>subject-token</i> <i>newgroups-token</i> <i>slabel-token</i> <i>[priv-token]</i> (if privilege used or required) <i>xclient-token</i> <i>return-token</i>				

TABLE B-213 XRotateProperties

Event Name	Message	Event ID	Event Class	Mask
AUE_RotateProperties	Rotate properties	9198	xp	0x10000000
Format: <i>header-token</i> <i>subject-token</i> <i>newgroups-token</i> <i>slabel-token</i> <i>[priv-token]</i> (if privilege used or required) <i>xwindow-token</i> <i>xproperty-token</i> <i>return-token</i>				

TABLE B-214 XSetModifierMapping, XSetPointerMapping

Event Name	Message	Event ID	Event Class	Mask
AUE_SetModifierMapping	Set modifier mapping	9201	xs	0x80000000
AUE_SetPointerMapping	Set pointer mapping	9200	xs	0x80000000
Format: <i>header-token</i> <i>subject-token</i> <i>newgroups-token</i> <i>slabel-token</i> <i>[priv-token]</i> (if privilege used or required) <i>xclient-token</i> <i>return-token</i>				

TABLE B-215 X Server Extensions

Event Name	Message	Event ID	Event Class	Mask
AUE_XExtensions	X extension protocols	9202	xp	
Format: <i>header-token</i> <i>subject-token</i> <i>newgroups-token</i> <i>slabel-token</i> <i>[priv-token]</i> (if privilege used or required) <i>xclient-token</i> <i>return-token</i>				

The AUE\_XExtensions audit record format is used when auditing extensions to the X11 library, such as XTSOLMakeTPWindow.

## User-Level Generated Audit Records

These audit records are created by programs that operate outside the kernel. The records are sorted alphabetically by program. The description of each record includes:

- The name of the program
- A man page reference (if appropriate)
- The audit event number
- The audit event name
- The audit record structure



**TABLE B-216** add\_drv(1M)

Event Name	Program	Event ID	Event Class	Mask
AUE_add_drv	/usr/sbin/add_drv	9018	as	0x00020000
Format: <i>header-token</i> <i>subject-token</i> <i>groups-token</i> <i>slabel-token</i> <i>return-token</i> <i>exec_args-token</i> (command-line arguments) <i>text-token</i> (driver name) <i>text-token</i> (base directory) <i>text-token</i> (class name) <i>text-token</i> (aliases)				

**TABLE B-217** Admin Editor Action - Modify System Files

Event Name	Program	Event ID	Event Class	Mask
AUE_te_modsysfiles	trusted editor	9322	ao	0x00080000
Format: <i>header-token</i> <i>path-token</i> (filename) <i>text-token</i> (changes) <i>host-token</i> <i>return-token</i> <i>subject-token</i> <i>slabel-token</i>				

**TABLE B-218** allocate(1M) - device success

Event Name	Program	Event ID	Event Class	Mask
AUE_allocate_succ	/usr/sbin/allocate	6200	ao	0x00080000
Format: <i>header-token</i> <i>subject-token</i> <i>[slabel-token]</i> (subject) <i>newgroups-token</i> <i>exit-token</i>				

**TABLE B-219** allocate(1M) - device failure

Event Name	Program	Event ID	Event Class	Mask
AUE_allocate_fail	/usr/sbin/allocate	6201	ao	0x00080000
Format: <i>header-token</i> <i>subject-token</i> <i>[slabel-token]</i> (subject) <i>newgroups-token</i> <i>exit-token</i>				

**TABLE B-220** allocate(1M) - list devices success

Event Name	Program	Event ID	Event Class	Mask
AUE_listdevice_succ	/usr/sbin/allocate	6205	ao	0x00080000
Format: <i>header-token</i> <i>subject-token</i> <i>[slabel-token]</i> (subject) <i>newgroups-token</i> <i>exit-token</i>				

**TABLE B-220** allocate(1M) - list devices success *(continued)*

**TABLE B-221** allocate(1M) - list devices failure

Event Name	Program	Event ID	Event Class	Mask
AUE_listdevice_fail	/usr/sbin/allocate	6206	ao	0x00080000
Format: <i>header-token</i> <i>subject-token</i> <i>[slabel-token]</i> (subject) <i>newgroups-token</i> <i>exit-token</i>				

**TABLE B-222** at(1) - create atjob

Event Name	Program	Event ID	Event Class	Mask
AUE_at_create	/usr/bin/at	6144	ao	0x00080000
Format: <i>header-token</i> <i>subject-token</i> <i>return-token</i> <i>exec_args-token</i> <i>text-token</i> (user name) <i>text-token</i> (job queue)				

**TABLE B-223** at(1) - delete atjob file (at or atrm)

Event Name	Program	Event ID	Event Class	Mask
AUE_at_delete	/usr/bin/at /usr/bin/atrm	6145	ao	0x00080000
Format: <i>header-token</i> <i>subject-token</i> <i>return-token</i> <i>exec_args-token</i> <i>text-token</i> (user name) <i>text-token</i> (job queue)				

**TABLE B-224** at(1) - permission

Event Name	Program	Event ID	Event Class	Mask
AUE_at_perm	/usr/bin/at	6146	ao	0x00080000
Format: <i>header-token</i> <i>subject-token</i> <i>[group-token]</i> <i>exit-token</i>				

**TABLE B-225** auditd(1M)

Event Name	Program	Event ID	Event Class	Mask
AUE_audit	/usr/sbin/audit	9016	aa	0x00040000
Format: <i>header-token</i> <i>text-token</i> ("new audit file"   "reread audit_control"   "terminate auditd"   "unknown option" > <i>return-token</i> <i>subject-token</i> <i>slabel-token</i>				

TABLE B-225 auditd(1M) (continued)

TABLE B-226 auditwrite(3)

Event Name	Program	Event ID	Event Class	Mask
AUE_auditwrite	auditwrite( )	9015	aa	0x00040000
Format: <i>header-token</i> <i>text-token</i> (error description) <i>subject-token</i> <i>return-token</i>				

TABLE B-227 automountd(1M) – mismatch

Event Name	Program	Event ID	Event Class	Mask
AUE_automountd_mismatch	/usr/lib/fs/autofs/automount	9034	ao	0x00080000
Format: <i>header-token</i> <i>path-token</i> (mount dir) <i>slabel-token</i> (auto* file slabel) <i>slabel-token</i> (remote host template slabel) <i>text-token</i> (remote host server) <i>return-token</i>				

**TABLE B-228** automountd(1M) – mount

Event Name	Program	Event ID	Event Class	Mask
AUE_automountd_mount	/usr/lib/fs/autofs/automount	9033	ao	0x00080000

Format:  
*header-token*  
*subject-token*  
*slabel-token* (subject slabel)  
*path-token* (mount dir)  
*return-token*  
*host-token* (machine name)

**TABLE B-229** chroot(1M)

Event Name	Program	Event ID	Event Class	Mask
AUE_chroot	/usr/sbin/chroot	9029	ao	0x00080000

Format:  
*header-token*  
*subject-token*  
*groups-token*  
*slabel-token*  
*return-token*  
*exec\_args-token* (command-line arguments)  
*path-token* (new root directory)  
*path-token* (command to execute)

**TABLE B-230** crontab(1) - crontab created

Event Name	Program	Event ID	Event Class	Mask
AUE_crontab_create	/usr/bin/crontab	6148	ao	0x00080000
Format: <i>header-token</i> <i>subject-token</i> <i>return-token</i> <i>exec_args-token</i> <i>text-token</i> (user name)				

**TABLE B-231** crontab(1) - crontab deleted

Event Name	Program	Event ID	Event Class	Mask
AUE_crontab_delete	/usr/bin/crontab	6149	ao	0x00080000
Format: <i>header-token</i> <i>subject-token</i> <i>return-token</i> <i>exec_args-token</i> <i>text-token</i> (user name)				

**TABLE B-232** crontab(1) - invoke atjob or crontab

Event Name	Program	Event ID	Event Class	Mask
AUE_cron_invoke	/usr/bin/crontab	6147	ao	0x00080000
Format: <i>header-token</i> <i>subject-token</i> <i>return-token</i> <i>exec_args-token</i> <i>text-token</i> (user name) <i>text-token</i> (job type: cron or at) <i>text-token</i> (cron command or at job name)				

**TABLE B-233** crontab(1) - permission

Event Name	Program	Event ID	Event Class	Mask
AUE_crontab_perm	/usr/bin/crontab	6150	ao	0x00080000
Format: <i>header-token</i> <i>subject-token</i> <i>[group-token]</i> <i>exit-token</i>				

**TABLE B-234** dbmgr

Event Name	Program	Event ID	Event Class	Mask
AUE_dm_add	/opt/SUNWadm/2.3/bin/dbmgr	9319	ao	0x00080000
AUE_dm_del		9320		



**TABLE B-234** dbmgr (continued)

Event Name	Program	Event ID	Event Class	Mask
AUE_dm_mod		9321		
Format: <i>header-token</i> <i>text-token</i> (database info) <i>text-token</i> (database type) <i>text-token</i> (error message) <i>return-token</i> <i>subject-token</i> <i>slabel-token</i>				

**TABLE B-235** deallocate(1M) - device success

Event Name	Program	Event ID	Event Class	Mask
AUE_deallocate_succ	/usr/sbin/deallocate	6202	ao	0x00080000
Format: <i>header-token</i> <i>subject-token</i> [ <i>slabel-token</i> ] (subject) <i>newgroups-token</i> <i>exit-token</i>				

**TABLE B-236**    deallocate(1M) — device failure

Event Name	Program	Event ID	Event Class	Mask
AUE_deallocate_fail	/usr/sbin/deallocate	6203	ao	0x00080000
Format: <i>header-token</i> <i>subject-token</i> <i>[slabel-token]</i> (subject) <i>newgroups-token</i> <i>exit-token</i>				

**TABLE B-237**    dispadmin(1M)

Event Name	Program	Event ID	Event Class	Mask
AUE_dispadmin	/usr/sbin/dispadmin	9025	as	0x00020000
Format: <i>header-token</i> <i>subject-token</i> <i>groups-token</i> <i>slabel-token</i> <i>return-token</i> <i>exec_args-token</i> (command-line arguments) <i>text-token</i> (scheduler class) <i>path-token</i> (input file)				

**TABLE B-238** dtfile(1) - copy and move

Event Name	Program	Event ID	Event Class	Mask
AUE_dtfile_copy	/usr/dt/bin/dtfile	9037	fm	0x00000008
AUE_dtfile_move		9038		
Format: <i>header-token</i> <i>return-token</i> <i>path-token</i> (target path) <i>slabel-token</i> (slabel of target) <i>path-token</i> (source path) <i>slabel-token</i> (slabel of source) <i>host-token</i>				

**TABLE B-239** eeprom(1M)

Event Name	Program	Event ID	Event Class	Mask
AUE_eeprom	/usr/sbin/eeprom	9032	as	0x00020000
Format: <i>header-token</i> <i>return-token</i> <i>path-token</i> (prom device) <i>text-token</i> (variable=old value) <i>text-token</i> (variable=new value)				

**TABLE B-240** fuser(1M)

Event Name	Program	Event ID	Event Class	Mask
AUE_fuser	/usr/sbin/fuser	9031	ao	0x00080000

Format:  
*header-token*  
*subject-token*  
*groups-token*  
*slabel-token*  
*return-token*  
*exec\_args-token* (command-line arguments)  
*path-token* (file name)  
*arg-token* (1, "PID", process-id)

**TABLE B-241** groupmgrp

Event Name	Program	Event ID	Event Class	Mask
AUE_gm_add_grp	/opt/SUNWadm/2.3/ bin/groupmgrp	9307	ao	0x00080000
AUE_gm_del_grp		9308	ao	0x00080000
AUE_gm_mod_grp		9309	ao	0x00080000

Format:  
*header-token*  
*text-token* (group info)  
*text-token* (error message)  
*return-token*  
*subject-token*  
*slabel-token*

TABLE B-242 halt(1M)

Event Name	Program	Event ID	Event Class	Mask
AUE_halt_solaris	/usr/sbin/halt	6160	ss	0x00010000
Format: <i>header-token</i> <i>subject-token</i> <i>return-token</i>				

TABLE B-243 hostmgr

Event Name	Program	Event ID	Event Class	Mask
AUE_hm_add_host	/opt/SUNWadm/2.3/bin/hostmgr	9310	ao	0x00080000
AUE_hm_del_host		9311		
AUE_hm_mod_host		9312		
AUE_hm_set_def		9313		
Format: <i>header-token</i> <i>text-token</i> (host info) <i>text-token</i> (error message) <i>return-token</i> <i>subject-token</i> <i>slabel-token</i>				

TABLE B-244 inetd(1M)

Event Name	Program	Event ID	Event Class	Mask
AUE_inetd_connect	/usr/sbin/inetd	6151	na	0x00000400
Format: <i>header-token</i> <i>subject-token</i> <i>text-token</i> (service name) <i>ip-address-token</i> <i>ip-port-token</i> <i>return-token</i>				

TABLE B-245 in.ftpd(1M) - ftp access

Event Name	Program	Event ID	Event Class	Mask
AUE_ftpd	/usr/sbin/in.ftpd	6165	lo	0x00001000
Format: <i>header-token</i> <i>subject-token</i> <i>text-token</i> (error message, failure only) <i>return-token</i>				

TABLE B-246 installf(1M)

Event Name	Program	Event ID	Event Class	Mask
AUE_installf	/usr/sbin/installf	9042	as	0x00020000
Format: <i>header-token</i> <i>return-token</i> <i>argument-token</i> (package name) <i>subject-token</i> <i>slabel-token</i>				

**TABLE B-246** installf(1M) (continued)

**TABLE B-247** login(1) — local

Event Name	Program	Event ID	Event Class	Mask
AUE_login	/usr/bin/login	6152	10	0x00001000
Format: <i>header-token</i> <i>text-token</i> <i>text-token</i> (message - success or failure) <i>subject-token</i> <i>return-token</i>				

**TABLE B-248** login(1) — rlogin

Event Name	Program	Event ID	Event Class	Mask
AUE_rlogin	/usr/bin/login	6155	10	0x00001000
Format: <i>header-token</i> <i>subject-token</i> <i>text-token</i> (error message) <i>return-token</i>				

**TABLE B-249** login(1) — telnet

Event Name	Program	Event ID	Event Class	Mask
AUE_telnet	/usr/bin/login	6154	lo	0x00001000
Format: <i>header-token</i> <i>subject-token</i> <i>text-token</i> (error message) <i>return-token</i>				

**TABLE B-250** logout(1)

Event Name	Program	Event ID	Event Class	Mask
AUE_logout	/usr/bin/login	6153	lo	0x00001000
Format: <i>header-token</i> <i>subject-token</i> <i>text-token</i> <i>return-token</i>				

**TABLE B-251** lpadmin(1M) - authorization

Event Name	Program	Event ID	Event Class	Mask
AUE_uauth	/usr/lib/lpadmin	9017	ao	0x00080000
Format: <i>header-token</i> <i>text-token</i> (authorization used) <i>return-token</i> <i>text-token</i> (lpadmin command line) <i>subject-token</i> <i>slabel-token</i> <i>host-token</i>				



**TABLE B-251** lpadmin(1M) - authorization (continued)

**TABLE B-252** lpsched(1M) - authorization

Event Name	Program	Event ID	Event Class	Mask
AUE_uauth	/usr/lib/lpsched	9017	ao	0x00080000
Format: <i>header-token</i> <i>text-token</i> (" print without banners   print without labels   print a PostScript file") <i>return-token</i> <i>text-token</i> (hostname/jobnumber-filenumber) <i>slabel-token</i> (label of print job) <i>subject-token</i> <i>slabel-token</i> <i>host-token</i>				

**TABLE B-253** lpsched(1M) - privilege

Event Name	Program	Event ID	Event Class	Mask
AUE_lp_cancel	/usr/lib/lpsched	9044	ao	0x00080000
AUE_lp_status		9045		
Format: <i>header-token</i> <i>return-token</i> <i>privilege-token</i> <i>text-token</i> (hostname/jobnumber-filenumber) <i>slabel-token</i> (print job label) <i>subject-token</i> <i>slabel-token</i> <i>host-token</i> (error message)				

**TABLE B-254** modload(1M), modunload(1M)

Event Name	Program	Event ID	Event Class	Mask
AUE_modload	/usr/sbin/modload	9020	as	0x00020000
AUE_modunload	/usr/sbin/modunload	9021		

Format:  
*header-token*  
*subject-token*  
*groups-token*  
*slabel-token*  
*return-token*  
*exec\_args-token* (command-line arguments)  
*text-token* (module pathname)

**TABLE B-255** mountd(1M) – NFS mount

Event Name	Program	Event ID	Event Class	Mask
AUE_mountd_mount	/usr/lib/nfs/mountd	6156	na	0x00000400

Format:  
*header-token*  
*argument-token*  
*slabel-token* (subject slabel)  
*text-token* (remote client hostname)  
*path-token* (mount dir)  
*slabel-token* (slabel of the directory)  
*text-token* (error message, failure only)  
*attribute-token*  
*subject-token*  
*return-token*

**TABLE B-256** mountd(1M) – NFS unmount

Event Name	Program	Event ID	Event Class	Mask
AUE_mountd_umount	/usr/lib/nfs/mountd	6157	na	0x00000400
Format: <i>header-token</i> <i>slabel-token</i> (subject slabel) <i>text-token</i> (remote client hostname) <i>path-token</i> (mount dir) <i>slabel-token</i> (slabel of the directory) <i>text-token</i> (error message, failure only) <i>attribute-token</i> <i>subject-token</i> <i>return-token</i>				

**TABLE B-257** passwd(1)

Event Name	Program	Event ID	Event Class	Mask
AUE_passwd_eod	/usr/bin/passwd	6163	lo	0x00001000
Format: <i>header-token</i> <i>subject-token</i> <i>text-token</i> (error message) <i>return-token</i>				

TABLE B-258 pbind(1M)

Event Name	Program	Event ID	Event Class	Mask
AUE_pbind	/usr/sbin/pbind	9026	as	0x00020000

Format:  
*header-token*  
*subject-token*  
*groups-token*  
*slabel-token*  
*return-token*  
*exec\_args-token* (command-line arguments)  
*text-token* (action: "BIND" | "UNBIND")  
*arg-token* (1, "CPU", processor id)  
*arg-token* (2, "PID", process-id)

TABLE B-259 pfsh(1M)

Event Names	Program	Event IDs	Event Class	Mask
AUE_pfsh_trusted_priv	/usr/bin/pfsh	9007	ao	0x00080000
AUE_pfsh_trusted_nopriv		9008		
AUE_pfsh_priv		9009		
AUE_pfsh_nopriv	/usr/bin/pfsh	9010	ap	0x00004000

Format:  
*header-token*  
*path-token* (of the executable)  
*exec\_args-token*  
*path-token* (of current directory)  
*privilege-token*  
*return-token*  
*exec\_env-token* (if AUDIT\_ARGE is on)  
*subject-token*  
*slabel-token*

TABLE B-260 pkgadd(1M)

Event Name	Program	Event ID	Event Class	Mask
AUE_pkginstall	/usr/sbin/pkgadd	9040	as	0x00020000
Format: <i>header-token</i> <i>return-token</i> <i>argument-token</i> (package name) <i>subject-token</i> <i>slabel-token</i>				

TABLE B-261 pkgrm(1M)

Event Name	Program	Event ID	Event Class	Mask
AUE_pkgremove	/usr/sbin/pkgrm	9041	as	0x00020000
Format: <i>header-token</i> <i>return-token</i> <i>argument-token</i> (package name) <i>subject-token</i> <i>slabel-token</i>				

TABLE B-262 printmgr

Event Name	Program	Event ID	Event Class	Mask
AUE_pm_add_prn	/opt/SUNWadm/2.3/bin/printmgr	9316	ao	0x00080000
AUE_pm_del_prn		9318	ao	0x00080000

**TABLE B-262** printmgr (continued)

Event Name	Program	Event ID	Event Class	Mask
AUE_pm_mod_prn		9317	ao	0x00080000
Format: <i>header-token</i> <i>text-token</i> (printer info) <i>text-token</i> (error message) <i>return-token</i> <i>subject-token</i> <i>slabel-token</i>				

**TABLE B-263** profmgr - add profile

Event Name	Program	Event ID	Event Class	Mask
AUE_pm_add_prof	/opt/SUNWadm/2.3/ bin/profmgr	9306	ao	0x00080000
Format: <i>header-token</i> <i>text-token</i> (new profile info) <i>text-token</i> (error message) <i>return-token</i> <i>subject-token</i> <i>slabel-token</i>				

TABLE B-264 profmgr - delete profile

Event Name	Program	Event ID	Event Class	Mask
AUE_pm_del_prof	/opt/SUNWadm/2.3/ bin/profmgr	9304	ao	0x00080000
Format: <i>header-token</i> <i>text-token</i> (profile info) <i>text-token</i> (error message) <i>return-token</i> <i>subject-token</i> <i>slabel-token</i>				

TABLE B-265 profmgr - modify profile

Event Name	Program	Event ID	Event Class	Mask
AUE_pm_mod_prof	/opt/SUNWadm/2.3/ bin/profmgr	9305	ao	0x00080000
Format: <i>header-token</i> <i>text-token</i> (old profile info) <i>text-token</i> (new profile info) <i>text-token</i> (error message) <i>return-token</i> <i>subject-token</i> <i>slabel-token</i>				

TABLE B-266 psradm(1m)

Event Name	Program	Event ID	Event Class	Mask
AUE_psradm	/usr/sbin/psradm	9027	ps	0x00100000
Format: <i>header-token</i> <i>subject-token</i> <i>groups-token</i> <i>slabel-token</i> <i>return-token</i> <i>exec_args-token</i> (command-line arguments) <i>text-token</i> (action: "ON"   "OFF") <i>arg-token</i> (1, "PID", processor id)				

TABLE B-267 reboot(1M)

Event Name	Program	Event ID	Event Class	Mask
AUE_reboot_solaris	/usr/sbin/reboot	6161	ss	0x00010000
Format: <i>header-token</i> <i>subject-token</i> <i>return-token</i>				

TABLE B-268 removef(1M)

Event Name	Program	Event ID	Event Class	Mask
AUE_removef	/usr/sbin/removef	9043	as	0x00020000
Format: <i>header-token</i> <i>return-token</i> <i>argument-token</i> (package name) <i>subject-token</i> <i>slabel-token</i>				



TABLE B-268 `removef(1M)` (continued)TABLE B-269 `rpc.rexd(1M)`

Event Name	Program	Event ID	Event Class	Mask
AUE_rexd	/usr/sbin/rpc.rexd	6164	10	0x00001000
Format: <i>header-token</i> <i>subject-token</i> <i>text-token</i> (error message, failure only) <i>text-token</i> (hostname) <i>text-token</i> (username) <i>text-token</i> (command to be executed) <i>exit-token</i>				

TABLE B-270 `in.rexecd(1M)`

Event Name	Program	Event ID	Event Class	Mask
AUE_rexecd	/usr/sbin/in.rexecd	6162	10	0x00001000
Format: <i>header-token</i> <i>subject-token</i> <i>text-token</i> (error message, failure only) <i>text-token</i> (hostname) <i>text-token</i> (username) <i>text-token</i> (command to be executed) <i>exit-token</i>				

**TABLE B-271** in.rshd(1M) - rsh access

Event Name	Program	Event ID	Event Class	Mask
AUE_rshd	/usr/sbin/in.rshd	6158	lo	0x00001000
Format: <i>header-token</i> <i>subject-token</i> <i>text-token</i> (command string) <i>text-token</i> (local user) <i>text-token</i> (remote user) <i>return-token</i>				

**TABLE B-272** rem\_drv(1M)

Event Name	Program	Event ID	Event Class	Mask
AUE_rem_drv	/usr/sbin/rem_drv	9019	as	0x00020000
Format: <i>header-token</i> <i>subject-token</i> <i>groups-token</i> <i>slabel-token</i> <i>return-token</i> <i>exec_args-token</i> (command-line arguments) <i>text-token</i> (driver name) <i>[text-token]</i> (base directory)				

**TABLE B-273**    `init(1M)` - run level change

Event Name	Program	Event ID	Event Class	Mask
AUE_run_level_change	/usr/sbin/init	9024	ss	0x00010000
Format: <i>header-token</i> <i>text-token</i> (new run level) <i>subject-token</i> <i>slabel-token</i> (if slabel policy on) <i>return-token</i>				

**TABLE B-274**    Selection Manager Transfer

Event Name	Program	Event ID	Event Class	Mask
AUE_sel_mgr_xfer		9039	ax	0x00002000
Format: <i>header-token</i> <i>subject-token</i> <i>slabel-token</i> <i>return-token</i>				

**TABLE B-275** sendmail(1M)

Event Name	Program	Event ID	Event Class	Mask
AUE_sendmail_deliver	/usr/lib/sendmail	9013	ao	0x00080000
AUE_sendmail_defer		9014		
Format: <i>header-token</i> <i>text-token</i> (message about status) <i>text-token</i> (to) <i>text-token</i> (message ID) <i>text-token</i> (from) <i>text-token</i> (from host) <i>text-token</i> (to user) <i>text-token</i> (to host) <i>return-token</i> <i>slabel-token</i>				

**TABLE B-276** sendmail(1M) - upgrade

Event Name	Program	Event ID	Event Class	Mask
AUE_sendmail_upgrade	/usr/lib/sendmail	9012	ao	0x00080000
Format: <i>header-token</i> <i>text-token</i> (message ID) <i>slabel-token</i> (old label) <i>slabel-token</i> (new label) <i>subject-token</i> <i>slabel-token</i>				

TABLE B-277 serialmgr

Event Name	Program	Event ID	Event Class	Mask
AUE_sm_del_ser	/opt/SUNWadm/2.3/bin/serialmgr	9315	ao	0x00080000
AUE_sm_mod_ser		9314		
Format: <i>header-token</i> <i>text-token</i> (port info) <i>text-token</i> (error message) <i>return-token</i> <i>subject-token</i> <i>slabel-token</i>				

TABLE B-278 setuname(1M)

Event Name	Program	Event ID	Event Class	Mask
AUE_setuname	/usr/bin/setuname	9022	as	0x00020000
Format: <i>header-token</i> <i>subject-token</i> <i>groups-token</i> <i>slabel-token</i> <i>return-token</i> <i>exec_args-token</i> (command-line arguments) <i>text-token</i> (action: "ADD"   "DELETE") <i>path-token</i> (swapname)				

**TABLE B-279** share(1M)

Event Name	Program	Event ID	Event Class	Mask
AUE_EXPORTFS	/usr/lib/fs.d/nfs/ share	61	ao	0x00080000
Format: <i>header-token</i> <i>subject-token</i> <i>slabel-token</i> (subject slabel) <i>path-token</i> (export directory) <i>slabel-token</i> (slabel of the directory) <i>text-token</i> (export options) <i>return-token</i>				

**TABLE B-280** Workspace SL Change

Event Name	Program	Event ID	Event Class	Mask
AUE_sl_change		9035	ap	0x00004000
Format: <i>header-token</i> <i>subject-token</i> <i>slabel-token</i> (original SL) <i>slabel-token</i> (new SL) <i>return-token</i> <i>host-token</i>				

TABLE B-281 su(1M)

Event Name	Program	Event ID	Event Class	Mask
AUE_su	/usr/bin/su	6159	lo	0x00001000
Format: <i>header-token</i> <i>subject-token</i> <i>text-token</i> (error message) <i>return-token</i>				

TABLE B-282 swap(1M)

Event Name	Program	Event ID	Event Class	Mask
AUE_swap	/usr/sbin/swap	9030	as	0x00020000
Format: <i>header-token</i> <i>subject-token</i> <i>groups-token</i> <i>slabel-token</i> <i>return-token</i> <i>exec_args-token</i> <i>text-token</i> (new node name   <i>"*none*"</i> ) <i>text-token</i> (new systemname   <i>"*none*"</i> )				

**TABLE B-283** uadmin(1M)

Event Name	Program	Event ID	Event Class	Mask
AUE_uadmin_cmd	/usr/sbin/uadmin	9023	ss	0x00010000

Format:  
*header-token*  
*subject-token*  
*groups-token*  
*slabel-token*  
*return-token*  
*exec\_args-token* (command-line arguments)  
*argument-token* (1, "cmd", command code)  
*argument-token* (2, "fcn", function code)

**TABLE B-284** uauth

Event Name	Program	Event ID	Event Class	Mask
AUE_uauth	use of authorization	9017	ao	0x00080000

(See Table B-252 for use of authorization with printing)  
 Format:  
*header-token*  
*text-token* (user name)  
*text-token* (authorization)  
*subject-token*  
*return-token*

**TABLE B-285** usermgr

Event Name	Program	Event ID	Event Class	Mask
AUE_um_add_user	/opt/SUNWadm/2.3/bin/usermgr	9302	ao	0x00080000
AUE_um_del_user		9301		



TABLE B-285 usermgr (continued)

Event Name	Program	Event ID	Event Class	Mask
AUE_um_mod_user		9300		
AUE_um_set_def		9303		
Format: <i>header-token</i> <i>text-token</i> (user info) <i>text-token</i> (error message) <i>return-token</i> <i>subject-token</i> <i>slabel-token</i>				

TABLE B-286 uname(1)

Event Name	Program	Event ID	Event Class	Mask
AUE_uname_set	/usr/bin/uname	9024	as	0x00020000
Format: <i>header-token</i> <i>subject-token</i> <i>groups-token</i> <i>slabel-token</i> <i>return-token</i> <i>exec_args-token</i> (command-line arguments) <i>text-token</i> (new node name)				

**TABLE B-287** unshare(1M)

Event Name	Program	Event ID	Event Class	Mask
AUE_exportfs	/usr/lib/fs.d/nfs/ share		na	0x00000400
Format: <i>header-token</i> <i>subject-token</i> <i>slabel-token</i> (subject slabel) <i>path-token</i> (export directory) <i>return-token</i>				

## Audit Reference

---

Auditing brings a number of additional utilities to the Trusted Solaris operating environment. The utilities are listed here in four sections, each of which has a table below. Each table gives utility names and a short description of the task performed by each utility. The sections are identified by the man page suffix. The fifth table gives the filesystem security attributes of files in the auditing subsystem.

**TABLE C-1** Section 1M — Maintenance Commands

Command	Task
<code>audit(1M)</code>	Control the audit daemon
<code>audit_startup(1M)</code>	Initialize the audit subsystem
<code>audit_warn(1M)</code>	Run the audit daemon warning script
<code>auditconfig(1M)</code>	Configure auditing
<code>auditd(1M)</code>	Control audit trail files
<code>auditreduce(1M)</code>	Merge and select audit records from audit trail files

**TABLE C-1** Section 1M — Maintenance Commands *(continued)*

Command	Task
<code>auditstat(1M)</code>	Display kernel audit statistics
<code>praudit(1M)</code>	Print contents of an audit trail file
<code>/etc/init.d/audit stop</code>	Halt auditing [ a script; see <code>init.d(4)</code> ]
<code>/etc/init.d/audit start</code>	Restart auditing [ a script; see <code>init.d(4)</code> ]

**TABLE C-2** Section 2 — System Calls

System Call	System Parameter	Task
<code>audit(2)</code>		Write a record to the audit log
<code>auditon(2)</code>		Manipulate auditing:
	<code>A_GETPOLICY</code>	Get audit policy flags
	<code>A_SETPOLICY</code>	Set audit policy flags
	<code>A_GETKMASK</code>	Get asynchronous audit event preselection mask
	<code>A_SETKMASK</code>	Set asynchronous audit event preselection mask
	<code>A_GETQCTRL</code>	Get the kernel audit queue control parameters
	<code>A_SETQCTRL</code>	Set the kernel audit queue control parameters
	<code>A_GETSTAT</code>	Get the audit system statistics
	<code>A_SETSTAT</code>	Reset the audit system statistics

**TABLE C-2** Section 2 — System Calls *(continued)*

System Call	System Parameter	Task
	A_GETCOND	Determine if auditing is on/off/disabled
	A_SETCOND	Set auditing to on/off
	A_GETFSIZE	Get the size limit for an audit trail file
	A_GETCLASS	Return the event to class mapping for the designated event
	A_SETCLASS	Set the event to class mapping for the designated audit event
	A_GETPINFO	Get the audit information for the specified process
	A_SETPMASK	Set the preselection mask for a specified process
	A_SETUMASK	Set the process mask for all processes of a specified audit ID
	A_SETSMASK	Set the process mask for all processes of a specified session ID
	A_GETCWD	Get the current working directory for this process
	A_GETCAR	Get the current active root for this process
auditsvc(2)		Write audit log to specified file descriptor
getaudit(2)		Get process audit information
setaudit(2)		Set process audit information

**TABLE C-2** Section 2 — System Calls *(continued)*

System Call	System Parameter	Task
<code>getaudit(2)</code>		Get user audit identity
<code>setaudit(2)</code>		Set user audit identity

**TABLE C-3** Section 3 — C Library Functions

Library Call	Task
<code>au_preselect(3)</code>	Preselect an audit event
<code>au_user_mask(3)</code>	Get user's binary preselection mask
<code>getacdir(3), getacmin(3), getacflg(3), getacna(3), setac(3), endac(3)</code>	Get <code>audit_control(4)</code> file information
<code>getauclassnam(3), getauclassnam_r(3), getauclassent(3), getauclassent_r(3), setauclass(3), endauclass(3)</code>	Get <code>audit_class(4)</code> entries
<code>getauditflagsbin(3), getauditflagschar(3)</code>	Convert audit flag specifications
<code>getauevent(3), getauevent_r(3), getauevnam(3), getauevnam_r(3), getauevnum(3), getauevnum_r(3), getauevnonam(3), setauevent(3), endauevent(3)</code>	Get <code>audit_event(4)</code> entries

**TABLE C-3** Section 3 — C Library Functions *(continued)*

Library Call	Task
<code>getauusernam(3)</code> , <code>getauuserent(3)</code> , <code>setauuser(3)</code> , <code>endauser(3)</code>	Get <code>audit_user(4)</code> entries
<code>getfauditflags(3)</code>	Generate the process audit state

**TABLE C-4** Section 4 — Headers, Tables, and Macros

Files	Task
<code>audit.log(4)</code>	Gives format for an audit trail file
<code>audit_class(4)</code>	Gives audit class definitions
<code>audit_control(4)</code>	Controls information for system audit daemon
<code>audit_data(4)</code>	Holds current information on the audit daemon
<code>audit_event(4)</code>	Holds audit event definition and class mapping
<code>audit_user(4)</code>	Holds per-user auditing information

**TABLE C-5** Filesystem Security Attributes for the Audit Subsystem

<b>Name</b>	<b>[SL]</b>	<b>DAC</b>	<b>Owner</b>	<b>Group</b>
audit(1M)	[ADMIN_LOW]	555	bin	bin
auditd(1M)				
auditconfig(1M)				
auditstat(1M)				
auditreduce(1M)				
praudit(1M)				
/etc/init.d/audit*	[ADMIN_LOW]	400	root	sys
audit_warn(1M)	[ADMIN_LOW]	640	root	sys
audit_startup(1M)				
audit.log(4)	[ADMIN_HIGH]	400	root	root
audit_class(4)	[ADMIN_LOW]	400	root	sys
audit_control(4)	[ADMIN_LOW]	400	root	sys
audit_data(4)	ADMIN_LOW[ADMIN_LOW]	660	root	root



**TABLE C-5** Filesystem Security Attributes for the Audit Subsystem *(continued)*

<b>Name</b>	<b>[SL]</b>	<b>DAC</b>	<b>Owner</b>	<b>Group</b>
audit_event(4)	[ADMIN_LOW]	400	root	sys
audit_user(4)	[ADMIN_LOW]	400	root	sys



# Index

---

## Special Characters

- + audit flag prefix 22
- audit flag prefix 22
- ^+ audit flag prefix 23
- ^- audit flag prefix 23

## A

- aa audit class 108, 110
- aa audit flag 105
- access audit record 176
- acct audit record 176
- acl token 151
- ad audit flag 105
- add\_drv audit record 280
- adjtime audit record 177
- Admin Editor audit record 281
- administrative roles
  - assuming 52
- ahlt policy
  - flag 36
- aliases
  - creating audit\_warn mail alias 62
- all
  - audit flag 105
    - caution for using 22
  - in user audit fields 27
- allhard string with audit\_warn script 31, 32
- allocate audit record
  - deallocate device 289
  - deallocate device failure 289
  - device allocate failure 282
  - device allocate success 281

- list device failure 283
- list device success 282
- allsoft string with audit\_warn script 31
- always-audit flags
  - described 27
  - process preselection mask 28
- ao audit class 110, 114
- ao audit flag 105
- ap audit class 114
- ap audit flag 105
- arbitrary token 151
- arg token 153
- arge policy
  - exec\_env token and 155
- argv policy
  - exec\_args token and 155
- as audit class 134
- as audit flag 106
- at audit record
  - at-create crontab 283
  - at-delete atjob 283
  - at-permission 284
- attr token 153
- audit -n command 77
- audit -s command
  - preselection mask for existing processes 25
  - rereading audit files 30, 74
  - resetting directory pointer 75
- audit -t command 74
- audit attributes, *see* audit tokens
- audit audit record 177, 284
- audit classes

- adding 66
- changing definitions 18
- mapping events 18
- overview 18, 19
- selecting for auditing 18
- setting mappings for attributable events 103
- setting mappings for non-attributable events 103
- audit clients 43
- audit daemon
  - audit trail creation 29, 30, 78
  - audit\_startup file 17
  - audit\_warn script
    - conditions invoking 31, 32
    - described 29, 31
  - directories suitable to 30
  - enabling auditing 17
  - functions 29
  - order audit files are opened 25
  - rereading the audit\_control file 25
  - starting 74
  - starting manually 102
- audit directories
  - creating 56
  - mounting 58
- audit events
  - audited by default 17
  - audit\_event file
    - audit event type 79
  - categories 19
  - finding in audit trail 90
  - including in audit trail 18
  - kernel events
    - audit tokens 79
    - auditconfig command options 34
    - described 19
  - mapping to classes 18
  - non-attributable 20
  - numbers 19
  - numbers of system calls 19
  - overview 18, 20
  - pseudo-events 19
  - record formats and 79
  - user-level events
    - audit tokens 79
    - auditconfig command options 34
    - described 19
- audit files
  - /etc/security/audit\_class file 17
  - /etc/security/audit\_control file 17
  - /etc/security/audit\_event file 17
  - /etc/security/audit\_user file 17
  - /etc/security/audit\_warn file 17
  - backup 93
  - cleaning up not\_terminated file 99
  - combing selected ones 90
  - copying login/logout messages to single file 89
  - directory locations 23, 24, 77
  - displaying in entirety 88
  - managing 85
  - managing size of 30
  - merging 87
  - minimum free space for file systems 25
  - names
    - closed files 84
    - examples 84
    - form 83
    - still-active files 84
    - time stamps 83
    - use 83
  - nonactive files marked
    - not\_terminated 99
  - order for opening 25
  - overflow prevention 97
  - printing 88, 89
  - reading closed file 87
  - reading still-open file 87
  - reducing size 91
  - reducing storage space requirements 32, 33
  - restoring 94
  - specifying location 59
  - switching to new file 77
  - time stamps 83
- audit flags
  - audit\_control file line 25
  - audit\_user file 26, 27
  - changing dynamically 73
  - definitions 21
  - list of 105, 107
  - machine-wide 21, 25
  - overview 21
  - policy flags 35

- prefixes 22
  - process preselection mask 28
  - syntax 22
- audit IDs
  - acquired at login 28
  - ensuring successful tracking 17
  - example audit record 81
- audit log files, *see* audit files
- audit mappings 105
- audit partitions
  - creating 51
  - removing free space 53
- audit policies
  - determining 70
  - setting 35
  - setting temporarily 71
- audit records 147, 314
  - adding sequence token 101
  - audit directories full 30, 32
  - audit ID 81
  - audit session ID 81
  - converting to human-readable format 87, 92
  - displaying by designated dates 90
  - displaying user activities 88
  - features in audit trail 81, 82
  - format 79
  - format in audit trail 79, 80
  - format or structure 79, 82, 147, 176
  - human-readable format 80
  - kernel-level generated 176, 257
  - login record 81, 82
  - overview 20
  - policy flags 35
  - printing user activities 89
  - pseudo-events 257
  - reading 81
  - removing sequence token 101
  - selecting from audit trail 85
  - self-contained records 82
  - sending to a different file 75
  - time-stamp format 83
  - use of privilege 257
  - user-level generated 280, 314
- audit script 48, 50
- audit servers
  - mount-point path names 23
  - partitioning example 51
  - planning 43
- audit session ID 29, 81
- audit tokens
  - acl token 151
  - arbitrary token 151
  - arg token 153
  - attr token 153
  - audit record format 79, 82, 147, 148
  - described 20
  - examples 150, 175
  - clearance token 154
  - exec\_args token 154
  - exec\_env token 155
  - exit token 153, 155, 156
  - file token 156
  - groups token 157
  - header token 79 to 81, 157
  - host token 158
  - in\_addr token 159
  - ip token 159
  - ipc token 160
  - ipc\_perm token 161
  - ipport token 162
  - liaison token 162
  - newgroups token 162
  - opaque token 163
  - order 79
  - order in audit record 79
  - path token 164
  - policy flags 35
  - priv token 164
  - privilege token 165
  - process token 166
  - reading 80
  - return token 167
  - seq token 167
  - slabel token 168
  - socket token 168, 169
  - socket-inet token 169
  - subject token 169
  - table of 148
  - text token 170
  - trailer token 79, 171
  - types 79
  - xatom token 171
  - xclient token 172
  - xcolormap token 172

- xcursor token 173
- xfont token 173
- xgc token 173
- xpixmap token 174
- xproperty token 174
- xselect token 174
- xwindow token 175
- audit trail
  - analysis
    - auditing features 81, 82
    - auditreduce command 85, 90
    - costs 32
    - finding failed login attempts 103
    - of cost 32
    - praudit command 87, 92
  - analyzing 85
  - auditreduce command 85, 90
  - creating
    - audit daemon's role 29, 30, 78
    - audit\_data file 29
    - directory suitability 30
    - managing audit file size 30
    - overview 78
  - debugging 100
  - directory locations 23, 24, 77
  - events included 18
  - merging 85
  - monitoring in real time 33
  - overflow prevention 97
  - praudit command 87, 92
- auditconfig command
  - audit flags as arguments 34
  - changing class mappings 102
  - options 34, 35
  - prefixes for flags 23
- auditd daemon
  - audit trail creation 29, 30, 78
  - audit\_startup file 17
  - audit\_warn script
    - conditions invoking 31, 32
    - described 29, 31
    - execution of 29, 30
  - directories suitable to 30
  - enabling auditing 17
  - functions 29
  - order audit files are opened 25
  - rereading the audit\_control file 25
- auditing
  - advanced setup procedures 66, 70
  - advanced tasks for security administrator 47
  - audit ID 81
  - audit session ID 81
  - for efficiency 40
  - basic setup procedures 51, 66
  - basic tasks for security administrator 46
  - client-server relationships 48, 50
  - considerations 39
  - defaults 17
    - audit\_startup file 17
  - disabling 48, 50
  - dynamic procedures 70
  - enabling 17, 49, 50
  - overview of administration 15, 17
  - planning 38, 45
  - removing free space 53
  - setup tasks for system administrator 45
  - shutdown 47
  - site planning 37
  - software packages 15
  - space planning 41, 42
  - startup 17, 47
  - user ID 81
  - warning of trouble 62
- auditon audit record
  - A\_GETCAR command 178
  - A\_GETCLASS command 178
  - A\_GETCOND command 178
  - A\_GETCWD command 178
  - A\_GETKMASK command 179
  - A\_GETSTAT command 179
  - A\_GPOLICY command 179
  - A\_GQCTRL command 180
  - A\_SETCLASS command 180
  - A\_SETCOND command 180
  - A\_SETKMASK command 181
  - A\_SETSMASK command 181
  - A\_SETSTAT command 182
  - A\_SETUMASK command 182
  - A\_SPOLICY command 182
  - A\_SQCTRL command 183
- auditpsa audit record 183
- auditreduce command
  - capabilities 85
  - cleaning not\_terminated files 100

- described 85
- distributed systems 85
- examples 86, 90, 100
- time stamp use 83
- auditstat audit record 183
- auditsvc
  - system call fails 32
- auditsvc audit record 184
- auditwrite audit record 285
- audit\_control file
  - audit daemon rereading after editing 25
  - audit\_user file modification 26
  - dir: line
    - examples 26
    - files 24
  - dir: line described 25
  - examples 26
  - flags: line
    - described 25
    - prefixes in 23
    - process preselection mask 28
  - minfree: line
    - audit\_warn condition 31
    - described 25
  - naflags: line 25
  - overview 17, 20
  - prefixes in flags line 23
  - problem with contents 32
- audit\_data file 29
- audit\_event file
  - overview 18, 20
- audit\_startup file 17
- audit\_user file
  - prefixes for flags 23
  - process preselection mask 28
  - user audit fields 26, 27
- audit\_warn script 31, 32
  - allhard string 31, 32
  - allsoft string 31
  - auditsvc string 32
  - conditions invoking 31, 32
  - described 29, 31
  - ebusy string 32
  - hard string 31
  - postsigterm signal 32
  - soft string 31
  - tmpfile string 32
- AUE\_... names 19

- authorization use audit record 312
- ax audit class 137
- ax audit flag 106

## B

- backup
  - audit files 93
- binary audit record format 79

## C

- caret (^) in audit flag prefixes 23
- change password audit record 299
- change workspace SL audit record 310
- chdir audit record 184
- chmod audit record 185
- chown audit record 185
- chroot audit record 186, 286
- chstate audit record 186
- cl audit class 114
- cl audit flag 106
- classes
  - changing definitions 18
  - flags and definitions 21
  - mapping events 18
  - overview 18, 20
  - selecting for auditing 18
- clearance token 154
- clock\_settime audit record 187
- close audit record 187
- cnt policy
  - flag 36
- commands
  - executing with privilege 52
- configuration files
  - distributing to workstations 63
- core dump audit record 224
- cost control 32, 33
  - analysis 32
  - processing time 32
  - storage 32, 33
- creat audit record 188
- creating the audit trail
  - audit daemon's role 29, 30
  - audit\_data file 29
  - directory suitability 30

- overview 78
- cron job 30
- crontab audit record
  - cron-invoke atjob 287
  - crontab 287
  - crontab-crontab created 286
  - crontab-crontab deleted 287
  - crontab-permission 288

**D**

- daemons, audit, *see* audit daemon
- date-time auditreduce command
  - options 90
- dbmgr audit records 288
- defaults
  - audit policies 35
  - machine-wide 21
- Device Allocation Manager
  - using 65
- devices
  - allocating and deallocating 65
- dir: line in audit\_control file
  - described 25
  - example 26
  - for files 24
  - for files subdirectory 24
- directories
  - audit daemon pointer 30
  - audit directories full 30 to 32
  - audit directory locations 23, 24, 77
  - audit\_control file definitions 25
  - diskfull machines 52
  - files subdirectory 56
  - files subdirectory for audit records 23
  - mounting audit directories 57
  - suitable to audit daemon 30
- disk space requirements 32, 33
- diskfull machines' audit directory 51
- dispadmin audit record 290
- distributed systems' auditreduce command
  - use 85
- drvpolicy audit record 188
- dtfile audit records 290

## E

- ebusy string and audit\_warn script 32

- eeprom audit record 291
- ending
  - signal received during auditing
    - shutdown 32
  - terminating audit daemon 48, 74
- enter prom audit record 189
- errors
  - audit directories full 30 to 32
  - internal errors 32
- /etc/init.d/audit script
  - disabling auditing 48, 49, 51
- /etc/security directory 23
- /etc/security/audit directory 23, 77
- /etc/security/audit\_data file 29
- /etc/security/audit\_event file
  - overview 18, 20
  - audit event type 79
- /etc/security/audit\_startup file 17
- /etc/security/audit\_warn script 29, 31, 32
- event numbers 19
- event-class mappings
  - changing 41
  - planning 40
- events
  - categories 19
  - including in audit trail 18
  - kernel events
    - audit tokens 79
    - auditconfig command options 34
    - described 19
  - mapping to classes 18
  - non-attributable 20
  - numbers 19
  - overview 18, 20
  - record formats and 79
  - user-level events
    - audit tokens 79
    - auditconfig command options 34
    - described 19
- exec audit record 189
- execve audit record 189
- exec\_args token 154, 155
- exec\_env token 155
- exit audit record 190
- exit prom audit record 189
- exit token 155, 156



## F

- fa audit class 115
- fa audit flag 106
- failure
  - audit flag prefix 22
  - turning off audit flags for 23
- fauditpsa audit record 190
- fc audit class 117
- fc audit flag 106
- fchdir audit record 191
- fchmod audit record 191
- fchown audit record 192
- fchroot audit record 193
- fcntl audit record 193
- fd audit class 118
- fd audit flag 106
- fgetcmwlabel audit record 197
- fgetslname audit record 194
- file systems
  - see also* audit files
  - free space on audit servers 25
- file token 156
- files
  - setting public object bit 69
- files and file systems
  - handling audit trail overflow 99
  - protecting 54
- files subdirectory 23, 56
- files, audit, *see* audit files
- flags
  - audit\_control file line 25
  - audit\_user file 26, 27
  - definitions 21
  - machine-wide 21, 25
  - overview 21
  - policy flags 35
  - process preselection mask 28
  - syntax 22
- flags: line in audit\_control file
  - described 25
  - prefixes in 23
  - process preselection mask 28
- fm audit class 115, 117 to 119, 122, 124, 126, 128, 130, 131
- fm audit flag 106
- fn audit class 121
- fn audit flag 106

- fork audit record 194
- fork1 audit record 194
- fr audit class 121
- fr audit flag 106
- freeze audit record 250
- fsetcmwlabel audit record 195
- fsetattrflag audit record 195
- fstatfs audit record 196
- ftpd login audit record 294
- fuser audit record 291
- fw audit class 122
- fw audit flag 106

## G

- getaudit audit record 196
- getauuid audit record 197
- getcmwfsrange audit record 197
- getcmwlabel audit record 197
- getdents audit record 198
- getfilepriv audit record 198
- getmldadorn audit record 199
- getmsg audit record 199
  - socket accept 200
  - socket receive 200
- getmsgqcmwlabel audit record 200
- getpmsg audit record 201
- getportaudit audit record 201
- getsemcmwlabel audit record 202
- getshmcmwlabel audit record 202
- getslname audit record 203
- group token 157
- groupmgr audit record 292
- groups token 157

## H

- halt audit record 292
- hard disk space requirements 32, 33
- hard string with audit\_warn script 31
- header token
  - described 80, 82, 157
  - fields 80, 82
  - format 158
  - order in audit record 79
  - praudit display 80, 158
- host token 158, 159

- hostmgr audit records 293
- human-readable audit record format
  - converting audit records to 87, 92
  - described 79

## I

- icons
  - for device allocation 65
- IDs
  - audit 17, 28, 81
  - audit session 29, 81
  - audit user 81
  - terminal 29
- in.ftpd audit record 294
- in.rexecd audit record 305
- in.rshd audit record 305
- inetd: service request audit record 293
- init audit record
  - run level change information 306
- installf audit record 294
- Internet-related tokens
  - socket token 169
- in\_addr token 159
- io audit class 123, 126
- io audit flag 106
- ioctl: ioctl to special devices audit record 203
- ip audit class 124
- ip audit flag 106
- ip token 159
- ipc token 160
- ipc type field values (ipc token) 160
- ipc\_perm token 161
- ipport token 162
- item size field values (arbitrary token) 152

## K

- kernel events
  - audit number range 19
  - audit records 176, 257
  - audit tokens 79
  - auditconfig command options 34
  - described 19
- kill audit record 204

## L

- lchown audit record 205

- lgetcmwlabel audit record 197
- liaison token 162
- link audit record 205
- lo audit flag 106
- log files
  - see also* audit files
- login audit record
  - logout 296
  - praudit display 81
  - rlogin 295
  - telnet login 295
  - terminal login 295
- login/logout messages, copying to single file 89
- lpsched audit records 297
- lsetcmwlabel audit record 236
- lstat audit record 206
- lxstat audit record 206

## M

- machine halt audit record 292
- machine reboot audit record 304
- mail
  - creating audit\_warn alias 62
- mask, process preselection
  - described 28
  - reducing storage costs 33
- memcntl audit record 207
- minfree: line in audit\_control file 31
  - audit\_warn 31
  - described 25
- minus (-) audit flag prefix 22, 167
- mkdir audit record 207
- mknod audit record 208
- mldsetfattrflag audit record 208
- mmap audit record 209
- modctl audit record
  - MODADDMAJBIND command 209
  - MODCONFIG command 210
  - MODLOAD command 210
  - MODUNLOAD command 211
- modify system files audit record 281
- modload audit record 297
- modunload audit record 297
- monitoring audit trail in real time 33
- mount audit record 211

- mountd audit record 298
- msgctl audit record
  - IPC\_RMID command 212
  - IPC\_SET command 213
  - IPC\_STAT command 213
- msgget audit record 214
- msggetl audit record 214
- msgrcv audit record 214
- msgsnd audit record 215
- munmap audit record 215

## N

- na audit class 127
- na audit flag 106
- naflags: line in audit\_control file 25
- names
  - audit classes 21
  - audit files
    - closed files 84
    - form 83
    - still-active files 84
    - time stamps 83
    - use 83
  - audit flags 21
  - audited kernel events 19
  - IDs
    - audit 17, 28
    - audit session 29, 81
    - terminal 29
  - mount-point path names on audit servers 23
- networked workstations
  - planning space 42
- never-audit flags 27
- newgroups token 162
- NFS request audit record
  - automount 285
  - mount 211, 298
  - mountd 298
  - unmount 298
- nice audit record 216
- no audit class 128
- no audit flag 107
- non-networked workstations
  - planning space 42
- nonattributable flags in audit\_control file 25
- nt audit class 130

- nt audit flag 107
- numbers
  - audit events 19

## O

- opaque token 163
- open audit record
  - read 216
  - read, create 217
  - read, create, truncate 217
  - read, truncate 217
  - read, write 218
  - read, write, create 218
  - read, write, create, truncate 219
  - read, write, truncate 219
  - write 220
  - write, create 220
  - write, create, truncate 221
  - write, truncate 221
- /opt/SUNWadm/2.3/bin/dbmgrp audit record 288
- /opt/SUNWadm/2.3/bin/groupmgr audit record 292
- /opt/SUNWadm/2.3/bin/hostmgr audit records 293
- /opt/SUNWadm/2.3/bin/printmgr audit records 301
- /opt/SUNWadm/2.3/bin/profmgr audit records 302
- /opt/SUNWadm/2.3/bin/profmgr audit records 302, 303
- /opt/SUNWadm/2.3/bin/serialmgr audit records 308
- /opt/SUNWadm/2.3/bin/usermgr audit records 312
- ot audit class 131
- ot audit flag 107

## P

- passwd audit record 299
- path policy flag 36
- path token 164
- pathconf audit record 222
- pbind audit record 299
- pc audit flag 107

- pfsh audit record 300
- pipe audit record 222
- pkgadd audit record 300
- pkgrm audit record 301
- pm audit flag 107
- postsigterm string and audit\_warn script 32
- praudit command
  - changing field separator 92
  - changing token separator 92
  - human-readable format 80
  - output formats 87, 92
  - piping auditreduce output to 87, 88
  - using 87, 92
- preadl audit record 223
- preselection mask
  - described 28
  - reducing storage costs 33
- primary audit directory 25
- print format field values (arbitrary token) 152
- printing with authorization audit record 297
- printmgr audit record 301
- priocntl audit record 223
- priv token 164
- privenable audit record 224
- privilege
  - audit records for use of 257
  - when executing commands 52
- privilege token 165
- privilege use audit record 257
- process audit characteristics 28, 29
  - audit ID 28
  - audit session ID 29
  - process preselection mask 28, 33
  - terminal ID 29
- process dumped core audit record 224
- process preselection mask
  - described 28
  - reducing storage costs 33, 34
- process token 166
- processing time costs 32
- profile shell
  - audit record 300
  - opening 53
- profmgr audit records 302, 303
- ps audit class 133
- ps audit flag 107
- pseudo-events
  - audit records 257

- psradm audit record 303
- putmsg audit record 225
  - socket connect 225
- putpmsg audit record 225
- pwritel audit record 255

## R

- read audit record 226
- readl audit record 226
- readlink audit record 226
- readvl audit record 226
- reboot audit record 304
- records, *see* audit records
- removef audit record 304
- rem\_drv audit record 306
- rename audit record 227
- restoration
  - audit files 94
- return token 167
- rmdir audit record 227
- rpc.rexd audit record 305
- rsh access audit record 305

## S

- secondary audit directory 25
- security
  - auditing tasks for administrators 46
- selection manager audit record 307
- semctl audit record
  - GETALL command 228
  - GETNCNT command 228
  - GETPID command 229
  - GETVAL command 229
  - GETZCNT command 230
  - IPC\_RMID command 230
  - IPC\_SET command 231
  - IPC\_STAT command 232
  - SETALL command 231
  - SETVAL command 232
- semget audit record 233
- semgetl audit record 233
- semop audit record 234
- sendmail audit record 307, 308
- seq policy flag 37
- seq token 167

- serialmgr audit records 308
- session ID 29, 81
- setacl audit record 234
- setaudit audit record 235
- setaudit audit record 235
- setclearance audit record 236
- setcmwlabel audit record 236
- setcmwplabel audit record 237
- setegid audit record 238
- seteuid audit record 239
- setfacl audit record 234
- setfattrflag audit record 239
- setfpriv audit record 240
- setgid audit record 238
- setgroups audit record 240
- setpattr audit record 241
- setpgrp audit record 241
- setppriv audit record 242
- setregid audit record 242
- setreuid audit record 242
- setrlimit audit record 243
- setuid audit record 243
- setuname audit record 309
- share audit record 309
- shmat audit record 244
- shmctl audit record
  - IPC\_RMID command 244
  - IPC\_SET command 245
  - IPC\_STAT command 245
- shmdt audit record 246
- shmget audit record 246
- shmgetl audit record 247
- signal received during auditing shutdown 32
- size
  - managing audit files 30
  - reducing audit files
    - storage space requirements 32, 33
- slabel token 168
- socket accept audit record 200
- socket connect audit record 225
- socket send audit record 225
- socket token 168
- socket-inet token 169
- soft limit
  - audit\_warn condition 31
  - minfree: line described 25
- soft string with audit\_warn script 31
- ss audit class 136
- ss audit flag 107
- stat audit record 247
- statfs audit record 247
- statvfs audit record 247
- stime audit record 248
- storage costs 32, 33
- su audit record 310
- subject token 169
- success
  - audit flag prefix 22
  - turning off audit flags for 23
- SUNWcar package 16
- SUNWcsr package 16
- SUNWcsu package 16
- SUNWhea package 16
- SUNWman package 16
- SUNWtsr package 16
- SUNWtsu package 16
- swap audit record 311
- symlink audit record 248
- sysinfo audit record 249
- system booted audit record 249
- system calls
  - audit event numbers 19
  - auditsvc fails 32
  - return token 167
- system rebooted audit record 251
- system remounted audit record 251
- system shutdown audit record 251
- System V IPC
  - ipc token 160

## T

- temporary file cannot be used 32
- terminal ID 29
- terminating
  - signal received during auditing shutdown 32
- text token 170
- time stamps in audit files 83
- time-date auditreduce command options 90
- tmpfile string and audit\_warn script 32
- tnif audit record 250
- tnrh audit record 250
- tnrhtp audit record 250
- tokens, *see* audit tokens

- tokmapper audit record 250
- trail, *see* audit trail
- trailer token
  - format 171
  - order in audit record 79
  - praudit display 171
- troubleshoot 97, 105
- trusted editor audit record 281

## U

- uadmin audit record
  - system freeze 250
  - system reboot 251
  - system remount 251
  - system shutdown 251
  - user-level command 311
- uauth audit record 297, 312
- uauth printing audit records 296
- UIDs
  - user ID (audit ID) 17, 28, 81
- umount: old version audit record 252
- uname audit record 313
- unlink audit record 252
- unshare audit record 313
- user audit fields 26, 27
- user ID (audit ID) 17
- user-level events
  - audit records 280, 314
  - audit tokens 79
  - auditconfig command options 34
  - described 19
- usermgr audit records 312
- users
  - auditing normal users 26
- /usr/bin/at audit record
  - at-create crontab 283, 284
- /usr/bin/crontab audit record
  - crontab-crontab created 286, 287
  - crontab-permission 288
  - cron-invoke atjob 287
  - crontab 287
- /usr/bin/login audit record
  - rlogin 295, 296
  - terminal login 295
- /usr/bin/passwd: change password audit
  - record 299
- /usr/bin/pfsh audit record 300

- /usr/bin/setuname audit record 309
- /usr/bin/su audit record 310
- /usr/bin/uname audit record 313
- /usr/dt/bin/dtfile audit record 290
- /usr/lib/lpadmin audit record 296
- /usr/lib/lpsched audit records 297
- /usr/lib/nfs/mountd audit record
  - NFS mount request 298
  - NFS unmount request 298
  - automount mismatch 285
  - automount request 285
- /usr/lib/sendmail audit record 307, 308
- /usr/sbin/add\_drv audit record 280
- /usr/sbin/allocate audit record
  - allocate-list device failure 283
  - allocate-list device success 282
  - device allocate failure 282
  - device allocate success 281
- /usr/sbin/audit audit record 284
- /usr/sbin/chroot audit record 286
- /usr/sbin/deallocate audit record
  - deallocate device failure 289
  - deallocate device 289
- /usr/sbin/eeprom audit record 291
- /usr/sbin/fuser audit record 291
- /usr/sbin/halt audit record 292
- /usr/sbin/in.ftpd audit record 294
- /usr/sbin/in.rexecd audit record 305
- /usr/sbin/in.rshd audit record 305
- /usr/sbin/inetd audit record 293
- /usr/sbin/init audit record 306
- /usr/sbin/installf audit record 294
- /usr/sbin/modload audit record 297
- /usr/sbin/modunload audit record 297
- /usr/sbin/pbind audit record 299
- /usr/sbin/psradm audit record 303
- /usr/sbin/reboot audit record 304
- /usr/sbin/removef audit record 304
- /usr/sbin/rpc.rexd audit record 305
- /usr/sbin/swap audit record 311
- /usr/sbin/uadmin audit record 311
- utime audit record 253
- utimes audit record 253
- utssys - fusers audit record 253

## V

vfork audit record 254  
vtrace audit record 254

## W

workspaces  
    Change SL audit record 310  
    changing to admin\_high 71  
workstations  
    planning audit space 41  
write audit record 255  
writel audit record 255  
writevl audit record 255

## X

X server extensions audit record 279  
xa audit class 137  
xa audit flag 107  
XAllocColor audit record 272  
XAllocColorCells audit record 272  
XAllocColorPlanes audit record 272  
XAllocNamedColor audit record 272  
xatom token 171  
XBell audit record 276  
xc audit class 138  
xc audit flag 107  
XChangeActivePointerGrab audit record 262  
XChangeGC audit record 267  
XChangeHosts audit record 278  
XChangeKeyboardControl audit record 276  
XChangeKeyboardMapping audit record 276  
XChangePointerControl audit record 276  
XChangeProperty audit record 260  
XChangeSaveSet audit record 258  
XChangeWindowAttributes audit record 258  
XCirculateWindow audit record 258  
XClearArea audit record 269  
xclient token 172  
XClientConnect audit record 258  
XClientDisconnect audit record 258  
xcolormap token 172, 173  
XConfigureWindow audit record 258  
XConvertSelection audit record 261  
XCopyArea audit record 269  
XCopyColormapAndFree audit record 273  
XCopyGC audit record 268

XCOPYPlane audit record 269  
XCreateColormap audit record 272  
XCreateCursor audit record 274  
XCreateGlyphCursor audit record 275  
XCreateWindow audit record 258  
xcursor token 173  
XDeleteProperty audit record 260  
XDestroySubwindows audit record 258  
XDestroyWindow audit record 258  
XFillPolygon audit record 270  
xfont token 173  
XForceScreenSaver audit record 277  
XFreeColormap audit record 273  
XFreeColors audit record 272  
XFreeCursor audit record 275  
XFreeGC audit record 268  
XFreePixmap audit record 276  
xgc token 173  
XGetAtomName audit record 260  
XGetGeometry audit record 258  
XGetImage audit record 271  
XGetInputFocus audit record 266  
XGetMotionEvents audit record 265  
XGetProperty audit record 260  
XGetSelectionOwner audit record 261  
XGetWindowAttributes audit record 258  
XGrabButton audit record 261  
XGrabKey audit record 263  
XGrabKeyboard audit record 263  
XGrabPointer audit record 262  
XGrabServer audit record 264  
XImageText16 audit record 271  
XImageText8 audit record 271  
XInstallColormap audit record 273  
XInternAtom audit record 260  
XKillClient audit record 278  
xl audit class 139  
xl audit flag 107  
XListInstalledColormap audit record 273  
XListProperty audit record 260  
XLookupColor audit record 274  
XMapSubwindows audit record 258  
XMapWindow audit record 258  
xmknod audit record 256  
xp audit class 140  
xp audit flag 107  
xpixmap token 174

XPolyArc audit record	270	XSetClipRectangles audit record	268
XPolyFillArc audit record	270	XSetCloseDownMode audit record	278
XPolyFillRectangle audit record	270	XSetDashes audit record	268
XPolyLine audit record	270	XSetFontPath audit record	267
XPolyPoint audit record	270	XSetInputFocus audit record	266
XPolyRectangle audit record	270	XSetModifierMapping audit record	279
XPolySegment audit record	270	XSetPointerMapping audit record	279
XPolyText16 audit record	271	XSetScreenSaver audit record	277
XPolyText8 audit record	271	XSetSselectionOwner audit record	261
xproperty token	174	xstat audit record	256
XPutImage audit record	271	XStoreColors audit record	274
XQueryColors audit record	274	XStoreNamedColor audit record	274
XQueryKeymap audit record	266	XTranslateCoords audit record	265
XQueryPointer audit record	264	XUngrabButton audit record	262
XQueryTree audit record	258	XUngrabKey audit record	263
XRecolorCursor audit record	275	XUngrabKeyboard audit record	263
XReparentWindow audit record	258	XUnGrabPointer audit record	262
XRotateProperties audit record	278	XUngrabServer audit record	264
xs audit class	143	XUninstallColormap audit record	273
xs audit flag	107	XUnmapSubwindows audit record	258
xselect token	174	XUnmapWindow audit record	258
XSendEvent audit record	265	XWarpPointer audit record	265
Xserver protocols		xwindow token	175
audit records	257, 280	xx audit flag	107
XSetAccessControl audit record	278		