A

ab2admin — Command-Line Interface for AnswerBook2 Administration

Synopsis

```
/usr/lib/ab2/bin/ab2admin [-h][-H command][-o command [arguments]]
```

Description

Use the `ab2admin` command-line interface to administer AnswerBook2 collections and documents on a specified AnswerBook2 server. You can perform the following tasks with the `ab2admin` command.

- Scan for locally installed collections and update the server database.
- Obtain a listing of collections and books.
- Stop the server.
- Start the server.
- Restart the server.
- Turn on or off the server log files.
- Rotate the log files.
- Configure the server to resolve links to books located on other AnswerBook2 servers.
- Control server access by adding or deleting users to the pool of administrative users.
- Enable or disable access control.

ab2admin can connect to any AnswerBook2 server, either local or remote. Certain functions such as stop, start, and restart apply only to the local AnswerBook2 server. If the AnswerBook2 server is protected by a password, then you must provide a user ID and password to initiate an administration task.

To run ab2admin interactively, type ab2admin from the command line and then enter commands as prompted. You can also execute the command entirely from the command line with the -o option.

The ab2admin command is run automatically as part of the installation of the SUNWab2u AnswerBook2 package.

Options

-h Display help.

New! -H [*command*] Display help and help for a specified *command*.

-o *subcommand* [*arguments*]

The supported subcommands are listed below.

New! access_off [-m *server*][-p *server-port-number*]

Disable the server access log file.

access_on [-m *server*][-p *server-port-number*]

Enable the server access log file.

add_admin -u *user-id* [-m *server*][-p *server-port-number*]

Add user to the authorized list of server administrators.

add_coll -d *path* [-m *server*][-p *server-port-number*]

Add AnswerBook1 or AnswerBook2 collections into the specified AnswerBook2 server database.

add_server -M *alternate-server*
-P *alternate-server-port-number* [-m *server*]
[-p *server-port-number*]

Add alternate server to the specified server.

auth_off [-m *server*][-p *server-port-number*]

Disable the server administration verification.

auth_on [-m *server*][-p *server-port-number*]

Enable the server administration verification.

New! autostart_no [-m *server*][-p *server-port-number*]

Stop AnswerBook2 server from starting automatically when system is (re)booted.

New! autostart_yes [-m *server*][-p *server-port-number*]

Automatically start AnswerBook2 server when system is (re)booted.

browser [-m *server*][-p *server-port-number*] New!

Launch a Web browser for accessing AnswerBook2 Administration pages.

change_password -u *admin-id* [-m *server*]
[-p *server-port-number*]

Change authorized administrator's password.

del_admin -u *user-id* [-m *server*][-p *server-port-number*]

Delete the user from the list of authorized server administrators.

del_coll -t *collection-title* [-m *server*]
[-p] *server-port-number*]

Remove AnswerBook1 or AnswerBook2 collections from the specified server's database.

del_server -M *alternate-server*
-P *alternate-server-port-number* [-m *server*]
[-p *server-port-number*]

Delete alternate server from list of servers known to the specified server.

error_off [-m *server*][-p *server-port-number*]

Disable the server error log file.

error_on [-m *server*][-p *server-port-number*]

Enable the server error log file.

help [*command*] New!

List information about all commands or about a specified command.

list [-m *server*][-p *server-port-number*]

List AnswerBook1 and AnswerBook2 collections available on the specified server. The listing includes the books contained within collections.

list_server [-m *server*][-p *server-port-number*] New!

List all alternate servers defined for the specified server.

menu Display a condensed list of command options. New!

modify_server_name -s *new-server-name* [-m *server*]
[-p *server-port-number*]

Modify the server's name.

modify_server_port-a*new-server-port-number*[-m *server*]
[-p *server-port-number*]

Modify the server's port number.

`restart`

Restart local AnswerBook2 server. Requires root access.

`rotate_access [-m` *`server`*`][-p` *`server-port-number`*`]`

Save and reset the server access log file.

`rotate_error [-m` *`server`*`][-p` *`server-port-number`*`]`

Save and reset the server error log file.

`scan [-m` *`server`*`][-p` *`server-port-number`*`]`

Scan for locally installed collections (AnswerBook1 or AnswerBook2) and update the collections on specified server's database.

`start` Start local AnswerBook2 server. Requires root access.

`start -D` Start local AnswerBook2 server in debug mode. Requires root access.

`stop` Stop local AnswerBook2 server. Requires root access.

`view_access [-m` *`server`*`][-p` *`server-port-number`*`]`

View the contents of the server access log file.

`view_config [-m` *`server`*`][-p` *`server-port-number`*`]`

View the configuration settings of the server.

`view_error [-m` *`server`*`][-p` *`server-port-number`*`]`

View the contents of the server error log file.

New! **Note —** The `install` and `uninstall` subcommands have been removed from the `ab2admin` command in the Solaris 8 release. Instead, use the `pkgadd` and `pkgrm` commands to install and uninstall AnswerBook2.

Usage

`quit` Exit interactive mode.

`bye` Exit interactive mode.

`exit` Exit interactive mode.

New! `? [`*`command`*`]` Get help in interactive mode.

New! `h [`*`command`*`]` Get help in interactive mode.

Examples

The following example lists AnswerBook2 collections on the local system `paperbark`. As the example shows, you must define an AnswerBook2 administrator ID and assign a password before you can perform any administrative functions.

Note — The administrator ID does not need to match a user’s system login ID. The administrator ID is used only by the `ab2admin` command for performing document-related administrative functions on a specific server.

```
paperbark% alias ab2admin /usr/lib/ab2/bin/ab2admin
paperbark% ab2admin -o list

You are trying to access the AnswerBook2 administration functions on
  this document server. Access to these functions is controlled by
  AnswerBook2-specific administrator ids and passwords.

At the present time, AnswerBook2 administrative access control is turned
  on; however, there is no defined AnswerBook2 administrator ID and
  password on this system.

To define an AnswerBook2 administrator ID and password, perform the
  following steps:

1. Log in as root on the AnswerBook2 server machine.
2. Run the following command:/usr/lib/ab2/bin/ab2admin -o add_admin -u
  ID
  Where: ID is the login ID for the administrator and can consist of
  any number of characters and numbers.
3. The AnswerBook2 software will prompt you to enter and verify the
  password for the specified login ID.


paperbark% su
# /usr/lib/ab2/bin/ab2admin -o add_admin -u winsor

Please enter password for winsor :
Please reenter the same password :
Administrative user created successfully
# exit
paperbark% ab2admin -o list

To do any administration, a password is required.
Please enter administrative ID : winsor
Please enter administrative password :


Collection Listing
  AnswerBook2 Collection:
     "KCMS Collection"
     "OpenBoot Collection"
     "Solaris 8 Common Desktop Environment Developer Collection"
     "Solaris 8 Installation Collection"
     "Solaris 8 Software Developer Collection"
     "Solaris 8 System Administrator Collection"
     "Solaris 8 User Collection"
     "AnswerBook2 Version 1.4.1 Help Collection - Japanese"
     "AnswerBook2 Version 1.4.1 Help Collection"
     "AnswerBook2 ???? - zh"
     "AnswerBook2 ??? ?? - ko"
     "AnswerBook2 ?????? - zh_TW"
     "Answerbook2 Version 1.4.1-Hilfe-Kollektion"
     "Colección de Ayuda AnswerBook2 Version 1.4.1"
     "Collection d'Aide AnswerBook2 Version 1.4.1"
     "Collezione sulla guida di AnswerBook2 Version 1.4.1"
     "Hjälpsamling för Answerbook2 Version 1.4.1"

paperbark%
```

If you do not want to define an administrative user, you can turn off access control, as shown in the following example.

Warning — If you turn off access control, any user who can access your documentation server can modify the server.

```
paperbark% ab2admin -o auth_off

To do any administration, a password is required.
Please enter administrative ID : winsor
Please enter administrative password :


Administrative access control turned off.
This will allow any user to administer the AnswerBook2 server software.
paperbark%
```

The following example lists collections available on the local server paperbark using port 8888.

```
castle% ab2admin -o list -m paperbark -p 8888
```

To use ab2admin in interactive mode for the same operation as the previous example, type ab2admin and press Return. The ab2admin>> prompt is displayed, and you can then type interactive commands without specifying the -o option.

```
castle% ab2admin
ab2admin >> list -m paperbark -p 8888

To do any administration, a password is required.
Please enter administrative ID : winsor
Please enter administrative password :


Collection Listing
  AnswerBook2 Collection:
     "KCMS Collection"
     "OpenBoot Collection"
     "Solaris 8 Common Desktop Environment Developer Collection"
     "Solaris 8 Installation Collection"
     "Solaris 8 Software Developer Collection"
     "Solaris 8 System Administrator Collection"
     "Solaris 8 User Collection"
     "AnswerBook2 Version 1.4.1 Help Collection - Japanese"
     "AnswerBook2 Version 1.4.1 Help Collection"
```

```
         "AnswerBook2 ???? - zh"
         "AnswerBook2 ??? ?? - ko"
         "AnswerBook2 ?????? - zh_TW"
         "Answerbook2 Version 1.4.1-Hilfe-Kollektion"
         "Colección de Ayuda AnswerBook2 Version 1.4.1"
         "Collection d'Aide AnswerBook2 Version 1.4.1"
         "Collezione sulla guida di AnswerBook2 Version 1.4.1"
         "Hjälpsamling för Answerbook2 Version 1.4.1"

ab2admin >>
```

The following example installs an AnswerBook2 collection introduced to the system with pkgadd(1M) that did not get updated to the server database.

```
# ab2admin -o add_coll -d /opt/answerbooks/english/solaris_2.8/
     SUNWabsdk/collinfo
```

Note — -d *path* must include the collinfo file. Refer to "Using AnswerBook2 to View Online Information" in your information library for more information.

The following example inspects how a Solaris 8 AnswerBook2 collection is defined.

```
paperbark% cat /opt/answerbooks/english/solaris_8/
  SUNWabsdk/collinfo
dwCollections {
  coll.45.13 dwCollection
}
s
dwSetParam coll.45.13 {
  location /opt/answerbooks/english/soslaris_8/SUNWabsdk
  title "Solaris 8 Software Developer Collection"
  type EbtCollection
}
paperbark%
```

Files

/var/log/ab2/catalog/local.socat

Catalog file.

/var/log/ab2/catalog/remote.socat

Catalog file.

/var/log/ab2/catalog/delegate.socat

Catalog file.

/var/log/ab2/catalog/libcat.socat

Catalog file.

/var/log/ab2/logs/access_8888.log

Default access log file.

`/var/log/ab2/logs/errors_8888.log`

Default error log file.

`/usr/lib/ab2/dweb/data/config/ab2_collections.template`

AnswerBook2 collection database.

`/var/log/ab2/catalog/ab1_cardcatalog`

AnswerBook1 collection database.

`/usr/lib/ab2/dweb/data/config/admin_passwd`

File containing *username*:*password*.

Attributes

See `attributes`(5) for descriptions of the following attributes.

Attribute Type	Attribute Value
Availability	SUNWab2u

See Also

`pkgadd(1M), pkgrm(1M), attributes(5)`

ab2cd — Run AnswerBook2 Server from the Documentation CD

Synopsis

New!

```
/cdrom/cdrom0/ab2cd [-h][stop][-s][-d path-to-CD-mountpoint]
  [-p port-number][-s][-v]
```

Description

Use the `ab2cd` command to run an AnswerBook2 server directly from the Documentation CD by creating necessary space in the `/tmp/.ab2` directory to store configuration files and other necessary data. `ab2cd` requires root access to the system on which the Solaris Documentation CD is mounted.

Options

`-d` *path-to-CD-mountpoint*

Specify a mount point for the CD other than `/cdrom`.

`-h` Display a usage statement and a brief list of options.

New! `-p [`*port-number*`]`

Specify a port number to use for the server. Default is 8888.

`-s` Scan for AnswerBook1 and AnswerBook2 collections installed on the system and add them to the database of the AnswerBook2 server running from the CD.

`stop` Stop AnswerBook2 server running from the CD and remove any files in the `/tmp/.ab2` directory.

`-v` Display the version number of the `ab2cd` script. New!

Usage

`ab2cd` expects `/cdrom` as the default mount point. To override this default, use the `-d` option.

Using the `/cdrom/cdrom0/ab2cd stop` option shuts down the server running from the Documentation CD and cleans up any files in `/tmp/.ab2`.

New! By default, `ab2cd` tries to launch a Web browser (preferably Netscape Navigator) with the appropriate URL to display the user's Library Page. If Netscape is not found in the user's path, `ab2cd` looks for other browsers.

For an Answerbook2 server to read multibyte characters correctly, the `iconv` command must be installed on the system. If it is not, the `ab2cd` command starts the server but the user won't be able to view Asian book titles or other information correctly.

Examples

The following example runs the AnswerBook2 server from the CD, and then, if there is a Web browser in the root path, asks you if you want to launch a Web browser with the URL for the Library Page.

> **Note —** The default root path does not include a path to a Web browser. If no browser is found in the path, the `ab2cd` command displays a message telling you to start the Web browser and gives you the URL to use. To enable the `ab2cd` command to launch the Netscape browser, first add `/usr/dt/bin` to the root path.

```
paperbark% su
# /cdrom/cdrom0/ab2cd
.......................................................


Scanning for collections and attempting to start AnswerBook2 server from
  CD.

Please wait ...

Adding AnswerBook2 Help collection in C locale
Adding AnswerBook2 Help collection in de locale
Adding AnswerBook2 Help collection in es locale
Adding AnswerBook2 Help collection in fr locale
Adding AnswerBook2 Help collection in it locale
Adding AnswerBook2 Help collection in ja locale
Adding AnswerBook2 Help collection in ko locale
Adding AnswerBook2 Help collection in sv locale
Adding AnswerBook2 Help collection in zh locale
Adding AnswerBook2 Help collection in zh_TW locale
```

```
Solaris 8 System Administrator Collection
Solaris 8 User Collection
Solaris 8 Software Developer Collection
KCMS Collection
Solaris 8 Common Desktop Environment Developer Collection
Solaris 8 Installation Collection
OpenBoot Collection
sort: can't read /tmp/ab1_sort.1887: No such file or directory

Starting AnswerBook2 server from CD ...
Started http-8888 service on port 8888 (as daemon)


To read documents from the CD, open a browser with the URL:
http://paperbark:8888

Do you want to start Netscape now? [y,n] y


Starting browser with URL http://paperbark:8888 ....

After you are finished reading documents from the CD, stop the server
  using:
/cdrom/cdrom0/ab2cd stop

#
```

The following example adds any locally installed collections to the server's database. Also, no browser is defined in the user's path.

```
# /cdrom/cdrom0/ab2cd -s
.......................................................


Scanning for collections and attempting to start AnswerBook2 server from
  CD.

Please wait ...

Adding AnswerBook2 Help collection in C locale
Adding AnswerBook2 Help collection in de locale
Adding AnswerBook2 Help collection in es locale
Adding AnswerBook2 Help collection in fr locale
Adding AnswerBook2 Help collection in it locale
Adding AnswerBook2 Help collection in ja locale
Adding AnswerBook2 Help collection in ko locale
Adding AnswerBook2 Help collection in sv locale
Adding AnswerBook2 Help collection in zh locale
Adding AnswerBook2 Help collection in zh_TW locale

Solaris 8 System Administrator Collection
Solaris 8 User Collection
Solaris 8 Software Developer Collection
KCMS Collection
```

```
Solaris 8 Common Desktop Environment Developer Collection
Solaris 8 Installation Collection
OpenBoot Collection

Detecting local collections ...

Duplicate SUNWakcs.

Duplicate SUNWdtad.

Duplicate SUNWinab.

Duplicate SUNWopen.

Duplicate SUNWaadm.

Duplicate SUNWabe.

Duplicate SUNWabsdk.
sort: can't read /tmp/ab1_sort.3178: No such file or directory

Starting AnswerBook2 server from CD ...
Started http-8888 service on port 8888 (as daemon)


To read documents from the CD, open a browser with the URL:
http://paperbark:8888


After you are finished reading documents from the CD, stop the server
  using:
/cdrom/cdrom0/ab2cd stop

#
```

The following example launches ab2cd successfully and locates the ab2cd script in a specific place. However, support for all locales is not provided.

```
example# ab2cd -d /home/myuser/CDROM

Warning : AnswerBook2 requires the following iconv packages to be
  installed
prior to running ab2cd:
SUNWciu8 SUNWhiu8 SUNWjiu8 SUNWkiu8 SUNWuiu8

If you continue running ab2cd, multiple-byte characters might not
  display correctly and collections with non-English titles will not be
  viewable with this server.
Do you want to continue? [y,n] y
```

```
Scanning for collections and attempting to start AnswerBook2 server from
  CD.

Please wait ...

Adding AnswerBook2 Help collection in C locale
Skipping AnswerBook2 Help collection in de locale
Skipping AnswerBook2 Help collection in es locale
Skipping AnswerBook2 Help collection in fr locale
Skipping AnswerBook2 Help collection in it locale
Skipping AnswerBook2 Help collection in ja locale
Skipping AnswerBook2 Help collection in ko locale
Skipping AnswerBook2 Help collection in sv locale
Skipping AnswerBook2 Help collection in zh locale
Skipping AnswerBook2 Help collection in zh_TW locale

Solaris 8 System Administrator Collection
Solaris 8 User Collection
Solaris 8 Software Developer Collection
KCMS Collection
Solaris 8 Common Desktop Environment Developer Collection
Skipping Solaris 8 Installation Collection - de collection
Skipping Solaris 8 Userbook Collection - de collection
Skipping Solaris 8 Installation Collection - de collection
Solaris Common Desktop Environment Developer Collection
.
.
.
Skipping Solaris 8 Installation Collection - sv collection
Solaris XGL 3.3 AnswerBook

Starting AnswerBook2 server from CD ...
Started http-8888 service on port 8888


To read documents from the CD, open a browser with the URL:
http://ow:8888

Do you want to start Netscape now? [y,n] n

After you are finished reading documents from the CD, stop the server
  using:
/home/myuser/CDROM/ab2cd stop
```

The following example stops the AnswerBook2 server running from the CD and cleans up any files in the `/tmp` directory.

```
# /cdrom/cdrom0/ab2cd stop
.......................................................


Stopping AnswerBook2 server from ab2cd ...
```

```
If you have shut down your regular AnswerBook2 server in order to run
  /cdrom/cdrom0/ab2cd, use "ab2admin -o start" to restart your regular
  AnswerBook2 server.
#
```

Files

`/tmp/.ab2/*`	Configuration files and other necessary data.

Attributes

See `attributes`(5) for descriptions of the following attributes.

Attribute Type	**Attribute Value**
Availability	Solaris Documentation CD

See Also

`answerbook2(1), ab2admin(1M), attributes(5)`

ab2regsvr — Register an AnswerBook2 Document Server with the Federated Naming Service

Synopsis

`/usr/lib/ab2/bin/ab2regsvr [-d][-h][-l][-r]` *`server-url`*

Description

Use the `ab2regsvr` command to set up the appropriate namespace for the AnswerBook2 document server, depending on which naming service has been selected by the system administrator. The naming service can be `nis`, `nisplus`, or `files`.

Registering an AnswerBook2 document server with FNS enables a system administrator to specify the default AnswerBook2 server that users access when they select the CDE desktop icon or OpenWindows root menu. The server's URL does not have to be entered into a Web browser.

To register the server with `nis`, you must be logged in as root on the NIS master server. To register with `nisplus`, you must have administrative privileges; you can register either from the nisplus master or an nisplus client. To register for `files`, you must be logged in as root on the system; this command is system specific and is not seen on other systems.

Options

`-d`	Delete the AnswerBook2 entry in FNS.
`-h`	Display a usage statement and a brief list of options.
`-l`	List currently registered AnswerBook2 document servers.
`-r`	Replace the currently defined URL for AnswerBook2 with a new URL.

Operands

`server-url`	Fully qualified URL for users to access the registered server.

Examples

The following example registers a server named `imaserver` located at port 8888.

```
# ab2regsvr http://imaserver.eng.sun.com:8888/
```

Attributes

See `attributes`(5) for descriptions of the following attributes.

Attribute Type	**Attribute Value**
Availability	`SUNWab2u`

See Also

`fnlookup(1), attributes(5), fns(5)`

accept, reject — Accept or Reject Print Requests

Synopsis

```
/usr/sbin/accept destination...
/usr/sbin/reject [-r "reason"] destination...
```

Description

Use the `accept` and `reject` commands on a print server system to turn on or off a print queue that stores requests to be printed. These commands have no meaning on a client system. If you invoke `accept` or `reject` on a client system, a warning message is displayed and the command exits.

A printer must accept print requests before you can enable it by using the `enable` command.

Use `lpstat -a` to check if destinations are accepting or rejecting print requests.

Note — accept and reject affect queuing only on the print server's spooling system. Requests made from a client system remain queued in the client system's queuing mechanism until they are cancelled or accepted by the print server's spooling system.

Options

The following option is supported for reject.

-r "*reason*" Assign a reason for rejection of print requests for *destination*. Enclose reason in quotes if it contains blanks. *reason* is reported by lpstat -a. By default, *reason* is unknown reason for existing destinations, and new printer for destinations added to the system but not yet accepting requests.

Operands

destination The name of the destination accepting or rejecting print requests. Destination specifies the name of a printer or class of printers (see lpadmin(1M)). Specify *destination* using atomic name, for example, accept seachild. See printers.conf(4) for information regarding the naming conventions for atomic names.

Examples

The following example enables the printer seachild to accept print requests.

```
seachild% su
# accept seachild
destination "seachild" now accepting requests
#
```

Exit Status

0	Successful completion.
non-zero	An error occurred.

Files

/var/spool/lp/*

LP print queue.

Attributes

See attributes(5) for descriptions of the following attributes.

Attribute Type	**Attribute Value**
Availability	SUNWpcu
CSI	Enabled

Note — accept is CSI enabled except for the destination names.

See Also

enable(1),lp(1),lpstat(1),lpadmin(1M),lpsched(1M),printers.conf(4), attributes(5)

acct — Overview of Accounting and Miscellaneous Accounting Commands

Synopsis

```
/usr/lib/acct/acctdisk
/usr/lib/acct/acctdusg [-u filename][-p filename]
/usr/lib/acct/accton [filename]
/usr/lib/acct/acctwtmp reason filename
/usr/lib/acct/closewtmp
/usr/lib/acct/utmp2wtmp
```

Description

Accounting software is structured as a set of commands (consisting of both C programs and shell procedures) that enable you to collect and record data about user connect time, CPU time charged to processes, and disk usage. The SUNWaccr and SUNWaccu packages install the accounting commands in the /usr/lib/acct directories. Accounting data is collected and stored in /var/adm/pacct. You can use the accounting data to generate reports and charge fees for system use.

acctsh(1M) describes the set of shell procedures built on top of the C programs.

New!

Connect-time accounting is handled by various commands that write records into /var/adm/wtmpx, as described in utmpx(4) (changed from utmp in the Solaris 8 release). The programs described in acctcon(1M) convert this file into session and charging records, which are then summarized by acctmerg(1M).

Process accounting is performed by the system kernel. When a process terminates, one record per process is written to a file (normally /var/adm/pacct). The programs in acctprc(1M) summarize this data for charging purposes; acctcms(1M) summarizes command use. You can examine current process data by using acctcom(1).

You can merge process accounting records and connect time accounting records (or any accounting records in the tacct format described in acct(3HEAD)) and summarized into total accounting records with acctmerg (see tacct format in acct(3HEAD)). Use prtacct (see acctsh(1M)) to format any or all accounting records.

acctdisk
Read lines that contain user ID, login name, and number of disk blocks and convert them to total accounting records that can be merged with other accounting records. Return an error if the input file is corrupt or improperly formatted.

/usr/lib/acct/acctdusg [-u *filename*][-p *filename*]

Gather all disk accounting information. acctdusg can process a maximum of 3,000 users with each invocation. Read standard input (usually from find / -print), and compute disk resource consumption (including indirect blocks) by *login*.

accton [*filename*]

Without arguments, turn process accounting off. If you specify *filename*, it must be the name of an existing file to which the kernel appends process accounting records (see acct(2) and acct(3HEAD)).

acctwtmp *reason filename* New!

The startup command (see acctsh(1M)) adds a boot record to /var/adm/wtmpx showing the system name and login name. You can also use acctwtmp to write a utmpx(4) record (changed from utmp in the Solaris 8 release) to *filename*. The record contains the current time and a string of characters that describe the reason. A record type of ACCOUNTING is assigned (see utmpx(4)). *reason* must be a string of 11 or fewer characters, numbers, $, or spaces. For example, the following are suggestions for use in reboot and shutdown procedures.

acctwtmp "acctg on" /var/adm/wtmpx New!

acctwtmp "acctg off" /var/adm/wtmpx

closewtmp — For each user currently logged on, put a false DEAD_PROCESS record in the /var/adm/wtmpx file. runacct (see runacct(1M)) uses this false DEAD_PROCESS record so that the connect accounting procedures can track the time used by users logged on before runacct was invoked. New!

utmp2wtmp — For each user currently logged on, runacct uses utmp2wtmp to create an entry in the file /var/adm/wtmpx that is created by runacct. Entries in /var/adm/wtmpx enable subsequent invocations of runacct to account for connect times of users currently logged in. New!

Options

-u *filename* — Put records in *filename* records consisting of those file names for which acctdusg charges no one (a potential source for finding users trying to avoid disk charges).

-p *filename* — Specify a password file, *filename*. This option is not needed if the password file is /etc/passwd.

Examples

If necessary, install the SUNWaccr and SUNWaccu packages by using the pkgadd or admintool command.

The following example installs /etc/init.d/acct as the startup script for run level 2 and as the stop script for run level 0.

```
# ln /etc/init.d/acct /etc/rc2.d/S22acct
# ln /etc/init.d/acct /etc/rc0.d/K22acct
#
```

The following example modifies the admcrontab file to start the ckpacct, runacct, and monacct programs automatically.

```
# crontab -e adm
0 * * * * /usr/lib/acct/ckpacct
30 2 * * * /usr/lib/acct/runacct 2> /var/adm/acct/nite/fd2log
30 7 1 * * /usr/lib/acct/monacct
```

The following example modifies the root crontab file to start the dodisk program automatically.

```
# crontab -e
30 22 * * 4 /usr/lib/acct/dodisk
```

Edit /etc/acct/holidays to include national and local holidays. The following example shows the default /etc/acct/holidays file.

```
paperbark% more /etc/acct/holidays
* @(#)holidays  January 1, 1999
*
* Prime/Non-prime Table for UNIX Accounting System
*
* Curr  Prime   Non-Prime
* Year  Start   Start
*
  1999  0800    1800
*
* only the first column (month/day) is significant.
*
* month/day     Company
*               Holiday
*
1/1             New Years Day
7/4             Indep. Day
12/25           Christmas
paperbark%
```

The following example starts accounting. You can also start accounting by rebooting the system.

```
# /etc/init.d/acct start
```

Environment Variables

If any of the LC_* variables (LC_TYPE, LC_MESSAGES, LC_TIME, LC_COLLATE, LC_NUMERIC, and LC_MONETARY) (see environ(5)) are not set in the environment, the operational behavior of acct for each corresponding locale category is determined by the value of the LANG environment variable. If LC_ALL is set, its contents override both the LANG and the other LC_* variables. If none of the above variables is set in the environment, the C (U.S. style) locale determines how acct behaves.

LC_CTYPE Determine how acct handles characters. When LC_CTYPE is set to a valid value, acct can display and handle text and file names containing valid characters for that locale. acct can display and handle Extended Unix Code (EUC) characters where any character can be 1, 2, or 3 bytes wide. acct can also handle EUC characters of 1, 2, or more column widths. In the C locale, only characters from ISO 8859-1 are valid.

LC_TIME Determine how acct handles date and time formats. In the C locale, date and time handling follows the U.S. rules.

Files

/etc/passwd Used for login name to user ID conversions.

/usr/lib/acctHolds all accounting commands listed in subclass 1M of this manual.

/var/adm/pacct

Current process accounting file.

/var/adm/wtmpx New!

History of user access and administration information. wtmpx is an extended database file that replaces the obsolete wtmp database file.

Attributes

See attributes(5) for descriptions of the following attributes.

Attribute Type	**Attribute Value**
Availability	SUNWaccu

See Also

acctcom(1), acctcms(1M), acctcon(1M), acctmerg(1M), acctprc(1M), acctsh(1M), fwtmp(1M), runacct(1M), acct(2), acct(3HEAD), passwd(4), utmpx(4), attributes(5), environ(5) New!

System Administration Guide, Volume II

acctcms — Command Summary from Process Accounting Records

Synopsis

```
/usr/lib/acct/acctcms [-a[-o][-p]][-c][-j][-n][-s][-t] filename...
```

Description

acctcms reads one or more file names, normally in the form described in acct(3HEAD). It adds all records for processes that executed identically named commands, sorts them, and writes them to the standard output, normally using an internal summary format.

> **Note —** Unpredictable output results if you use the -t option on new-style internal summary format files or if you do not use it with old-style internal summary format files.

Options

-a Print output in ASCII instead of in the internal summary format. The output includes command name, number of times executed, total kcore-minutes, total CPU minutes, total real minutes, mean size (in kilobytes), mean CPU minutes per invocation, hog factor, characters transferred, and blocks read and written, as in acctcom(1). Output is normally sorted by total kcore-minutes.

You can use the following options only with the -a option.

-o Output a (non-prime) offshift-time-only command summary.

-p Output a prime-time-only command summary.

Using -o and -p together produces a combination prime-time and non-prime-time report. All the output summaries are total usage except number of times executed, CPU minutes, and real minutes, which are split into prime and non-prime.

-c Sort by total CPU time instead of total kcore-minutes.

-j Combine all commands invoked only once under ***other.

-n Sort by number of command invocations.

-s Report any file names encountered hereafter in internal summary format.

-t Process all records as total accounting records. The default internal summary format splits each field into prime- and non-prime-time parts. This option combines the prime- and non-prime-time parts into a single field that is the total of both and provides upward compatibility with old-style acctcms internal, summary-format records.

Examples

The following example shows a typical sequence for performing daily command accounting and for maintaining a running total.

```
castle% acctcms /var/adm/pacct > today
castle% cp total previoustotal
castle% acctcms -s today previoustotal > total
castle% acctcms -a -s today
                                     TOTAL COMMAND SUMMARY
COMMAND   NUMBER     TOTAL       TOTAL       TOTAL   MEAN    MEAN     HOG
```

```
   CHARS     BLOCKS
NAME       CMDS    KCOREMIN     CPU-MIN   REAL-MIN  SIZE-K  CPU-MIN  FACTOR
   TRNSFD      READ

TOTALS       381      502.81      0.44       101.08 1148.41   0.00  0.00
   529579616       2210

wtmpfix        1      216.15      0.27         0.46  792.24   0.27  0.59
   526385152         16
dtdbcach       1       62.53      0.02         0.02 3473.78   0.02  0.74
   670208         79
dtgreet        1       40.19      0.01         1.60 4230.74   0.01  0.01
   483264        166
dwhttpd        1       13.14      0.00         0.07 3754.67   0.00  0.05
   64424        300
dtpad          1       13.06      0.00         0.04 3405.91   0.00  0.09
   133568         13
acctcms        4       10.12      0.01         0.01  979.35   0.00  0.82
   248942         15
sh            54        8.42      0.01         2.17  885.89   0.00  0.00
   78527        115
dtsessio       3        8.24      0.00         0.01 1977.92   0.00  0.33
   740          2
```

(*Additional lines deleted from this example*)

Attributes

See `attributes`(5) for descriptions of the following attributes.

Attribute Type	**Attribute Value**
Availability	`SUNWaccu`

See Also

`acctcom(1)`,`acct(1M)`,`acctcon(1M)`,`acctmerg(1M)`,`acctprc(1M)`,`acctsh(1M)`, New!
`fwtmp(1M)`, `runacct(1M)`, `acct(2)`, `acct(3HEAD)`, `utmpx(4)`, `attributes(5)`

acctcon, acctcon1, acctcon2 — Connect-Time Accounting

Synopsis

```
/usr/lib/acct/acctcon [-l lineuse][-o reboot]
/usr/lib/acct/acctcon1 [-p][-t][-l lineuse][-o reboot]
/usr/lib/acct/acctcon2
```

Description

Use the `acctcon` commands to convert a sequence of login/logoff records to total accounting records (see the `tacct` format in `acct`(3HEAD)). The login/logoff records are read from standard input.

acctcon is a combination of the acctcon1 and acctcon2 commands. Use acctcon1 to convert login/logoff records from the fixed /var/adm/wtmpx file to ASCII output. Use acctcon2 to read the ASCII records produced by acctcon1 and convert them to tacct records. You can use acctcon1 with any or all of the options.

New!

Note — The file /var/adm/wtmpx (changed from wtmp in the Solaris 8 release) is usually the source of the login/logoff records. However, because the file may contain corrupted records or system date changes, you should first fix it by using the wtmpfix (see fwtmp(1M)) command with the /var/adm/wtmpx file as an argument. You can then redirect the fixed version of the /var/adm/wtmpx file to acctcon. The tacct records are written to standard output.

The acctcon, acctcon1, and acctcon2 commands can process the following maximums during a single invocation. If the actual number of any one of these items exceeds the maximum, the command does not succeed.

- 6,000 distinct sessions.
- 1,000 distinct terminal lines.
- 2,000 distinct login names.

Options

-p
Print input only, showing line name, login name, and time (in both numeric and date/time formats).

-t
Use the last time found from acctcon1 input instead of the current time, thus ensuring reasonable and repeatable numbers for noncurrent files. The acctcon1 command maintains a list of lines on which users are logged in. When it reaches the end of its input, acctcon1 produces a session record for each line that still seems to be active. acctcon1 ordinarily assumes that its input is a current file so that it uses the current time as the ending time for each session still in progress.

-l *lineuse*
Create a *lineuse* file containing a summary of line usage that shows the line name, number of minutes used, percentage of total elapsed time used, number of sessions charged, number of logins, and number of logoffs. This file helps track line usage, identify bad lines, and find software and hardware oddities. Hangup, termination of login(1), and termination of the login shell each generate logoff records, so that the number of logoffs is often three to four times the number of sessions. See init(1M) and utmpx(4).

-o *reboot*
Fill *reboot* with an overall record for the accounting period, giving starting time, ending time, number of reboots, and number of date changes.

Examples

The following example uses the wtmpfix file to fix the /var/adm/wtmpx file and redirects the output to /tmp/tmpwtmpx.

```
paperbark% /usr/lib/acct/wtmpfix /var/adm/wtmpx > /tmp/tmpwtmpx

checking offset 351665
checking offset 351666
checking offset 351667
checking offset 351670
checking offset 351671
checking offset 351672
...(Additional lines deleted from this example)
paperbark%
```

The following example shows a typical use of the `acctcon` command, using the fixed `/tmp/tmpwtmpx` file as input, and shows the contents of the `lineuse` and `reboots` files created by `acctcon`.

```
paperbark% /usr/lib/acct/acctcon -l lineuse -o reboots <
  /tmp/tmpwtmpx > ctacct
paperbark% more lineuse
TOTAL DURATION IS 50275 MINUTES
LINE         MINUTES  PERCENT  # SESS  # ON  # OFF
console      5718     11       28      28    108
pts/7        0        0        0       0     9
TOTALS       5718     --       28      28    117
paperbark% more reboots
from Wed Sep  8 12:18:43 1999
to   Wed Oct 13 10:13:15 1999
43      system boot
34      run-level 3
23      run-level 0
2       run-level 6
2       acctg on
1       acctcon
paperbark%
```

The following example shows a typical use of the `acctcon1` command.

```
paperbark% /usr/lib/acct/acctcon1 -l lineuse -o reboots <
  /tmp/tmpwtmpx | sort +1n +2 > ctmp
The old time is: Thu Jan  1 08:00:00 1970
the new time is: Tue Sep 21 13:14:26 1999
The old time is: Tue Sep 21 13:14:26 1999
the new time is: Tue Sep 21 13:14:30 1999
The old time is: Tue Sep 21 13:14:30 1999
the new time is: Tue Sep 21 13:14:40 1999
...(Additional lines deleted from this example)
paperbark%
```

The following example shows a typical use of the `acctcon2` command and shows the contents of the `ctmp` file created by `acctcon2`.

```
paperbark% /usr/lib/acct/acctcon2 < ctmp > ctacct
paperbark% more ctmp
0   0     root    81      0      939104584   Tue Oct  5 14:23:04 1999
0   1001  winsor  12714   50400  939627987   Mon Oct 11 15:46:27 1999
0   1001  winsor  13310   0      938145574   Fri Sep 24 11:59:34 1999
```

```
0   1001  winsor  14575   50400  939715806   Tue Oct 12 16:10:06 1999
0   1001  winsor  17980   0      939609907   Mon Oct 11 10:45:07 1999
0   1001  winsor  18877   0      939084046   Tue Oct  5 08:40:46 1999
...(Additional lines deleted from this example)
paperbark%
```

Files

New! `/var/adm/wtmpx`

History of user access and administration information.The `wtmpx` database file supersedes the obsolete `wtmp` database file.

Attributes

See `attributes`(5) for descriptions of the following attributes.

Attribute Type	Attribute Value
Availability	`SUNWaccu`

See Also

New! `acctcom(1)`, `login(1)`, `acct(1M)`, `acctcms(1M)`, `acctmerg(1M)`, `acctprc(1M)`, `acctsh(1M)`, `fwtmp(1M)`, `init(1M)`, `runacct(1M)`, `acct(2)`, `acct(3HEAD)`, `utmpx(4)`, `attributes(5)`

System Administration Guide, Volume II

acctdisk — Convert Accounting Data to Total Accounting Records

Synopsis

```
/usr/lib/acct/acctdisk
```

Description

See `acct`(1M).

acctdusg — Compute Disk Resource Consumption by Login

Synopsis

```
/usr/lib/acct/acctdusg [-u filename][-p filename]
```

Description

See `acct`(1M).

acctmerg — Merge or Add Total Accounting Files

Synopsis

```
/usr/lib/acct/acctmerg [-a][-i][-p][-t][-u][-v][filename]...
```

Description

Use the `acctmerg` command to read standard input and up to nine additional files, all in the `tacct` format (see `acct`(3HEAD)) or an ASCII version thereof. `acctmerg` merges these inputs by adding records whose keys (normally user ID and name) are identical and expects the inputs to be sorted on those keys.

Options

`-a` Produce output in ASCII version of `tacct`.

`-i` Specify that input files are in ASCII version of `tacct`.

`-p` Print input with no processing.

`-t` Produce a single record that totals all input.

`-u` Summarize by user ID instead of by user ID and name.

`-v` Produce output in verbose ASCII format with more precise notation for floating-point numbers.

Examples

The following example shows a sequence that is useful for making repairs to any files kept in the `acctmerg` format.

```
castle% acctmerg -v < filename1 > filename2
Edit filename2 as desired.
castle% acctmerg -i < filename2 > filename1
```

Attributes

See `attributes`(5) for descriptions of the following attributes.

Attribute Type	**Attribute Value**
Availability	SUNWaccu

See Also

`acctcom(1)`, `acct(1M)`, `acctcms(1M)`, `acctcon(1M)`, `acctprc(1M)`, `acctsh(1M)`, `fwtmp(1M)`, `runacct(1M)`, `acct(2)`, `acct(3HEAD)`, `utmpx(4)`, `attributes(5)` **New!**

System Administration Guide, Volume II

accton — Append Process Accounting Records to an Existing File

Synopsis

```
/usr/lib/acct/accton [filename]
```

Description

See acct(1M).

acctprc, acctprc1, acctprc2 — Process Accounting

Synopsis

```
/usr/lib/acct/acctprc
/usr/lib/acct/acctprc1 [ctmp]
/usr/lib/acct/acctprc2
```

Description

Use the acctprc command to read the standard input in the form described by acct(3HEAD) and to convert it to total accounting records (see the tacct record in acct(3HEAD)). acctprc divides CPU time into prime time and non-prime time and determines mean memory size (in memory segment units). It then summarizes the tacct records according to user IDs and adds login names corresponding to the user IDs. The summarized records are then written to the standard output.

acctprc1 reads input in the form described by acct(3HEAD), adds login names corresponding to user IDs then writes for each process an ASCII line giving user ID, login name, prime CPU time (ticks), non-prime CPU time (ticks), and mean memory size (in memory segment units). If you specify *ctmp*, it is expected to contain a list of login sessions sorted by user ID and login name. If you do not supply this file, acctprc1 obtains login names from the password file just as acctprc does. The information in *ctmp* helps it distinguish between different login names sharing the same user ID.

From the standard input, acctprc2 reads records in the form written by acctprc1, summarizes them according to user ID and name, then writes the sorted summaries to the standard output as total accounting records.

The acctprc, acctprc1, and acctprc2 commands can process the following maximum values during a single invocation. If at some point the actual number of any of these items exceeds the maximum, the command does not succeed.

- 6,000 distinct sessions.
- 1,000 distinct terminal lines.
- 2,000 distinct login names.

Notes

Although it is possible for acctprc1 to distinguish among login names that share user IDs for commands run normally, it is difficult for acctprc1 to do so for those commands run from cron(1M). You can convert more precisely by using the acctwtmp program in acct(1M). acctprc does not distinguish among users with identical user IDs.

A memory segment of the mean memory size is a unit of measure for the number of bytes in a logical memory segment on a particular processor.

Examples

The following example shows a typical use of the acctprc command.

```
paperbark% /usr/lib/acct/acctprc < /var/adm/pacct > ptacct
paperbark% file ptacct
ptacct:         data
paperbark% ls -l ptacct
-rw-r--r--   1 winsor   staff        304 Oct 13 15:21 ptacct
paperbark%
```

The following example shows a typical use of the acctprc1 command.

```
paperbark% /usr/lib/acct/acctprc1 ctmp < /var/adm/pacct
0       root    3       0       65
0       root    0       0       0
0       root    1       0       88
(Additional lines deleted from this example)
1       daemon  1       0       126
1       daemon  3       0       372
1       daemon  21      0       470
1       daemon  7       0       95
0       root    2       0       101
0       root    1       0       54
0       root    2       0       60
(Additional lines deleted from this example)
1001    winsor  1       0       190
1001    winsor  1       0       193
1001    winsor  1       0       226
1001    winsor  1       0       207
(Additional lines deleted from this example)
paperbark%
```

The following example show a typical use of the acctprc2 command.

```
paperbark% acctprc2 > ptacct
```

Files

/etc/passwd   System password file.

Attributes

See attributes(5) for descriptions of the following attributes.

Attribute Type	Attribute Value
Availability	SUNWaccu

See Also

New!

acctcom(1),acct(1M),acctcms(1M),acctcon(1M),acctmerg(1M),acctsh(1M), cron(1M), fwtmp(1M), runacct(1M), acct(2), acct(3HEAD), utmpx(4), attributes(5)

acctsh, chargefee, ckpacct, dodisk, lastlogin, monacct, nulladm, prctmp, prdaily, prtacct, shutacct, startup, turnacct — Shell Procedures for Accounting

Synopsis

```
/usr/lib/acct/chargefee login-name number
/usr/lib/acct/ckpacct [blocks]
/usr/lib/acct/dodisk [-o][filename...]
/usr/lib/acct/lastlogin
/usr/lib/acct/monacct number
/usr/lib/acct/nulladm filename...
/usr/lib/acct/prctmp filename
/usr/lib/acct/prdaily [-c][-l][mmdd]
/usr/lib/acct/prtacct filename ["heading"]
/usr/lib/acct/shutacct ["reason"]
/usr/lib/acct/startup
/usr/lib/acct/turnacct on | off | switch
```

Description

> **Note —** See runacct(1M) for the main daily accounting shell script, which daily accumulates connect, process, fee, and disk accounting. It also creates summaries of command usage.

chargefee *login-name number*

If you provide special user services on a request basis, such as restoring files or remote printing, you may want to bill users by runningchargefee.chargefeerecordschargesinthe/var/adm/fee file and merges new entries into the total accounting records each time the runacct(1M) command is executed.

ckpacct [*blocks*]

Initiated by cron(1M) to periodically check the size of /var/adm/pacct. If the size exceeds *blocks*, 500 by default, invoke turnacct with argument *switch*. If the number of free disk blocks in the /var file system falls below 500, automatically turn off the collection of process accounting records by using the off argument to turnacct. When at least 500 blocks are restored, activate the accounting again on the next invocation of ckpacct. This feature is sensitive to the frequency at which ckpacct is executed, usually by cron.

dodisk [-o][*filename*...]

Invoked by cron to perform the disk accounting functions. The following example modifies the root crontab file to start the dodisk command automatically.

```
# crontab -e
30 22 * * 4 /usr/lib/acct/dodisk
```

For a complete example of how to set up accounting, see “Examples” on page 996.

Information gathered by running dodisk is stored in /var/adm/acct/nite/disktacct. The information in this file is overwritten the next time you run dodisk. Therefore, you should avoid running dodisk twice in the same day.

lastlogin Invoked by runacct(1M) to update /var/adm/acct/sum/loginlog, which shows the last date on which each person logged in.

monacct *number*

Invoke once each month or each accounting period. *number* indicates which month or period it is. If you do not specify *number*, the default is the current month (01-12). This default is useful if monacct is executed by cron(1M) on the first day of each month. monacct creates a report based on data stored in /var/adm/acct/fiscal that has been updated daily by runacct. After creating the report, monacct cleans up the summary files in /var/adm/acct/sum to prepare them for new runacct data.

nulladm *filename*...

Create *filename* with mode 664, and ensure that owner and group are adm. nulladm is called by various accounting shell procedures.

prctmp *filename*

Print the session record file (normally /var/adm/acct/nite/ctmp created by acctcon1 (see acctcon(1M)).

prdaily [-c][-l][*mmdd*]

Invoke by runacct(1M) to format a report of the previous day's accounting data. The report resides in /var/adm/acct/sum/rprt/*mmdd* where *mmdd* is the month and day of the report. You can print the current daily accounting reports by typing `prdaily`. Print accounting reports for previous days by using the *mmdd* option and specifying the exact report date desired.

`prtacct` *`filename`* `["`*`heading`*`"]`

Use to format and print any total accounting (`tacct`) file.

`shutacct ["`*`reason`*`"]`

New!

When the `shutdown` command is used, `shutacct` is invoked automatically to turn off process accounting and append a *`reason`* record to `/var/adm/wtmpx` (changed from `wtmp` in the Solaris 8 release).

`startup` Invoke when the system is brought to a multiuser state to turn on process accounting.

`turnacct on | off | switch`

Provide an interface to `accton` (see `acct`(1M)) to turn on or off process accounting. The `switch` argument moves the current `/var/adm/pacct` to the next free name in `/var/adm/pacct`*`incr`* (where *`incr`* is a number starting with 1 and incrementing by 1 for each additional `pacct` file), then turns on accounting again. This procedure is called by `ckpacct` and thus can be taken care of by `cron` and used to keep `pacct` to a reasonable size. `shutacct` uses `turnacct` to stop process accounting. `startup` uses `turnacct` to start process accounting.

Options

`-c` Print a report of exceptional resource usage by command. You can use this option only on the accounting data for the current day.

`-l` Print a report of exceptional usage by login ID for the specified date. Previous daily reports are cleaned up and are, therefore, inaccessible after each invocation of `monacct`.

`-o` Use `acctdusg` (see `acct`(1M)) to do a slower version of disk accounting by login directory.

Operands

`filename...` Specify one or more file-system names where disk accounting is done. If you specify *`filename...`*, disk accounting is done only on these file systems. With the `-o` option, *`filename...`* should be mount points of mounted file systems. If you omit the `-o` option, *`filename...`* should be the special file names of mountable file systems.

Files

`/usr/lib/acct`

Holds all accounting commands listed in section 1M.

`/usr/lib/acct/ptecms.awk`

Contains, by command name, the limits for exceptional usage.

`/usr/lib/acct/ptelus.awk`

Contains, by login ID, the limits for exceptional usage.

`/var/adm/acct/fiscal`

Fiscal reports directory.

`/var/adm/acct/nite`

Working directory.

`/var/adm/acct/sum`

Summary directory contains information for `monacct`.

`/var/adm/acct/sum/loginlog`

File updated by last login.

`/var/adm/fee` Accumulator for fees.

`/var/adm/pacct`

Current file for per-process accounting.

`/var/adm/pacct`*`incr`*

Used if `pacct` gets large and during execution of daily accounting procedure.

`/var/adm/wtmpx` New!

History of user access and administration information. The `wtmpx` database file supersedes the obsolete `wtmp` database file.

Attributes

See `attributes`(5) for descriptions of the following attributes.

Attribute Type	**Attribute Value**
Availability	`SUNWaccu`

See Also

New!

`acctcom(1), acct(1M), acctcms(1M), acctcon(1M), acctmerg(1M), acctprc(1M), cron(1M), fwtmp(1M), runacct(1M), acct(2), acct(3HEAD), utmpx(4), attributes(5)`

acctwtmp — Write a utmpx Record to a File

Synopsis

```
/usr/lib/acct/acctwtmp reason filename
```

Description

See acct(1M).

adbgen — Generate adb Script

Synopsis

```
/usr/lib/adb/adbgen [-m model] filename.adb...
```

Description

Use the adbgen command to write adb(1) macros that do not contain hard-coded dependencies on structure member offsets. The input to adbgen is a file named *filename*.adb that contains the following elements.

- Header information.
- A null line.
- The name of a structure.
- An adb script.

adbgen deals with one structure only per file; all member names are assumed to be in this structure. The output of adbgen is an adb script in *filename*. adbgen generates a C program that determines structure member offsets and sizes, which in turn generate the adb script.

The header lines up to the null line are copied verbatim into the generated C program. Typically, these are #include statements, which include the headers containing the relevant structure declarations.

The adb script part can contain any valid adb commands (see adb(1)) and can also contain adbgen requests, each enclosed in braces ({}). The following list describes the request types.

- Print a structure member. The request form is {*member*,*format*}. *member* is a member name of the structure given earlier, and *format* is any valid adb format request or any of the adbgen format specifiers listed below, such as {POINTER}. For example, to print the p_pid field of the proc structure as a decimal number, write {p_pid,d}.
- Print the appropriate adb format character for the given adbgen format specifier. This action takes the data model into consideration. The request form is {*format specifier*}. The following adbgen format specifiers are valid.

{POINTER}	Pointer value in hexadecimal.
{LONGDEC}	Long value in decimal.
{ULONGDEC}	Unsigned long value in decimal.
{ULONGHEX}	Unsigned long value in hexadecimal.
{LONGOCT}	Long value in octal.
{ULONGOCT}	Unsigned long value in octal.

- Reference a structure member. The request form is {**member*,*base*}. *member* is the member name whose value you want, and *base* is an adb register name that contains the base address of the structure. For example, to get the p_pid field of the proc structure, get the proc structure address in an adb register, say, <f, and write {*p_pid,<f}.
- Tell adbgen that the offset is valid. The request form is {OFFSETOK}. This form is useful after you invoke another adb script that moves the adb dot.
- Get the size of the structure. The request form is {SIZEOF}. adbgen replaces this request with the size of the structure. This form is useful in incrementing a pointer to step through an array of structures.
- Calculate an arbitrary C expression. The request form is {EXPR,*expression*}. adbgen replaces this request with the value of the expression. This form is useful when more than one structure is involved in the script.
- Get the offset to the end of the structure. The request form is {END}. This form is useful at the end of the structure to get adb to align the dot for printing the next structure member.

adbgen tracks the movement of the adb dot and generates adb code to move forward or backward as needed before printing any structure member in a script. adbgen's model of the behavior of adb's dot is simple: it assumes that the first line of the script is of the form struct_address/*adb text* and that subsequent lines are of the form +/*adb text*. The adb dot then moves in a sane fashion. adbgen does not check the script to ensure that these limitations are met. adbgen also checks the size of the structure member against the size of the adb format code and warns if they are not equal.

Options

-m *model* Specify the data type model to be used by adbgen for the macro. This model affects the outcome of the {*format specifier*} requests and the offsets and sizes of data types. *model* can be ilp32 for 32-bit programs or lp64 for 64-bit programs. If you do not specify the -m option, the default data type model is ilp32.

Operands

filename.adb Input file that contains header information followed by a null line, the name of the structure, and finally an adb script.

Examples

Suppose you have an include file `x.h`, that contains the following data.

```
struct x {
       char *x_cp;
       char x_c;
       int x_i;
};
```

An adbgen file called `script.adb` to print the file `x.h` would look like the following example.

```
#include "x.h"
x ./"x_cp"16t"x_c"8t"x_i"n{x_cp,{POINTER}}{x_c,C}{x_i,D}
```

When you run adbgen using the following command

```
% /usr/lib/adb/adbgen script.adb
```

the output file `script` contains the following information.

```
./"x_cp"16t"x_c"8t"x_i"nXC3+D
```

For a macro generated for a 64-bit program using the `lp64` data model as follows,

```
% /usr/lib/adb/adbgen/ -m lp64 script.adb
```

the output file script would contain

```
./"x_cp"16t"x_c"8t"x_i"nJC3+D
```

To invoke the script, type the following command.

```
castle% adb program
x$<script
```

Files

`/usr/platform/`*`platform-name`*`/lib/adb/*`

Platform-specific adb scripts for debugging the 32-bit kernel.

`/usr/platform/`*`platform-name`*`/lib/adb/sparcv9/*`

Platform-specific adb scripts for debugging the 64-bit SPARC V9 kernel.

`/usr/lib/adb/*`

adb scripts for debugging the 32-bit kernel.

`/usr/lib/adb/sparcv9/*`

adb scripts for debugging the 64-bit SPARC V9 kernel.

Note — You can find *`platform-name`* with the `-i` option of uname(1).

Attributes

See attributes(5) for descriptions of the following attributes.

Attribute Type	Attribute Value
Availability	SUNWesu

See Also

adb(1), uname(1), kadb(1M), attributes(5)

Diagnostics

Warnings are given about structure member sizes not equal to adb format items and about badly formatted requests. The C compiler complains if you reference a structure member that does not exist. It also complains about an ampersand before array names; you can ignore these complaints.

Bugs

adb syntax is ugly; there should be a higher-level interface for generating scripts.

Structure members that are bit fields cannot be handled because C does not give the address of a bit field. The address is needed to determine the offset.

add_drv — Add a New Device Driver to the System

Synopsis

```
/usr/sbin/add_drv [-b basedir][-c class-name][-i 'identify-name...']
  [- m 'permission','...'][-n][-f][-v] device-driver
```

Description

Use the add_drv command to inform the system about newly installed device drivers.

Each device on the system has a name associated with it. This name is represented by the name property for the device. Similarly, the device can also have a list of driver names associated with it. This list is represented by the compatible property for the device.

The system determines which devices are managed by the driver being added by examining the contents of the name property and the compatible property (if it exists) on each device. If the value in the name property does not match the driver being added, each entry in the compatible property is tried, in order, until either a match occurs or there are no more entries in the compatible property.

Note — In some cases, adding a new driver may require a reconfiguration boot. Aliases may require quoting (with double quotes) if they contain numbers.

You can add a driver for a device already being managed by a different driver when the driver being added appears in the device's compatible list before the current driver. In such cases, you must do a reconfiguration boot (`boot -r`) (see `boot`(1M) and `kernel`(1M)). After the reconfiguration boot, device nodes in `/devices`, entries in /`dev`, and references to these files may no longer be valid (see the `-v` option). If a reconfiguration boot is required to complete the driver installation, `add_drv` fails unless you specify the `-f` option.

You should not run `add_drv` when installing a STREAMS module. See the *STREAMS Programming Guide* for details.

Options

`-b` *basedir* Install the driver on the system with a root directory of *basedir* instead of installing on the system executing `add_drv`. You typically use this option in package postinstallation scripts when the package is not being installed on the system executing the `pkgadd` command. The system using *basedir* as its root directory must reboot to complete the driver installation.

`-c`*class-name* Specify that the driver being added to the system export the class *class-name*.

`-f` Force addition of the driver even if a reconfiguration boot is required. See the `-v` option. Normally, if a reconfiguration boot is required to complete the configuration of the driver into the system, `add_drv` does not add the driver.

`-i '`*identify-name*`'`

Specify a white-space-separated list of aliases for the driver *device-driver*.

`-m '`*permission*`'`

Specify the file-system permissions for device nodes created by the system on behalf of *device-driver*.

`-n` Do not try to load and attach *device-driver*; just modify the system configuration files for the *device-driver*.

`-v` Provide additional information regarding the success or failure of a driver's configuration into the system.

Examples

The following example adds the `SUNW,example` driver to the system with an alias name of `SUNW,alias`. It assumes the driver has already been copied to `/usr/kernel/drv`.

```
# add_drv -m '* 0666 bin bin','a 0644 root sys' -i 'SUNW,alias'
    SUNW,example
```

Every minor node created by the system for the `SUNW,example` driver has the permission `0666` and must be owned by user `bin` in the group `bin`, except for the minor device `a`, which is owned by `root`, group `sys`, and has a permission of `0644`.

The following example adds the driver to the client /export/root/sun1. The driver is installed and loaded when the client machine, sun1, is rebooted. This example produces the same result as the first, except the changes are on the diskless client, sun1, and the client must be rebooted to complete the driver installation.

```
# add_drv -m '* 0666 bin bin','a 0644 root sys' -i 'SUNW,alias' -b
    /export/root/sun1 SUNW,example
```

The following example adds a new driver for a device that is already managed by an existing driver. Consider a device that is currently managed by the driver dumb_framebuffer. The name and compatible properties for this device are as follows.

```
name="display" compatible="whizzy_framebuffer", "dumb_framebuffer"
```

If you use add_drv without any options to add the whizzy_framebuffer driver, the following message is displayed.

```
# add_drv whizzy_framebuffer
Error: Could not install driver (whizzy_framebuffer)
Device managed by another driver.
#
```

If you specify the -v option, an error message is also displayed, as shown in the following example.

```
# add_drv -v whizzy_framebuffer
Error: Could not install driver (whizzy_framebuffer)
Device managed by another driver.
Driver installation failed because the following entries in /devices
  would be affected:
/devices/iommu@f,e0000000/sbus@f,e0001000/display[:*] (Device currently
  managed by driver "dumb_framebuffer")
The following entries in /dev would be affected:
/dev/fbs/dumb_framebuffer0
#
```

If you specify both the -v and -f options, as shown in the following example, the driver is added.

```
# add_drv -vf whizzy_framebuffer
#
```

You must perform a reconfiguration boot (boot -r) to complete the installation of this driver.

The following entries in /devices are affected.

```
/devices/iommu@f,e0000000/sbus@f,e0001000/display[:*] (Device currently
    managed by driver "dumb_framebuffer"
```

The following entries in /dev are affected.

```
/dev/fbs/dumb_framebuffer0
```

The above example is currently relevant only to devices exporting a generic device name.

Exit Status

`0`	Success.
`1`	Failure.

Files

`/kernel/drv` Boot device drivers.

`/usr/kernel/drv`

Other drivers that could potentially be shared among platforms.

`` /platform/`uname -i`/kernel/drv ``

Platform-dependent drivers.

`/etc/driver_aliases`

Driver aliases file.

`/etc/driver_classes`

Driver classes file.

`/etc/minor_perm`

Minor node permissions.

`/etc/name_to_major`

Major number binding.

Attributes

See `attributes`(5) for descriptions of the following attributes.

Attribute Type	**Attribute Value**
Availability	`SUNWcsu`

See Also

`boot(1M)`, `devlinks(1M)`, `disks(1M)`, `drvconfig(1M)`, `kernel(1M)`, `modinfo(1M)`, `ports(1M)`, `rem_drv(1M)`, `tapes(1M)`, `driver.conf(4)`, `system(4)`, `attributes(5)`, `ddi_create_minor_node(9F)`

Writing Device Drivers

Bugs

`add_drv` tries to use a full path name for *`device-driver`*. However, the kernel does not use the full path name; it uses only the final component and searches the internal driver search path for the driver. This behavior can lead to the kernel loading a different driver than expected.

For this reason, it is not recommended that you use `add_drv` with a full path name. See `kernel`(1M) for more information on the driver search path.

add_install_client — Script to Add or Remove Clients for Network Installation

Synopsis

```
cdrom-mnt-pt/Solaris_8/Tools/add_install_client [-i IP-address]
  [-e Ethernet-address][-s server-name : path][-c server-name :
  path][-n [server] : name-service [(netmask]][-p server-name : path]
  hostname platform-group
```

Description

See install_scripts(1M).

add_to_install_server — Script to Copy Packages from Additional Solaris Copackaged CDs to an Existing Net Install Server

New!

Synopsis

```
cdrom-mnt-pt/Sol_8_sparc_2/Solaris_8/Tools/add_to_install_server[-s]
  [-p product-image-path] install-server-path
```

Description

See install_scripts(1M).

addbadsec — Map Out Defective Disk Blocks

Synopsis

```
addbadsec [-p][-a blkno [blkno...]][-f filename] raw-device
```

Description

Use addbadsec to map out bad disk blocks on IA systems. Normally, these blocks are identified during surface analysis. Occasionally, the disk subsystem reports unrecoverable data errors indicating a bad block. You can feed a block number reported in this way directly into addbadsec, and the block is remapped. addbadsec first tries hardware remapping. This feature is supported on SCSI drives and takes place at the disk hardware level. If the target is an IDE drive, then software remapping is used. For software remapping to succeed, the partition must contain an alternate slice and there must be room in this slice to perform the mapping.

Bad blocks lead to data loss. Remapping a defective block does not repair a damaged file. If a bad block occurs to a disk-resident file-system structure such as a superblock, you may have to recover the entire slice from a backup.

Note — The `format`(1M) command is available to format, label, analyze, and repair SCSI disks. This command is included with the `addbadsec`, `diskscan`(1M), `fdisk`(1M), and `fmthard`(1M) commands available for IA. To format an IDE disk, use the DOS `format` command; however, to label, analyze, or repair IDE disks on IA systems, use the Solaris `format`(1M) command.

Options

`-a` Add the specified blocks to the hardware or software map. If you specify more than one block number, quote the entire list and separate block numbers with white space.

`-f` Add the specified blocks to the hardware or software map. List the bad blocks, one per line, in the specified file.

`-p` Print the current software map. The output shows the defective block and the assigned alternate. You cannot use this option to print the hardware map.

Operands

`raw-device` The address of the disk drive.

Files

The raw device should be `/dev/rdsk/c`*`?`*`[t`*`?`*`]d`*`?`*`p0`.

Attributes

See `attributes`(5) for descriptions of the following attributes.

Attribute Type	**Attribute Value**
Architecture	`IA`
Availability	`SUNWcsu`

See Also

`disks(1M), diskscan(1M), fdisk(1M), fmthard(1M), format(1M), attributes(5)`

admintool — System Administration with a Graphical User Interface

Synopsis

```
/usr/bin/admintool
```

Description

Note — admintool modifies files only on the local system; that is, the system on which you are running admintool. admintool does not modify or update global networked databases such as NIS or NIS+.

admintool is not the tool for a distributed environment. It is used for local administration.

admintool is a graphical user interface that enables you to accomplish several system administration tasks on a local system. You can run admintool as root or by using your user name if you are a member of the sysadmin group (GID 14). Members of the sysadmin group can use admintool to create, delete, and modify local system files. Nonmembers have read-only permissions (where applicable).

Help is available through the Help button.

Warning — If you use admintool to add a host, and your local system and your site uses a network nameservice such as NIS or NIS+, admintool host operations may not have the desired effect. The reason is that information in the network nameservice takes precedence over the information in the local /etc/hosts file, which is where admintool updates information.

Usage

admintool enables you to do the following tasks.

- Add, delete, or modify user accounts. admintool makes the appropriate changes to the system's /etc/passwd file (see passwd(4)).
- Add, delete, or modify groups. admintool makes the appropriate changes to the system's /etc/group file (see group(4)).
- Add, delete, or modify hosts. admintool makes the appropriate changes to the system's /etc/hosts file (see hosts(4)).
- Add or delete access to a printer or modify a system's printer access. admintool makes the appropriate changes to the system's /etc/lp directory.
- Enable or disable serial port services. admintool sets up the software services necessary to use a modem or terminal attached to a system's serial port.
- Add or remove software. admintool adds software from a product CD or on a hard disk to an installed system or removes software from an installed system.

Exit Status

admintool terminates with exit status 0.

Attributes

See `attributes`(5) for descriptions of the following attributes.

Attribute Type	**Attribute Value**
Availability	SUNWadmap

See Also

`group(4), hosts(4), passwd(4), attributes(5)`

New!

afbconfig, SUNWafb_config — Configure the AFB Graphics Accelerator

Synopsis

```
/usr/sbin/afbconfig [-dev device-filename][-res video-mode [now | try]
  [noconfirm | nocheck]][-file machine | system][-deflinear true |
  false][-defoverlay true | false][-overlayorder first | last]
  [expvis enable | disable] [-sov enable | disable] [-maxwinds n]
  [-extovl enable | disable] [-g gamma-correction-value][-gfile
  gamma-correction-file][-linearorder first | last][-propt][-prconf]
  [-defaults]
/usr/sbin/afbconfig [-propt] [-prconf]
/usr/sbin/afbconfig [-help] [-res ?]
```

Description

The `afbconfig` command is new in the Solaris 8 release. Use it to configure the AFB Graphics Accelerator and some of the X11 window system defaults for AFB.

The following form of `afbconfig` stores the specified options in the `OWconfig` file.

```
/usr/sbin/afbconfig [-dev device-filename][-res video-mode [now | try]
    [noconfirm | nocheck]][-file machine | system][-deflinear true |
    false][-defoverlay true | false][-overlayorder first | last][expvis
    enable | disable] [-sov enable | disable] [-maxwinds n] [-extovl
    enable | disable] [-g gamma-correction-value][-gfile
    gamma-correction-file][-propt][-prconf][-defaults]
```

Use the options to initialize the AFB device the next time the window system is run on that device. When you update options in the `OWconfig` file, the settings persist across window system sessions and system reboots.

The following forms of the `afbconfig` command invoke only the `-prconf`, `-propt`, `-help`, and `-res ?` options. None of these options update the `OWconfig` file.

```
/usr/sbin/afbconfig [-propt] [-prconf]
/usr/sbin/afbconfig [-help] [-res ?]
```

Additionally, the following invocation of `afbconfig` ignores all other options.

```
/usr/sbin/afbconfig [-help] [-res ?]
```

You can specify options for only one AFB device at a time. To specify options for multiple AFB devices, invoke the afbconfig command for each device you want to configure.

You can specify only AFB-specific options for afbconfig. You still specify the normal window system options for default depth, visual class, and so forth as device modifiers on the openwin command line.

You can also specify the OWconfig file that is to be updated. By default, the machine-specific file in the /etc/openwin directory tree is updated. You can use the -file option to specify an alternate file to use. For example, you can update the system-global OWconfig file in the /usr/openwin directory tree instead.

Both of the standard OWconfig files can be written only by root. Consequently, the afbconfig program, which is owned by the root user, always runs with setuid root permission.

Option Defaults

If you do not specify an option on the command line, the corresponding OWconfig option is not updated; it retains its previous value. When the window system is run, if an AFB option has never been specified by way of afbconfig, a default value is used. The option defaults are shown below.

-dev	/dev/fbs/afb0
-file	machine
-res	none
-deflinear	false
-defoverlay	false
-linearorder	last
-overlayorder	last
-expvis	enabled
-sov	enabled
-maxwids	32
-extovl	enabled
-g	2.22

The default of none for the -res option means that, when the window system is run, the screen resolution is the video mode currently programmed in the device.

This default provides compatibility for users who are accustomed to specifying the device resolution through the PROM. On some devices (for example, GX) the PROM is the only way of specifying the video mode, which means that the PROM ultimately determines the default AFB video mode.

Options

-defaults Reset all option values to their default values.

-deflinear true | false

AFB possesses two types of visuals: linear and nonlinear. Linear visuals are gamma corrected and nonlinear visuals are not. Two visuals have both linear and nonlinear versions: 24-bit TrueColor and 8-bit StaticGray.

If `true`, the default visual is set to the linear visual that satisfies other specified default visual selection options (specifically, the `Xsun`(1) `defdepth` and `defclass` options described in the *OpenWindows Reference Manual*).

If `false` or if no linear visual satisfies the other default visual selection options, the nonlinear visual specified by these other options are chosen as the default. You cannot use this option when the `-defoverlay` option is present because AFB doesn't possess a linear overlay visual.

`-defoverlay true | false`

The AFB provides an 8-bit PseudoColor visual whose pixels are disjoint from the rest of the AFB visuals. This visual is called the overlay visual. Windows created in this visual do not damage windows created in other visuals. The converse, however, is not true. Windows created in other visuals damage overlay windows.

The number of colors available to the windows created with this visual depends on the settings for the `-extovl` option. If the `-extovl` is enabled, extended overlay with 256 opaque color values is available. See `-extovl`. If `-extovl` is disabled, extended overlay is not available and the visual has 256 `-maxwids` number of opaque color values. See `-maxwids`.

If `-defoverlay` is `true`, the overlay visual is made the default visual. If `-defoverlay` is `false`, the nonoverlay visual that satisfies the other default visual selection options, such as `def`, `depth`, and `defclass`, are chosen as the default visual. See the *OpenWindows Reference Manual*.

Whenever the `defoverlay true` option is used, the default depth and class specified on the `openwin` command line must be 8-bit PseudoColor. If not, a warning message is printed and the `-defoverlay` option is treated as `false`.

You cannot use the `-defoverlay` option when you specify the `-deflinear` option because AFB doesn't possess a linear overlay visual.

`-dev` *`device-filename`*

Specify the AFB special file. The default is `/dev/fbs/afb0`.

`-expvis enable | disable`

If enabled, activate OpenGL Visual Expansion. Multiple instances of selected visual groups (8-bit PseudoColor, 24-bit TrueColor, and so forth) are in the screen visual list.

`-extovl enable | disable`

If enabled, make extended overlay available. The overlay visuals have 256 opaque colors. The SOV visuals have 255 opaque colors and 1 transparent color.

This option also enables hardware-supported transparency, thus provides better performance for windows using the SOV visuals.

`-file machine | system`

Specify which `OWconfig` file to update. If you specify `machine`, use the machine-specific `OWconfig` file in the `/etc/openwin` directory tree. If you specify `system`, use the global `OWconfig` file in the `/usr/openwin` directory tree. If the specified file does not exist, create it.

`-g` *`gamma-correction-value`*

Change the gamma correction value. All linear visuals provide gamma correction. By default, *`gamma-correction-value`* is `2.22`. Any value less than `0` is illegal. Apply the gamma correction value to the linear visual, which then has an effective gamma value of `1.0`, which is the value returned by `XSolarisGetVisualGamma`(3). See `XSolarisGetVisualGamma`(3) for a description of that function.

This option can be used while the window system is running. Changing the gamma correction value affects all the windows being displayed using the linear visuals.

`-gfile` *`gamma-correction-file`*

Load the gamma correction table from the specified *`gamma-correction-file`* file. You should format this file to provide 256 gamma correction values for R, G, and B channels on each line. Each of these values should be in hexadecimal format and separated by at least one space.

An example of a *`gamma-correction-file`* follows.

```
0x00 0x00 0x00
0x01 0x01 0x01
0x02 0x02 0x02
...
...
0xff 0xff 0xff
```

Using this option, you can load the gamma correction table while the window system is running. The new gamma correction affects all the windows being displayed using the linear visuals. When gamma correction is being done with a user-specified table, the gamma correction value is undefined. By default, the window system assumes a gamma correction value of `2.22` and loads the gamma table it creates corresponding to this value.

`-help` Print a list of the `afbconfig` command-line options, along with a brief explanation of each.

`-linearorder first | last`

If `first`, linear visuals come before their nonlinear counterparts on the X11 screen visual list for the AFB screen. If `last`, the nonlinear visuals come before the linear ones.

`-maxwids` *n* Specify the maximum number of AFB X channel pixel values that are reserved for use as window ID s (WIDs). The remainder of the pixel values in overlay colormaps are used for normal X11 opaque color pixels. The reserved WIDs are allocated on a first-come first-serve basis by 3D graphics windows (such as XGL), MBX windows, and windows that have a nondefault visual. The X channel codes 0 to (`255 -` *n*) are opaque color pixels. The X channel codes (`255 - n + 1`) to `255` are reserved for use as WIDs. Legal values are `1`, `2`, `4`, `8`, `16`, `32`, and `64`.

This option is available only if `-extovl` is disabled.

`-overlayorder first | last`

If `first`, the depth 8 PseudoColor Overlay visual comes before the non-overlay visual on the X11 screen visual list for the AFB screen. If `last`, the non-overlay visual comes before the overlay one.

`-propt` Print the current values of all AFB options in the `OWconfig` file specified by the `-file` option for the device specified by the `-dev` option. Print the values of options as they will be in the `OWconfig` file after the call to `afbconfig` completes.

The following display is typical.

```
--- OpenWindows Configuration for /dev/fbs/afb0 ---
OWconfig: machine
Video Mode: 1280x1024x76
Default Visual: Non-Linear Normal Visual
Visual Ordering: Linear Visuals are last
Overlay Visuals are last
OpenGL Visual Expansion: enabled
Server Overlay Visuals: enabled
Extended Overlay: enabled
Underlay WIDs: 64 (not configurable)
Overlay WIDs: 4 (not configurable)
Gamma Correction Value: 2.220
Gamma Correction Table: Available
```

`-prconf` Print the AFB hardware configuration.

The following display is typical.

```
--- Hardware Configuration for /dev/fbs/afb0 ---
Type: double-buffered AFB with Z-buffer
Board: rev 0 (Horizontal)
Number of Floats: 6
PROM Information: @(#)afb.fth x.xx xx/xx/xx
AFB ID: 0x101df06d
```

```
DAC: Brooktree 9070, version 1 (Pac2)
3DRAM: Mitsubishi 130a, version x
EDID Data: Available - EDID version 1 revision x
Monitor Sense ID: 4  (Sun 37x29cm RGB color monitor)
Monitorpossibleresolutions:1024x768x77,1024x800x84,1
1152x900x76, 1280x1024x67, 1280x1024x76, 960x680xx108s
Current resolution setting: 1280x1024x76
```

`-sov enable | disable`

If enabled, advertise the root window's SERVER_OVERLAY_VISUALS property. Export SOV visuals and this property can retrieve their transparent types, values, and layers. If disabled, the SERVER_OVERLAY_VISUALS property is not defined and SOV visuals are not exported.

`-res` *`video-mode`* `[now | try [noconfirm | nocheck]]`

Specify the video mode used to drive the monitor connected to the specified AFB device.

The format of these built-in video modes is *`width`*`x`*`height`*`x`*`rate`*, where *`width`* is the screen width in pixels, *`height`* is the screen height in pixels, and *`rate`* is the vertical frequency of the screen refresh.

The `s` suffix of `960x680x112s` and `960x680x108s` signifies stereo video modes. The `i` suffix of `640x480x60i` and `768x575x50i` signifies interlaced video timing. If absent, noninterlaced timing is used.

As a convenience, `-res` also accepts formats with an at sign (@) in front of the refresh rate instead of *n*, (1280x1024@76). Some *`video-mode`*s supported by AFB may not be supported by the monitor. You can display the list of *`video-mode`*s supported by the AFB device and the monitor by running `afbconfig` with the `-res ?` option.

The following list shows all possible video-modes supported on AFB.

```
1024x768x60
1024x768x70
1024x768x75
1024x768x77
1024x800x84
1152x900x66
1152x900x76
1280x800x76
1280x1024x60
1280x1024x67
1280x1024x76
960x680x112s    (Stereo)
960x680x108s    (Stereo)
640x480x60
640x480x60i     (Interlaced)
768x575x50i     (Interlaced)
```

For convenience, some of the *video-mode*s supported on the AFB have symbolic names. Instead of the form *width*x*height*x*rate*, you can supply one of these names as the argument to the -res option. The symbolic name none means that when the window system is run, the screen resolution is the video mode that is currently programmed in the device.

The following list shows the symbolic names for video modes that are supported on AFB.

Name	Corresponding Video Mode
svga	1024x768x60
1152	1152x900x76
1280	1280x1024x76
stereo	960x680x112s
ntsc	640x480x60i
pal	768x575x50i
none	Use the video mode currently programmed in the device

The -res option also accepts additional, optional arguments immediately following the video mode specification. You can specify any or all of the following arguments.

noconfirm
Bypass confirmation messages and program the requested video mode anyway. This option is useful when afbconfig is run from a shell script.

Because it is possible to put the system into an unusable state with the -res option if there is any ambiguity in the monitor sense codes, afbconfig, by default, prints a warning message and prompts to find out if it is OK to continue.

nocheck
Suspend the normal error checking based on the monitor sense code. Accept the video mode specified by the user regardless of whether it is appropriate for the currently attached monitor. (This option is useful if a different monitor is to be connected to the AFB device.) Use of this option implies noconfirm as well.

now
Update the video mode in the OWconfig file, and immediately program the AFB device to display this video mode. This option is useful for changing the video mode before starting the window system.

It is not advisable to use this argument with afbconfig while the configured device is being used (for example, while running the window system); unpredictable results may occur. To run afbconfig with the now argument, first bring the window system down. If you use the now argument within a window system session, the video mode is changed immediately, but the width and height of the affected screen don't change until the window system is exited and reentered. In addition, the system may not recognize changes in stereo mode. Consequently, this usage is strongly discouraged.

`try` Program the specified video mode on a trial basis. You are asked to confirm the video mode by typing `y` within 10 seconds. Or, you can terminate the trial before 10 seconds are up by typing any character. Any character other than `y` or Return is considered a no. The previous video mode is restored, and `afbconfig` does not change the video mode in the `OWconfig` file (other options specified still take effect). If you press Return, you are prompted for a `yes` or `no` answer on whether to keep the new video mode. This option implies the `now` argument (see the warning note on the `now` argument).

Examples

The following example switches the monitor type to a resolution of 1280 x1024 at 76 Hz.

```
example% /usr/sbin/afbconfig -res 1280x1024x76
```

Attributes

See `attributes`(5) for descriptions of the following attributes.

Attribute Type	**Attribute Value**
Availability	`SUNWafbcf`

See Also

`mmap(2), attributes(5)`
OpenWindows Reference Manual

aliasadm — Manipulate the NIS+ Aliases Map

Synopsis

```
/usr/bin/aliasadm -a alias expansion [options comments][optional-flags]
/usr/bin/aliasadm -c alias expansion [options comments][optional-flags]
/usr/bin/aliasadm -d alias [optional-flags]
/usr/bin/aliasadm -e alias [optional-flags]
/usr/bin/aliasadm -l alias [optional-flags]
/usr/bin/aliasadm -m alias [optional-flags]
/usr/bin/aliasadm [-I][-D domainname][-f filename][-M mapname]
```

Description

Use the `aliasadm` command to create, modify, and delete aliases in the NIS+ alias map. Mail aliases must be unique within the domain. To use the `aliasadm` command, you must be root, a member of the NIS+ group that owns the `Aliases` database, or the person who created the database. Alternatively, if you have Solstice AdminSuite, you can use the Database Manager to edit the `aliases` database.

The alias map is an NIS+ table object with four columns.

`alias` The name of the alias as a null-terminated string.

`expansion` The value of the alias as it would appear in a `sendmail/etc/aliases` file.

`options` A list of options applicable to this alias. The only option currently supported is CANON. With this option, if the user has requested an inverse alias lookup and there is more than one alias with this expansion, this alias is given preference.

`comments` An arbitrary string containing comments about this alias. The `sendmail`(1M) command reads this map in addition to the NIS aliases map and the local `/etc/aliases` database.

Options

`-a` Add an alias.

`-c` Change an alias.

`-d` Delete an alias.

`-D` *domainname*

Edit the map in domain *domainname* instead of the current domain.

`-e` Edit the alias map.

`-f` *filename* When editing or listing the database, use *filename* instead of invoking the editor.

`-I` Initialize the NIS+ aliases database.

`-l` List the alias map.

-m Print or match an alias.

-M *mapname* Edit *mapname* instead of *mail_aliases*.

Examples

The following example lists the contents of the aliases table for a newly created NIS+ server in alphabetical order by alias.

```
# aliasadm -l
paperbark# # aliasadm -l
MAILER-DAEMON: postmaster
Postmaster: root
nobody: /dev/null
#
```

If you have a large aliases table, listing the entire contents can take a while. You can pipe the output through `grep` if you are searching for a specific entry.

The following example lists an individual entry in the NIS+ `mail_aliases` table.

```
# aliasadm -m ignatz
ignatz: ignatz@castle  # Alias for Iggy Ignatz
#
```

The `aliasadm -m` option matches only the complete alias name. If you want partial matches, pipe the `aliasadm -l` command through `grep`.

Files

`/etc/aliases` Mail aliases for the local host in ASCII format.

Attributes

See `attributes`(5) for descriptions of the following attributes.

Attribute Type	**Attribute Value**
Availability	SUNWnisu

See Also

`sendmail(1M), attributes(5)`

allocate — Device Allocation

Synopsis

```
/usr/sbin/allocate [-s][-U uname] device
/usr/sbin/allocate [-s][-U uname] -g dev-type
/usr/sbin/allocate [-s][-U uname] -F device
```

Description

Use the `allocate` command to manage the ownership of devices. `allocate` ensures that each device is used by only one qualified user at a time.

Use the *`device`* argument to specify the device to be manipulated. To preserve the integrity of the device's owner, the allocate operation is executed on all the device special files associated with that device.

Use the *`dev-type`* argument to specify the device type to be operated on. You can use the *`dev-type`* argument only with the `-g` option.

The default allocate operation allocates the device special files associated with device to the UID of the current process.

If you specify the `-F` option, the device cleaning program is executed when allocation is performed. This cleaning program is found in `/etc/security/lib`. The name of this program is found in the `device_allocate`(4) entry for the device in the `dev-exec` field.

Note — The functionality described in this manual page is available only if the Basic Security Module (BSM) has been enabled. See `bsmconv`(1M) for more information.

Options

`-F` *`device`* Reallocate the device allocated to another user. This option is often used with `-U` to reallocate a specific device to a specific user. Only superuser is permitted to use this option.

`-g` *`dev-type`* Allocate a nonallocated device with a device type matching *`dev-type`*.

`-s` Suppress any diagnostic output.

`-U` *`uname`* Use the user ID *`uname`* instead of the user ID of the current process when performing the allocate operation. Only superuser is permitted to use this option.

Exit Status

`allocate` returns a non-zero exit status in the event of an error.

Files

`/etc/security/device_allocate`

File that contains mandatory access control information about each physical device.

`/etc/security/device_maps`

File that contains access control information about each physical device.

`/etc/security/dev/*`

Directory that contains security device files.

`/etc/security/lib/*`

Directory that contains security executables.

Attributes

See attributes(5) for descriptions of the following attributes.

Attribute Type	Attribute Value
Availability	SUNWcsu

See Also

bsmconv(1M), device_allocate(4), device_maps(4), attributes(5)

amiserv — AMI Keyserver

New!

Synopsis

```
/usr/lib/security/amiserv
```

Description

The Authentication Management Infrastructure (AMI) is new in the Solaris 8 release. AMI is a public-key-based authentication/privacy system that includes a set of security-related APIs and commands. You can use these APIs to tailor a public key infrastructure (PKI) suitable for your site's authentication and privacy requirements.

The AMI keyserver, amiserv, is a per-host daemon that performs cryptographic operations with private keys. The daemon is initiated at system startup by the /etc/rc2.d/S95amiserv run control script. The private key operations performed by the keyserver include digital signature and decryption operations.

amiserv also acts as a repository for user and host keystores. When users require access to their keystore, they retrieve it from amiserv and not from the (possibly insecure) naming service.

Users register their keystore with amiserv by executing the amilogin(1) command, which uses the entered password to cryptographically validate the keystore and decrypt the private key (see amikeystore(1)). The validated keystore is then sent to amiserv via a UNIX RPC call. The keystore remains with amiserv until the user runs the amilogout(1) command or amiserv is restarted. The amilogout command removes the user's keystore from amiserv.

Note — The user's keystore is not automatically removed from amiserv when a user runs logout(1).

amiserv stores all keystores that have been registered with it in memory in an obscured form. Keystores that have been permanently registered with amiserv (by the amilogin command with the -p option) are also stored internally on the local file system. This file is read by amiserv on startup to initialize itself with a set of permanent host and user keystores.

Note — You must be root to run amiserv.

Exit Status

0	Successful completion.
1	An error occurred.

Attributes

See attributes(5) for descriptions of the following attributes.

Attribute Type	**Attribute Value**
Availability	SUNWami

See Also

amikeystore(1), amilogin(1), amilogout(1), attributes(5)

answerbook2_admin — AnswerBook2 GUI Administration Tool

Synopsis

```
/usr/dt/bin/answerbook2_admin [-h]
```

Description

Use the answerbook2_admin command to open the default Web browser showing the administration interface for the local AnswerBook2 server. The AnswerBook2 administration tool based on the Web browser provides the same functionality as the ab2admin(1M) command-line administration tool.

This functionality is also accessible through the AnswerBook2 Admin option within the System_Admin subset of the Application Manager function on the CDE front panel Applications menu.

> **Note —** Once a Web browser is open and you have access to the AnswerBook2 Administration tool, use its online Help system to find out more about administering the AnswerBook2 server.

Options

-h	Display a usage statement.

Usage

At startup time, answerbook2_admin starts up the default Web browser (for example, Netscape or HotJava) and displays the URL specified for administering the local AnswerBook2 server (http://localhost:8888). If the user has set up administration access control, the Web browser prompts for a valid administrator login and password for this document server before displaying the administration tool.

Files

`/usr/lib/ab2/dweb/data/config/admin_passwd`
: File containing *username*:*password*.

Attributes

See `attributes`(5) for descriptions of the following attributes.

Attribute Type	Attribute Value
Availability	SUNWab2m

See Also

`ab2admin(1M), attributes(5)`

arp — Address Resolution Display and Control

Synopsis

```
/usr/sbin/arp hostname
/usr/sbin/arp -a
/usr/sbin/arp -d hostname
/usr/sbin/arp -f filename
/usr/sbin/arp -s hostname ether-address [temp][pub][trail]
```

Description

Use the `arp` command to display and modify the Internet-to-Ethernet address translation tables used by the address resolution protocol (see `arp`(7P)). You can specify *hostname* either by name or by number, using Internet dot notation.

`arp` is useful as a diagnostic tool to resolve point-to-point connectivity problems. If the ARP table of a host system contains an incorrect entry, the system is unreachable because outgoing packets contain the wrong Ethernet address.

Options

`-a`
: Display all of the current ARP entries. The following flags are used in the table.

`P`	Publish; include IP address for the machine and the addresses that have explicitly been added with the `-s` option. ARP responds to ARP requests for this address.
`S`	Static; not learned for the ARP protocol.
`U`	Unresolved; waiting for ARP response.
`M`	Mapping; used only for the multicast entry for 224.0.0.0

`-d` Delete an entry for the host called *hostname*. Only superuser can use this option.

`-f` Read the file named *filename* and set multiple entries in the ARP tables. Entries in the file should be of the following form.

```
hostname ether-address [temp][pub][trail]
```

(See option `-s` for argument definitions.)

`-s` Create an ARP entry for the host called *hostname* with the Ethernet address *ether-address*. Specify the Ethernet address as six hexadecimal bytes separated by colons. The entry is permanent unless you specify the word `temp` in the command. If you specify the word `pub`, the entry is published. For example, this system responds to ARP requests for *hostname* even though the hostname is not its own. The word `trail` indicates that trailer encapsulations can be sent to this host. You can use `arp - s` for a limited form of proxy ARP when a host on one of the directly attached networks is not physically present on the subnet. Another system can then be configured with `arp -s` to respond to ARP requests. This option is useful in certain SLIP or PPP configurations.

Operands

hostname Specify the host either by name or by number, using Internet dot notation.

ether-address

Specify the Ethernet address as six hexadecimal bytes separated by colons.

Examples

The following example uses `arp -a` to display the ARP table.

```
# arp -a

Net to Media Table: IPv4
Device   IP Address               Mask      Flags   Phys Addr
------ -------------------- --------------- ----- ---------------
hme0   G3                   255.255.255.255       00:05:02:35:aa:1c
hme0   castle               255.255.255.255       08:00:20:18:69:71
hme0   paperbark            255.255.255.255 SP    08:00:20:7d:79:d4
hme0   224.0.0.0            240.0.0.0       SM    01:00:5e:00:00:00
#
```

The `S` flag means static, the `P` flag means publish, the `M` flag means mapping.

The following example lists the ARP entry for the system `paperbark`.

```
# arp paperbark
paperbark (172.16.8.22) at 8:0:20:7d:79:d4 permanent published
#
```

The following example deletes the ARP entry for the system paperbark.

```
# arp -d paperbark
paperbark (172.16.8.22) deleted
#
```

The following example recreates the paperbark entry in the ARP table as permanent published.

```
# arp -s paperbark 8:0:20:7d:79:d4 pub
# arp paperbark
paperbark (172.16.8.22) at 8:0:20:7d:79:d4 permanent published
#
```

Attributes

See attributes(5) for descriptions of the following attributes.

Attribute Type	Attribute Value
Availability	SUNWcsu

See Also

ifconfig(1M), attributes(5), arp(7P)

aset — Monitor or Restrict Access to System Files and Directories

Synopsis

```
/usr/aset/aset [-p][-d aset-dir][-l sec-level][-n user@host]
  [-u userlist-file]
```

Description

The Automated Security Enhancement Tool (ASET) is a set of administrative commands that enable you to monitor and control system security by automatically performing tasks that you would otherwise do manually. You can use it to check the settings of system files, including attributes such as permissions and ownership and the contents of the system files. It warns the users of potential security problems and, where appropriate, sets the system files automatically according to the specified security level.

You can set ASET to operate at one of three security levels either by specifying the -l option at the command line or by setting the ASETSECLEVEL environment variable to low, med, or high. All the functionality operates according to the value of the security level.

- At the low level, ASET ensures that attributes of system files are set to standard release values. ASET performs several checks and reports potential security weaknesses. At this level, ASET takes no action and does not affect system services.

- At the med level, ASET provides adequate security control for most environments. ASET modifies some of the system file settings and parameters, restricting system access to reduce the risks from security attacks. ASET reports security weaknesses and any modifications it makes to restrict access. At this level, ASET does not affect system services, and all of the system applications and commands maintain all of their original functionality.
- At the high level, ASET provides a highly secure system. ASET adjusts many system files and parameter settings to minimize access permissions. Most system applications and commands continue to function normally, but at this level, security considerations take precedence over other system behavior. The majority of system applications and commands maintain their functionality, although there may be a few that exhibit behaviors that are not familiar in normal system environment.

Refer to the *System Administrator's Manual, Volume I* for more exact definitions of what aset does at each of these levels. The asetenv(4) file and the master files (see asetmasters(4)) determine to a large extent what aset performs at each level, and experienced administrators can use these files to redefine the definitions of the levels to suit their particular needs. These files are provided by default to fit most security-conscious environments, and, in most cases, the files provide adequate security safeguards without modification. They are, however, designed so that they can be easily edited by experienced administrators with specific needs.

You can periodically activate aset at the specified security level with default definitions by using the -p option. aset is automatically activated at a frequency specified by the administrator, starting from a designated future time (see asetenv(4)). Without the -p option, aset operates only once, immediately.

Options

-d *aset-dir* Specify a working directory other than /usr/aset for ASET. /usr/aset is the default working directory where ASET is installed, and it is the root directory of all ASET commands and data files. If you use another directory as the ASET working directory, you can define it either with the -d option or by setting the ASETDIR environment variable before invoking aset. If you specify the command-line option, it overwrites the environment variable.

-l *sec-level* Specify a security level (low, med, or high). The default level is low. You can also specify the level by setting the ASETSECLEVEL environment variable before invoking aset. If you specify the command-line option, it overwrites the environment variable.

-n *user@host* Notify *user* at system *host*. Send the output of aset to *user* by e-mail. If you do not specify this option, send the output to the standard output. Note that this output is not the reports of ASET, but instead is an execution log that includes any error messages. This output is typically fairly brief. The actual reports of ASET are found in the /usr/aset/reports/latest directory. See the -d option.

`-p` Schedule `aset` to be executed periodically by adding an entry for `aset` in the `/etc/crontab` file. Use the `PERIODIC_SCHEDULE` environment variable in the `/usr/aset/asetenv` file to define the time for execution. See `crontab`(1) and `asetenv`(4). If a `crontab` (1) entry for `aset` already exists, produce a warning in the execution log.

`-u` *`userlist-file`*

Specify a file containing a list of users. `aset` performs environment checks (for example, `UMASK` and `PATH` variables) on these users. By default, `aset` checks only for root. *`userlist-file`* is an ASCII text file. Each entry in the file is a line that contains only one user name (login name).

Usage

The following paragraphs discuss the tasks ASET performs. Execute the first task, `tune`, only once per installation of ASET. Execute the other tasks periodically at the specified frequency.

tune Task

Use the `tune` task to tighten system file permissions. ASET provides three configurable files—`tune.low`, `tune.med`, and `tune.high`—that define the available ASET security levels. In standard releases, system files or directories have permissions defined to maximize open information sharing. In a more security-conscious environment, you may want to redefine these permission settings to more restrictive values. `aset` enables you to reset these permissions, based on the specified security level. Generally, at the low level, the permissions are set to what they should be as released. At the medium level the permissions are tightened to ensure reasonable security that is adequate for most environments. At the high level, they are further tightened to very restrictive access. See `asetmasters`(4).

cklist Task

Use the `cklist` task to examine system files and to compare each one with a description of that file listed in a master file. The `/usr/aset/masters/cklist.level` is created the first time ASET runs the `cklist` task. See `asetenv`(4). Any discrepancies found are reported in the `cklist.rpt` file. The following information is compared for directories and files.

- Owner and group.
- Permission bits.
- Size and checksum (if file).
- Number of links.
- Last modification time.

The lists of directories to check are defined in `asetenv`(4), based on the specified security level, and are configurable with the `CKLISTPATH_LOW`, `CKLISTPATH_MED`, and `CKLISTPATH_HIGH` environment variables. Typically, the lower-level lists are subsets of the higher-level lists.

usrgrp Task

Use the usrgrp task to check the consistency and integrity of user accounts and groups as defined in the passwd and group databases and to report any potential problems in the usergrp.rpt file. This task checks for the following violations.

- passwd file entries not in the correct format.
- User accounts without a password.
- Duplicate user names.
- Duplicate user IDs. Duplicate user IDs are reported unless allowed by the uid_alias file. See asetmasters(4)).
- Invalid login directories.
- If C2 is enabled, check C2 hidden passwd format.

Potential problems for the group file include the following.

- Group file entries not in the right format.
- Duplicate group names.
- Duplicate group IDs.
- Null group passwords.

aset checks the local passwd file. If the YPCHECK environment variable is set to true, aset also checks the NIS passwd files. See asetenv(4). Problems in the NIS passwd file are reported only and are not corrected automatically. The checking is done for all three security levels except where noted.

sysconf Task

Use the sysconf task to check various system configuration tables, most of which are in the /etc directory. aset checks and makes appropriate corrections for each system table at all three levels, except where noted. All problems are reported in the sysconf.rpt file. The following discussion assumes familiarity with the various system tables. See the manual pages for these tables for further details.

The operations for each system table are shown below.

/etc/hosts.equiv

The default file contains a single + line, thus making every known host a trusted host: not advised for system security. aset performs the following operations.

Low	Warn administrators about the + line.
Medium	Warn administrators about the + line.
High	Warn about and delete the + entry.

/etc/inetd.conf

Check the following entries for system daemons for possible weaknesses.

tftp(1) does not do any authentication. Ensure that in.tftpd(1M) is started in the right directory on the server and is not running on clients. At the low level, aset warns if the mentioned condition is not true. At the medium and high levels, aset warns, and changes (if necessary), the in.tftpd entry to include the -s /tftpboot option after ensuring the directory /tftpboot exists.

ps(1) and netstat(1M) provide valuable information to potential system crackers. Disable these commands when aset is executed at a high security level.

rexd is known to have poor authentication mechanism. Disable rexd for medium and high security levels by commenting out the entry. If rexd is activated with the -s (secure RPC) option, do not disable it.

/etc/aliases The decode alias of UUCP is a potential security weakness. Disable the alias for medium and high security levels by commenting out the entry.

/etc/default/login

Check the CONSOLE= line to enable root login only at a specific terminal depending on the security level.

Low	Take no action.
Medium	Take no action.
High	Add the following line to the file. CONSOLE=/dev/console

/etc/vfstab Check for world-readable or writeable device files for mounted file systems.

/etc/dfs/dfstab

Check for file systems that are exported without any restrictions.

/etc/ftpusersAt high security level, ensure that root is in /etc/ftpusers (and create the entry if necessary), thus disallowing ftp(1) to be used as root.

/var/adm/utmpx

New!

Make this files not world-writeable for the high level. (Some applications may not run properly with this setting.) Note that the utmpx database file in Solaris 8 supersedes the obsolete utmp database file of previous releases.

/.rhosts The use of an .rhosts file for the entire system is not advised. Warn for the low level and move the file to /.rhosts.bak for levels medium and high.

env task

The env task checks how the PATH and UMASK environment variables are set for root and users specified with the -u *userlist-file* option by parsing the /.profile, /.login, and /.cshrc files. It checks the PATH variable to ensure that it does not contain . as a directory, which makes an easy target for Trojan horse attacks. It also checks that the directories in the PATH variable are not world-writeable. It checks the UMASK variable to ensure files are not created as readable or writeable by world. Any problems found by these checks are reported in the env.rpt file.

eeprom task

Newer versions of the EEPROM enable you to specify a `secure` parameter. See `eeprom`(1M). `aset` recommends that the administrator set the parameter to `command` for the medium level and to `full` for the high level. `aset` gives warnings in the `eeprom.rpt` file if it detects the parameter is not set adequately.

firewall task

The `firewall` task ensures that the system can be safely used as a network relay. At the high security level, `aset` takes proper measures so that the system can be safely used as a firewall in a network. These measures mainly involve disabling IP packet forwarding and making routing information invisible. Firewalling provides protection against external access to the network. Any changes made by this task are reported in the `firewall.rpt` file.

Environment Variables

`ASETDIR` Specify the ASET working directory. Default is `/usr/aset`.

`ASETSECLEVEL` Specify the ASET security level. Default is `low`.

`TASKS` Specify the tasks to be executed by `aset`. Default is all tasks.

Files

`/usr/aset/reports`

Directory of ASET reports.

Attributes

See `attributes`(5) for descriptions of the following attributes.

Attribute Type	**Attribute Value**
Availability	`SUNWast`

See Also

`crontab(1), ftp(1), ps(1), tftp(1), eeprom(1M), in.tftpd(1M), netstat(1M), asetenv(4), asetmasters(4), attributes(5)`

System Administration Guide, Volume I

aset.restore — Restore System Files Affected by ASET

Synopsis

`/usr/aset/aset.restore [-d` *`aset-dir`*`]`

Description

Use the `aset.restore` command to restore system files that are affected by the Automated Security Enhancement Tool (ASET) to their pre-ASET content. When ASET is executed for the first time, it saves and archives the original system files in the `/usr/aset/archives` directory. The `aset.restore` command reinstates these files. It also deschedules ASET if it is currently scheduled for periodic execution. See `asetenv`(4).

If you have made changes to system files after running ASET, these changes are lost when you run `aset.restore`. If you want to be absolutely sure that you keep the existing system state, back up your system before using `aset.restore`.

Use `aset.restore` under the following circumstances.

- You want to remove ASET permanently and restore the original system (if you want to deactivate ASET, you can remove it from scheduling).
- You are unfamiliar with ASET and want to experiment with it. You can use `aset.restore` to restore the original system state.
- When some major system functionality is not working properly and you suspect that ASET is causing the problem; you may want to restore the system to see if the problem persists without ASET.

`aset.restore` requires root privileges to execute.

Options

`-d` *`aset-dir`* Specify that the working directory for ASET is located under *`aset-dir`*. By default, this directory is `/usr/aset`.

Files

`/usr/aset/archives`

Archive of system files before `aset` is executed.

Attributes

See `attributes`(5) for descriptions of the following attributes.

Attribute Type	**Attribute Value**
Availability	`SUNWast`

See Also

`aset(1M), asetenv(4), attributes(5)`

System Administration Guide, Volume I

aspppd, aspppls — Asynchronous PPP Link Manager

Synopsis

```
/usr/sbin/aspppd [-d debug-level]
/usr/sbin/aspppls
```

Description

The aspppd link manager is a user-level daemon that automates the process of connecting to a remote host for Point-to-Point Protocol (PPP) service. It works with the IP-Dialup driver (ipdcm) and PPP streams module (ppp(7M)) to provide IP network services over an analog modem using dialed, voice-grade telephone lines. This automated process starts whenever any activity that generates IP traffic takes place (for example, a user logging in to a remote system). When a remote host tries to establish a connection, the link manager on the local host completes the connection.

aspppd can initiate the connection process either by sending an IP datagram to a (disconnected) peer host or by receiving a notification that a peer host wants to establish a connection.

The aspppls command is invoked as a login shell that starts PPP after you dial up and log in. Its function is similar to the /usr/lib/uucp/uccico command. When you configure a system as a dial-in server, you must specify aspppls as the login shell in the /etc/passwd file in the entries for every nomadic computer allowed to dial in to the local host.

aspppls is invoked by the serial port monitor when a peer machine logs into a PPP-enabled account. It enables the aspppd link manager to accept the incoming call.

Options

-d *debug-level*

Specify the debug level as a number between 0 and 9. Higher numbers give more detailed debugging information. Write the output to the log file /etc/log/asppp.log.

Usage

If the configuration file /etc/asppp.cf is present, the link manager is invoked at boot time. After parsing the configuration file and building a path object for each peer host, it sleeps until one of the following events occurs.

- An IP datagram is routed to one of the ipd or ipdptp interfaces (see ppp(7M)). The link manager consults the UUCP database, dials the modem, logs into the peer host, establishes the PPP data link, brings up IP, and forwards the IP datagram that initiated the process.
- It is notified by the login service that a peer host is trying to make a connection. The link manager opens the file descriptor supplied by the login service, establishes the PPP data link, and brings up IP.

If the link manager determines that there has been no IP traffic for the period specified by the inactivity_timeout keyword, it disconnects the link by bringing down IP and PPP and closing the connection to the peer host.

You can reinitialize the link manager by sending it the -HUP signal (with kill(1), for example), which disconnects all open PPP links and rereads the configuration file.

Path

A path object contains the state of a connection with a peer host. Information such as system names, interface names, timeout values, and other attributes are kept in the path object. A path exists for each potential peer host. You define paths in the configuration file.

Interfaces

The link manager supports two types of IP layer interfaces: the point-to-multipoint interface (ipd) and the point-to-point interface (ipdptp) (see ppp(7M)).

- The point-to-multipoint interface logically connects the host machine to a network containing one or more peer hosts. IP traffic to or from any of the peer hosts is routed through the point-to-multipoint interface. When an ipd interface is configured, only one IP address, that of the host, is assigned. In other words, it behaves similarly to an Ethernet interface, although the broadcast capability is not supported. This type of interface is well suited for a dial-in PPP server.
- The point-to-point interface logically connects the host machine with one peer host. Only IP traffic to or from the peer host is routed through this interface. When an ipdptp interface is configured, two IP addresses are assigned. This type of interface is well suited to support a remote or nomadic machine.

An interface must be fully configured and enabled (that is, up) before an IP datagram is routed to it. A point-to-multipoint interface must also be fully configured and enabled before the link manager can associate an incoming connection with it. You do not, however, need to configure and enable a point-to-point interface before an incoming connection is assigned to it. A point-to-point interface that is "plumbed," but otherwise not configured or enabled (that is, down), can be used to accept an incoming connection if the path associated with the potential connection contains a dynamic interface specification (for example, interface ipdptp*). In this case, the link manager selects a disabled (down) interface, configures the host and peer addresses, brings the interface up, and assigns it for the duration of the connection.

Routing

You should pay special attention to routing issues that can arise if a host has more than one interface configured and enabled. By definition, a host with more than one enabled interface is a router, and the routing daemon (typically, in.routed) advertises the routes provided by the PPP interfaces. This behavior is normally acceptable for a dial-in server but can cause network disruptions if not administered properly.

To prevent routing information packets (RIP) from flowing over point-to-point interfaces, specify the norip keyword followed by the interface name in the /etc/gateways file. The following entries, for example, prevent RIP from being sent over ipdptp0 and ipdptp1.

```
norip   ipdptp0
norip   ipdptp1
```

See in.routed(1M) for further information.

Authentication

You can configure the link manager to support either the Password Authentication Protocol (PAP) or the Challenge Handshake Authentication Protocol (CHAP) as specified in RFC1334. Both protocols can be configured simultaneously, in which case, CHAP has precedence. A single host can participate as an authenticator (the local host requests that the peer host authenticate itself) or an authenticatee (the local host has been asked by the peer host to authenticate itself) or as both. It is also possible for a host to be an authenticator for one protocol and an authenticatee for the other protocol.

- PAP is a simple protocol similar to a standard login/password type of authentication. The PAP authenticator sends a message to its peer, requesting that the peer authenticate itself. The peer responds with an authenticate request packet that contains an ID and a password (both in plaintext). The ID and password are matched against a local copy, and if they match, the connection is established. If they don't match, the connection is dropped.
- CHAP does not pass any plaintext authentication data across the link. The CHAP authenticator sends a challenge packet to the peer that contains a random string. The peer then takes the string in the challenge packet and computes a response string that is a function of the challenge string and a shared secret key. The peer then sends a response packet back to the authenticator. The authenticator computes a string based on the original challenge string and the shared secret key and matches that result with the received response. If they match, the connection is established. Otherwise, the connection is dropped.

Configuration File

The primary purpose of the `/etc/asppp.cf` configuration file is to define each path used by the link manager to establish and maintain communication with a peer system.

The file consists of a sequence of tokens separated by white space (blanks, Tabs, and newlines). No record boundaries or any other constraints are put on the placement of the tokens. If a token begins with a pound sign (#), all characters between the pound sign and the next newline (`\n`) are ignored (that is, they are treated as a comment). Alphanumeric tokens are case insensitive and are translated by the lexical analyzer into lower case before further processing.

A string is a single token that does not contain embedded white space. You can use the standard ANSI C \ escape sequence to embed special characters (see an ANSI C manual for a list of escaped special characters). Use `\s` for the space character. If a pound sign appears at the beginning of a string, you must escape it (\#) to prevent it from being interpreted as a comment. A null (`\0`) truncates the string.

Groups of tokens are assembled into units known as paths (essentially a human-readable form of the path object). A path begins with the keyword `path` and ends at the token found before any subsequent path (or defaults) keyword or at the last token in the file. The tokens that make up a path are further partitioned into small groups consisting mostly of keyword/value pairs that define the attributes of the current path. If a particular keyword/value pair is not listed for a path, the default value is assumed.

The token sequences that begin with the substrings `ipcp_` or `lcp_` refer to PPP initial configuration options as specified in RFC1332, *The PPP Internet Protocol Control Protocol (IPCP)*. See the RFC for a more complete definition of these options.

The following is an alphabetic list of the token sequences that can be contained in a configuration file. Required sequences are noted.

Keywords

`chap_name` *`string`*

Specify one or more octets representing the identification of this host. Do not terminate the name with null or CR/LF. Send the name to the authenticator in a response packet. Put this key/value pair in the authenticatee's configuration file.

`chap_peer_secret` *`string`*

Specify one or more octets, preferably at least 16, that contain the secret key that is used with the challenge value to generate the string to match with the response received from the peer. Put this key/value pair in the authenticator's configuration file.

`chap_peer_name` *`string`*

Specify one or more octets representing the identification of the peer transmitting the packet. Do not terminate the name with null or CR/LF. The name is received from the peer in a response packet. Put this key/value pair in the authenticator's configuration file.

`chap_secret` *`string`*

Specify one or more octets, preferably at least 16, that contain the secret key that is used with the received challenge value to generate the response sent to the authenticator. Put this key/value pair in the authenticatee's configuration file.

`debug_level` *`number`*

Specify a debug level. *`number`* is a value between 0 and 9. Higher numbers give more detailed debugging information. Write the output to the `/etc/log/asppp.log` file. The value set by the `debug_level` keyword overrides the `-d` command line option.

`0`	Errors only.
`1`	Minimal information.
`4`	Some UUCP chat-script information.
`5`	All UUCP chat-script information.
`7`	Maximum UUCP information.
`8`	PPP message traces.
`9`	Raw IP packets.

`defaults` Specify that all following token sequences up the next path keyword, or the end-of-file, set default attributes that affect subsequently defined paths.

`default_route`

When the IP layer corresponding to the current path is fully operational, add the peer IP address to the route table as the default destination. The route is removed when the IP layer is brought down.

Note that the `default_route` keyword is installed only by point-to-point interfaces.

`ifconfig` *`parameters`*

(Required) Pass the `ifconfig` keyword and associated parameters to the shell for evaluation and execution. Use it to define an interface. See the `ifconfig`(1M) manual page for more information.

`inactivity_timeout` *`seconds`*

Specify in *`seconds`* the maximum number of seconds that the connection associated with the current path can remain idle before it is terminated. You can specify `0` to indicate no timeout. The default is 120 seconds.

`interface (ipd`*`n`*` | ipdptp`*`n`*` | ipdptp*)`

(Required) Associate a specific point-to-multipoint or point-to-point interface as denoted by the nonnegative integer *`n`* with the current path. The third form, `ipdptp*`, indicates that the interface associated with the path is a dynamic interface that is selected at connect time from a pool of previously configured, inactive (down) point-to-point interfaces.

`ipcp_async_map` *`hex-number`*

Specify the async control character map for the current path. *`hex-number`* is the natural (that is, big-endian) form representation of the four octets that make up the map. The default value is `ffffffff`.

`ipcp_compression (vj | off)`

Indicate whether IP compression is enabled. If enabled (`vj`), use the Van Jacobson compression algorithm. The default is compression (`vj`).

`lcp_compression (on | off)`

Indicate whether PPP address, control, and protocol field compression are enabled. If enabled, both the address and control field compression and the protocol field compression options are set. The default is compression (`on`).

`lcp_mru` *`number`*

Specify a desired maximum receive unit packet size in octets. The default is `1500`.

`negotiate_address (on | off)`

Indicate whether local IP address assignment is obtained through negotiation and assigned dynamically. If enabled, the local address is obtained from the remote end of the PPP link. If so obtained, you can use any local address other than `0.0.0.0` to initially configure the interface. The default is to not negotiate (`off`).

`pap_id` *`string`*

Specify one or more octets that represent the host name that is sent to the authenticator. To indicate a zero length string, do not include the keyword. Put this key/value pair in the authenticatee's configuration file.

`pap_password` *`string`*

Specify one or more octets that indicate the password that is sent to the authenticator for this host. To indicate a zero-length string, do not include the keyword. Put this key/value pair in the authenticatee's configuration file.

`pap_peer_id` *`string`*

Specify one or more octets that indicate the name of the peer to be authenticated. To indicate a zero-length string, do not include the keyword. Put this key/value pair in the authenticator's configuration file.

`pap_peer_password` *`string`*

Specify one or more octets that indicate the password to be used for authentication. To indicate a zero-length string, do not include the keyword. Put this key/value pair in the authenticator's configuration file.

`path` (Required) Group all following token sequences as attributes of this (current) path. Terminate the collection of attributes that make up the current path by the occurrence of a subsequent `path` or `defaults` keyword or by the end-of-file.

`peer_ip_address` *`IP-address`*

(Required for point-to-multipoint paths) Associate the IP-address with the current path. Ignore the value if the path specifies a point-to-point interface. *`IP-address`* can be in dotted decimal, hexadecimal, or symbolic (that is, host-name) format.

`peer_system_name` *`name`*

(Required) Associate the peer system name with the current path. Use the name to look up modem and peer-specific information for outbound connections in the UUCP `/etc/uucp/Systems` file. For incoming connections, determine the appropriate path by matching *`name`* with the login name that was used to obtain the connection (that is, an entry in the `/etc/passwd` file specifies *`name`* in the `username` field).

`require_authentication (off | pap [chap] | chap [pap])`

Indicate that the local host is the authenticator and that the peer is required to authenticate itself. If you specify either `pap` or `chap`, the peer must participate in the authentication protocol or the connection is terminated. If you specify both `pap` and `chap`, then the local host tries to negotiate `chap`, and if that fails, the connection is terminated. The local host does not try to negotiate `pap`. The default does not require authentication (`off`).

If pap is required, then you should specify the pap_peer_id and pap_peer_password keywords and values for the associated path. If you do not specify them, set the corresponding values to the null string. If chap is required, then you must specify the chap_peer_name and chap_peer_secret keywords and values for the associated path.

version *n*
: Specify that the contents of the configuration file correspond to format version *n*. If this keyword is present, it must be the first keyword in the file. If absent, the version is assumed to be 1. This document contains the definition of the version 1 format for the configuration file.

will_do_authentication (off | pap [chap] | chap [pap])
: Indicate that the local host is a potential authenticatee and is willing to participate in the specified authentication protocol. If both pap and chap are present, then the local host is willing to participate in either authentication protocol. The default does not participate in authentication (off).

If pap is available, then you should specify the pap_id and pap_password keywords and values for the associated path. If they are not specified, the corresponding values are set to the null string. If chap is available, then you must specify the chap_name and chap_secret keywords and values for the associated path.

Examples

The following example shows a remote machine that is likely to be nomadic or a home machine with a single modem.

```
#
# Dial in to two servers
#
ifconfig ipdptp0 plumb nomad1 dialin1 private up
path
                  interface ipdptp0
                  peer_system_name Pdialin1
                  will_do_authentication pap
                  ap_id nomad1
                  pap_password secret
ifconfig ipdptp1 plumb nomad1 dialin2 private up
path
                  interface ipdptp1
                  peer_system_name Pdialin2
                  lcp_mru 1006
```

The following example shows a dial-in server supporting a point-to-multipoint interface. Several modems can be attached to this server. The network addressed by the ipd interface is advertised by the router, and all traffic destined for that network is routed through this host. For that reason, it is unwise to support multiple dial-in servers with point-to-multipoint interfaces to the same network.

```
#
# A point-to-multipoint dial in server
```

```
#
ifconfig ipd0 plumb dialin1 netmask + up
defaults
      interface ipd0
      inactivity_timeout 900 # 15 minutes
      require_authentication chap pap
      chap_peer_name nomads path
      peer_system_name Pnomad1
      chap_peer_secret abcd
      pap_peer_id nomad1
      pap_peer_password secret
      peer_ip_address nomad1
path
      peer_system_name Pnomad2
      chap_peer_secret a\sspace
      peer_ip_address nomad2
path
      peer_system_name Pnomad3
      inactivity_timeout 0 # No timeout for this host
      chap_peer_secret \#123;.
      peer_ip_address nomad3
path
      peer_system_name Pnomad4
      chap_peer_secret My\sSecret#Word
      peer_ip_address nomad4
```

The following example is for another dial-in server that supports dynamic point-to-point interfaces. Usually the server has one modem for each interface. One advantage of using dynamic interfaces is that (host) routes are advertised only when an interface is up. Therefore, multiple dial-in servers can be supported.

```
#
# A dynamic point-to-point dial in server
#
ifconfig ipdptp0 plumb dialin2 client1 down
ifconfig ipdptp1 plumb dialin2 client2 down
ifconfig ipdptp2 plumb dialin2 client3 down
defaults
      interface ipdptp*
      inactivity_timeout 900
      debug_level 5
path
      peer_system_name Pnomad1
path
      peer_system_name Pnomad2
path
      peer_system_name Pnomad3
path
      peer_system_name Pnomad4
```

Files

`/etc/asppp.cf`

Configuration file.

`/etc/log/asppp.log`

Message log file.

`/etc/uucp/Devices`

File that contains information for all the devices that can be used to establish a link to a remote computer.

`/etc/uucp/Dialers`

File that contains dialing instructions for many commonly used modems.

`/etc/uucp/Sysfiles`

File that enables you to assign different files to be used by `uucp` and `cu` as `Systems`, `Devices`, and `Dialers` files.

`/etc/uucp/Systems`

File that contains the information needed by the `uucico` daemon to establish a communication link to a remote computer. It is the first file you need to edit to configure UUCP.

`/tmp/.asppp.fifo`

Communication path between `aspppd` and `aspppls`.

`/usr/sbin/aspppd`

Link manager.

`/usr/sbin/aspppls`

Login service.

Attributes

See `attributes`(5) for descriptions of the following attributes.

Attribute Type	**Attribute Value**
Availability	`SUNWapppu`

See Also

`kill(1), ifconfig(1M), in.routed(1M), attributes(5), ppp(7M)`

TCP/IP and Data Communications Administration Guide

asppppls — Asynchronous PPP Link Manager

Synopsis

```
/usr/sbin/asppppls
```

Description

See aspppd(1M).

audit — Control the Behavior of the Audit Daemon

Synopsis

```
/usr/sbin/audit -n | -s | -t
```

Description

When the Basic Security Module is enabled (see bsmconv(1M) for more information), you can use the suite of auditing commands to detect potential security breaches. Auditing can reveal suspicious or abnormal patterns of system use and provide a way to trace suspect actions back to a specific user. Auditing can serve as a deterrent: if users know that their actions are likely to be audited, they may be less likely to attempt malicious activities.

Successful auditing depends on two other security features.

- Identification.
- 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 if a user changes identity (see the su(1M) command), all actions performed are tracked with the same audit ID.

Auditing makes it possible for you to perform the following tasks.

- Monitor security-relevant events that take place on the system.
- Record the events in an audit trail.
- Detect misuse or unauthorized activity (by analyzing the audit trail).

During system configuration, the system administrator selects the activities to monitor. The administrator can also fine-tune the degree of auditing that is done for individual users.

After audit data is collected, audit-reduction and interpretation tools enable you to examine interesting parts of the audit trail. For example, you can choose to look at audit records for individual users or groups, look at all records for a certain type of event on a specific day, or select records that were generated at a certain time of day.

Note — The audit command does not modify a process's preselection mask. It affects only which audit directories are used for audit data storage and specifies the minimum size free.

You start audit by enabling the auditd daemon. See auditd(1M).

When auditing is enabled, use the audit command as the administrative interface for maintaining an audit trail. You can perform the following tasks with the audit command.

- Notify the audit daemon to read the contents of the audit_control(4) file and to reinitialize the current audit directory to the first directory listed in the audit_control file.
  - If auditd receives the signal SIGUSR1, it closes the current audit file and opens another.
  - If SIGHUP is received, auditd closes the current audit trail, rereads the audit_control file, and opens a new trail.
  - If SIGTERM is received, auditd closes the audit trail and terminates auditing.
- Open a new audit file in the current audit directory specified in the audit_control file as last read by the audit daemon.
- Signal the audit daemon to close the audit trail and disable auditing.

Each time the audit daemon opens a new audit trail file, it updates the file audit_data(4) to include the correct name.

Options

-n	Signal audit daemon to close the current audit file, and to open a new audit file in the current audit directory.
-s	Signal audit daemon to read the audit control file. The audit daemon stores the information internally.
-t	Signal audit daemon to close the current audit trail file, disable auditing, and die.

Diagnostics

0	Success.
>0	Failure.

Files

/etc/security/audit_user

An access-restricted database that stores per-user auditing preselection data.

/etc/security/audit_control

File that contains audit control information used by auditd(1M).

Attributes

See attributes(5) for descriptions of the following attributes.

Attribute Type	Attribute Value
Availability	SUNWcsu

See Also

bsmconv(1M), praudit(1M), audit(2), audit_control(4), audit_user(4), attributes(5)

auditconfig — Configure Auditing

Synopsis

```
/usr/sbin/auditconfig option...
```

Description

Use the auditconfig command-line interface to get and set kernel audit parameters.

Note — The functionality described in this manual page is available only if the Basic Security Module (BSM) has been enabled. See bsmconv(1M) for more information.

Options

-chkconf
Check the configuration of kernel audit event to class mappings. If the runtime class mask of a kernel audit event does not match the configured class mask, report a mismatch.

-conf
Configure kernel audit event to class mappings. Runtime class mappings are changed to match those in the audit event to class database file.

-getfsize
Return the maximum audit file size in bytes and the current size of the audit file in bytes.

-setfsize *size*
Set the maximum size of an audit file to *size* bytes. When the size limit is reached, close the audit file and start another.

-getcond
Display the kernel audit condition. The condition displayed is the literal string auditing, which means auditing is enabled and turned on (the kernel audit module is constructing and queuing audit records) or noaudit, which means auditing is enabled but turned off (the kernel audit module is not constructing and queuing audit records), or disabled, which means that the audit module has not

been enabled. See `auditon`(2) and `auditd`(1M) for further information.

`-setcond [auditing|noaudit]`

Set the kernel audit condition to the condition specified where `auditing` enables auditing or `noaudit` disables auditing.

`-getclass` *`event`*

Display the preselection mask associated with the specified kernel audit event. *`event`* is the kernel event number or event name.

`-setclass` *`event audit-flag`* `[,`*`audit-flag`*`...]`

Map the kernel event *`event`* to the classes specified by *`audit-flag`*. *`event`* is an event number or name. An *`audit-flag`* is a two-character string representing an audit class. See `audit_control`(4) for further information.

`-lsevent` Display the currently configured (runtime) kernel and user-level audit event information.

`-getpinfo` *`pid`*

Display the audit ID, preselection mask, terminal ID, and audit session ID for the specified process.

`-setpmask` *`pid flags`*

Set the preselection mask of the specified process. *`flags`* is the ASCII representation of the flags similar to that in *`audit_control`*(4).

`-setsmask` *`asid flags`*

Set the preselection mask of all processes with the specified audit session ID.

`-setumask` *`auid flags`*

Set the preselection mask of all processes with the specified audit ID.

`-lspolicy` Display the kernel audit policies with a description of each policy.

`-getpolicy` Display the kernel audit policy.

`-setpolicy[+|-]`*`policy-flag`*`[,`*`policy-flag`*`...]`

Set the kernel audit policy. *`policy-flag`* is a literal string that denotes an audit policy. A prefix of `+` adds the specified policies to the current audit policies. A prefix of `-` removes the specified policies from the current audit policies. The following list shows the valid policy flag strings. You can also list the current, valid, audit policy-flag strings with `auditconfig -lspolicy`.

`arge` Include the `execv`(2) system call environment arguments to the audit record. This information is not included by default.

`argv` Include the `execv`(2) system call parameter arguments to the audit record. This information is not included by default.

`cnt` Do not suspend processes when audit resources are exhausted. Instead, drop audit records and keep a count of the number of records dropped. By default, processes are suspended until audit resources become available.

`group` Include the supplementary group token in audit records. By default, the group token is not included.

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

`trail` Include the trailer token in every audit record. By default, the trailer token is not included.

`seq` Include the sequence token as part of every audit record. By default, the sequence token is not included. The sequence token attaches a sequence number to every audit record.

Examples

The following example uses the `-getcond` option to show that auditing is enabled.

```
# auditconfig -getcond
audit condition = auditing
#
```

The following example uses the `-getpinfo` option to show information about the `auditd` daemon process.

```
# ps -ef | grep auditd
    root   296     1  0 13:03:45 ?        0:00 /usr/sbin/auditd
    root   517   492  0 13:25:13 pts/6    0:00 grep auditd
# auditconfig -getpinfo 296
audit id = root(0)
process preselection mask = no(0x0,0x0)
terminal id (maj,min,host) = 0,0,unknown(0.0.0.0)
audit session id = 0
#
```

The following example uses the `-setclass` option to map audit event number `10` to the `fr` audit class.

```
# auditconfig -setclass 10 fr
#
```

The following example uses the `-setpolicy` option to turn on inclusion of `exec` arguments in `exec` audit records. You can use the `-setpolicy` option to change the default Solaris BSM audit policies.

```
# auditconfig -setpolicy +argv
```

Exit Status

`0`	Successful completion.
`1`	An error occurred.

Files

`/etc/security/audit_event`

An ASCII system file that stores event definitions and specifies the event to class mappings.

`/etc/security/audit_class`

An ASCII system file that stores class definitions.

Attributes

See `attributes`(5) for descriptions of the following attributes.

Attribute Type	**Attribute Value**
Availability	`SUNWcsu`

See Also

`auditd(1M), bsmconv(1M), praudit(1M), auditon(2), execv(2), audit_class(4), audit_control(4), audit_event(4), attributes(5)`

auditd — Control the Generation and Location of Audit Trail Files

Synopsis

`/usr/sbin/auditd`

Description

The `auditd` audit daemon controls the generation and location of audit trail files. When the Basic Security Module is enabled, the `audit_control` file is read at system startup and the `auditd` daemon is automatically started. See `bsmconv`(1M) for information on how to enable the Basic Security Module.

The `/etc/security/audit_startup` file is an executable script that is invoked as part of the startup sequence just before the execution of the audit daemon (see `audit_startup`(1M)). A default `audit_startup` script, shown below, automatically configures the event to class mappings and sets the audit policies during the BSM (Basic Security Module) package installation. You can edit the `audit_startup` file to define your own auditing policies.

```
#!/bin/sh
auditconfig -conf
auditconfig -setpolicy none
auditconfig -setpolicy +cnt
```

auditd reads the audit_control(4) file to get a list of directories into which audit files can be written and the percentage limit for how much space to reserve on each file system before changing to the next directory.

Use the audit(1M) command to send signals to the auditd daemon. See audit(1M).

Each time the audit daemon opens a new audit trail file, it updates the file audit_data(4) to include the correct name.

Auditing Conditions

The audit daemon invokes the audit_warn(1M) command to warn you of the following specific auditing conditions.

`audit_warn soft` *pathname*

The file system on which *pathname* resides has exceeded the minimum free space limit defined in audit_control(4). A new audit trail has been opened on another file system.

`audit_warn allsoft`

All available file systems have been filled beyond the minimum free space limit. A new audit trail has been opened anyway.

`audit_warn hard` *pathname*

The file system on which *pathname* resides has filled or for some reason become unavailable. A new audit trail has been opened on another file system.

`audit_warn allhard` *count*

All available file systems have been filled or for some reason become unavailable. The audit daemon repeats this call to audit_warn every 20 seconds until space becomes available. *count* is the number of times that audit_warn has been called since the problem arose.

`audit_warn ebusy`

An audit daemon is already running.

`audit_warn tmpfile`

The file /etc/security/audit/audit_tmp exists, indicating a fatal error.

`audit_warn nostart`

The internal system audit condition is AUC_FCHDONE. Auditing cannot be started without rebooting the system.

`audit_warn auditoff`

The internal system audit condition has been changed to not be AUC_AUDITING by someone other than the audit daemon. The audit daemon exits.

`audit_warn postsigterm`

An error occurred during the orderly shutdown of the auditing system.

`audit_warn getacdir`

There is a problem getting the directory list from `/etc/security/audit/audit_control`.Theauditdaemonhangsin a sleep loop until this file is fixed.

Examples

The following example shows the contents of the default `audit_control` file.

```
# more audit_control
#
# Copyright (c) 1988 by Sun Microsystems, Inc.
#
#ident  @(#)audit_control.txt  1.3     97/06/20 SMI
#
dir:/var/audit
flags:
minfree:20
naflags:lo
#
```

The following example shows the contents of the `audit_data` file.

```
# more /etc/security/audit_data
296:/var/audit/19991018050345.not_terminated.paperbark
#
```

Files

`/etc/security/audit_user`

An access-restricted database that stores per-user auditing preselection data.

`/etc/security/audit_control`

File that contains audit control information used by `auditd`(1M).

Attributes

See `attributes`(5) for descriptions of the following attributes.

Attribute Type	**Attribute Value**
Availability	SUNWcsu

See Also

`audit(1M), audit_warn(1M), bsmconv(1M), praudit(1M), auditon(2), auditsvc(2),audit.log(4),audit_control(4),audit_data(4),attributes(5)`

auditreduce — Merge and Select Audit Records from Audit Trail Files

Synopsis

```
/usr/sbin/auditreduce [options][audit-trail-file...]
```

Description

Use the `auditreduce` command to choose sets of records to examine. For example, you can select all records from the past 24 hours to generate a daily report; you can select all records generated by a specific user to examine that user's activities; or you can select all records resulting from a specific event type to see how often that type occurs. Use `auditreduce` to merge audit records from one or more input audit files or to perform a postselection of audit records.

Audit records from one or more input audit trail files are merged into a single output file. The records in an audit trail file are assumed to be sorted in chronological order (oldest first), and this order is maintained by `auditreduce` in the output file.

Unless instructed otherwise, `auditreduce` merges the entire audit trail, which consists of all the audit trail files in the directory structure *`audit_root_dir/*/`*`files` (see `audit_control`(4) for details of the structure of the audit root). Unless you specify the `-R` or `-S` option, `audit_root_dir` defaults to `/etc/security/audit`. By using the file selection options, you can select some subset of these files, files from another directory, or files named explicitly on the command line.

The `select` function enables you to select audit records according to numerous criteria relating to the record's content (see `audit.log`(4) for details of record content). A record must meet all of the record-selection-option criteria to be selected.

> **Note —** The functionality described in this manual page is available only if the Basic Security Module (BSM) has been enabled. See `bsmconv`(1M) for more information.

The Solaris 8 release adds new options to the `auditreduce` command. See “Options."

New!

Audit Trail File-Name Format

Any audit trail file not named on the command line must conform to the audit trail file-name format. Files produced by the audit system already have this format. Output file names produced by `auditreduce` are in the audit trail file-name format shown below.

`start-time.end-time.suffix`

`start-time` is the 14-character timestamp of when the file was opened, *`end-time`* is the 14-character timestamp of when the file was closed, and *`suffix`* is the name of the machine that generated the audit trail file, or some other meaningful suffix (such as `all` if the file contains a combined group of records from many machines). *`end-time`* can be the literal string `not_terminated`, to signify that the file is still being written to by the audit system. Timestamps are of the form *yyyymmddhhmmss* (year, month, day, hour, minute, second). The timestamps are in Greenwich Mean Time (GMT).

Options

File-Selection Options

The file-selection options indicate files that are to be processed and certain types of special treatment.

`-A` Select all of the records from the input files regardless of their timestamp. This option effectively disables the `-a`, `-b`, and `-d` options and is useful in preventing the loss of records if you use the `-D` option to delete the input files after they are processed. Note, however, that if a record is not selected because of another option, then `-A` does not override that case.

`-C` Process only complete files. Files whose file-name end-time timestamp is `not_terminated` are not processed (such a file is currently being written to by the audit system). This option is useful in preventing the loss of records if you use `-D` to delete the input files after they are processed. It does not apply to files specified on the command line.

`-D` *`suffix`* Delete input files after they are processed. The files are deleted only if the entire run is successful. If `auditreduce` detects an error while reading a file, then that file is not deleted. If you specify `-D`, then `-A`, `-C`, and `-O` are also implied. Give *`suffix`* to the `-O` option. This option helps prevent the loss of audit records by ensuring that all of the records are written, only complete files are processed, and the records are written to a file before being deleted. Note that if you specify both `-D` and `-O` on the command line, the order of specification is significant. The suffix associated with the latter specification is in effect.

`-M` *`machine`* Enable selection of records from files with *`machine`* as the file-name suffix. If you do not specify `-M`, process all files regardless of suffix. You can also use `-M` to enable selection of records from files that contain combined records from many machines and have a common suffix (such as `all`).

New! `-N` Select objects in *`newmode`*. This option is off by default to retain backward compatibility. In the existing *`oldmode`*, the `-e`, `-f`, `-g`, `-r`, or `-u` options select not only actions taken with those IDs but also certain objects owned by those IDs. When the system is running in *`newmode`*, only actions are selected. To select objects in *`newmode`*, you must use the `-o` option.

`-O` *`suffix`* Direct output stream to a file in the current *`audit-root-dir`* with the indicated suffix. *`suffix`* can contain a full path name, in which case the last component is taken as the suffix, ahead of which the timestamps are placed, ahead of which the remainder of the path name is placed. If you do not specify the `-O` option, send the output to the standard output. When `auditreduce` puts timestamps in *`filename`*, it uses the times of the first and last records in the merge as the start time and end time.

`-Q` Suppress notification about errors with input files.

`-R` *pathname* Specify the path name of an alternate audit root directory *audit-root-dir* to be *pathname*. Instead of using `/etc/security/audit/*/`*files* by default, examine *pathname*`/*/`*files* instead.

`-S` *server* Read audit trail files from a specific location (server directory). *server* is normally interpreted as the name of a subdirectory of the audit root; therefore, `auditreduce` looks in *audit-root-dir/server/files* for the audit trail files. But if server contains any `/` characters, it is the name of a specific directory not necessarily contained in the audit root. In this case, consult *server/files*. This option enables you to easily manipulate archived files without requiring that they be physically located in a directory structure like that of `/etc/security/audit`.

`-V` Display the name of each file as it is opened and the total number of records written to the output stream.

Record Selection Options

Use the record selection options listed below to indicate which records are written to the output file produced by `auditreduce`.

Note — You cannot specify multiple arguments of the same type.

`-a` *date-time* Choose records that occurred at or after *date-time*. The *date-time* argument is described under “Option Arguments” on page 106. *date-time* is in local time. You can use the `-a` and `-b` options together to form a range.

`-b` *date-time* Choose records that occurred before *date-time*.

`-c` *audit-classes*

Choose records with events that are mapped to the audit classes specified by *audit-classes*. Audit class names are defined in `audit_class`(4). The *audit-classes* can be a comma-separated list of audit flags like those described in `audit_control`(4). Using the audit flags, you can choose records according to success and failure criteria.

`-d` *date-time* Choose records that occurred on a specific day (a 24-hour period beginning at 00:00:00 of the day specified and ending at 23:59:59). The day specified is in local time. Ignore the time portion of the argument if supplied. Choose any records with timestamps during that day. If any hours, minutes, or seconds are given in *date-time*, ignore them. You cannot use `-d` with `-a` or `-b`.

`-e` *effective-user*

Choose records with the specified *effective-user*.

`-f` *`effective-group`*

Choose records with the specified *`effective-group`*.

`-g`*`real-group`* Choose records with the specified *`real-group`*.

`-j`*`subject-ID`* Choose records with the specified *`subject-ID`*, where *`subject-ID`* is a process ID.

`-m` *`event`* Choose records with the specified *`event`*. The event is the literal string or the event number.

`-o` *`object-type`*`=`*`objectID-value`*

Choose records by object type. A match occurs when the record contains the information describing the specified *`object-type`* and the object ID equals the value specified by *`objectID-value`*. The following object types and values are allowed.

`file=`*`pathname`*

Choose records containing file-system objects with the specified path name, where *`pathname`* is a comma-separated list of regular expressions. If a regular expression is preceded by a tilde (~), exclude files matching the expression from the output. For example, the following option chooses all files in `/usr` or `/etc` except those in `/usr/openwin`.

```
file="~/usr/openwin,/usr,/etc"
```

The order of the regular expressions is important because `auditreduce` processes them from left to right and stops when a file is known to be either chosen or excluded. Thus, the following option chooses all files in `/usr` and all files in `/etc`.

```
file=/usr, /etc, ~/usr/openwin
```

Files in `/usr/openwin` are not excluded because the regular expression `/usr` is matched first. Surround the path name with quotes to prevent the shell from expanding any tildes.

New! `filegroup=`*`group`*

Select records containing file-system objects owned by *`group`*.

New! `fileowner=`*`user`*

Select records containing file-system objects owned by *`user`*.

`msgqid=`*`ID`* Choose records containing message queue objects with the specified ID where *`ID`* is a message queue ID.

msgqgroup=*group* New!

Select records containing message queue objects owned or created by *group*.

pid=*ID* Choose records containing process objects with the specified ID where *ID* is a process ID. Note that processes are objects when they are receivers of signals.

procgroup=*group* New!

Select records containing process objects with the real or effective *group*.

procowner=*user* New!

Select records containing process objects with the real or effective *user*.

semid=*ID* Choose records containing semaphore objects with the specified ID where *ID* is a semaphore ID.

semgroup=*group* New!

Select records containing semaphore objects owned or created by *group*.

semowner=*user* New!

Select records containing semaphore objects owned or created by *user*.

shmid=ID Choose records containing shared memory objects with the specified ID where *ID* is a shared memory ID.

shgroup=*group* New!

Select records containing shared memory objects owned or created by *group*.

showner=*user* New!

Select records containing shared memory objects owned or created by *user*.

sock=*port-number*|*machine*

Choose records containing socket objects with the specified *port-number* or the specified machine, where *machine* is a machine name as defined in hosts(4).

-r *real-user* Choose records with the specified *real-user*.

-u*audit-user* Choose records with the specified *audit-user*. When one or more file-name arguments appear on the command line, process only the named files.

Files specified in this way need not conform to the audit trail file-name format. However, you cannot use `-M`, `-S`, and `-R` when processing named files. If the file name is `-`, then take the input from the standard input.

Option Arguments

`audit-trail-file`

An audit trail file as defined in `audit.log`(4). An audit trail file not named on the command line must conform to the audit trail file-name format. Audit trail files produced as output of `auditreduce` use the following format.

`start-time.end-time.suffix`

`start-time` is the 14-character timestamp denoting when the file was opened. *`end-time`* is the 14-character timestamp denoting when the file was closed. *`end-time`* can also be the literal string `not_terminated`, signifying that the file is still being written to by the audit daemon or that the file was not closed properly (a system crash or abrupt halt occurred). *`suffix`* is the name of the machine that generated the audit trail file (or some other meaningful suffix. For example, `all` would be a good suffix if the audit trail file contains a combined group of records from many machines).

`date-time` The *`date-time`* argument to `-a`, `-b`, and `-d` can be of two forms. An absolute date-time takes the following form

`yyyymmdd` [*`hh`* [*`mm`* [*`ss`*]]]

where *`yyyy`* specifies a year (with `1970` as the earliest value), *`mm`* is the month (`01-12`), *`dd`* is the day (`01-31`), *`hh`* is the hour (`00-23`), *`mm`* is the minute (`00-59`), and *`ss`* is the seconds (`00-59`). The default is `00` for *`hh`*, *`mm`*, and *`ss`*.

You can also specify an offset such as `+`*`n`* `d|h|m|s` where *`n`* is a number of units and the tags `d`, `h`, `m`, and `s` stands for days, hours, minutes, and seconds. An offset is relative to the starting time. Thus, you can use this form only with the `-b` option.

`event` The literal string or ordinal event number as found in `audit_event`(4). If *`event`* is not found in the `audit_event` file, it is considered invalid.

`group` The literal string or ordinal group ID number as found in `group`(4). If *`group`* is not found in the `group` file, it is considered invalid. *`group`* can be negative.

`pathname` A regular expression describing a path name.

`user` The literal user name or ordinal user ID number as found in `passwd`(4). If the user name is not found in the `passwd` file, it is considered invalid. *`user`* can be negative.

Examples

The following example pipes the output of auditreduce through the praudit(1M) command to display audit records in human-readable form.

```
# auditreduce | praudit
file,Mon 18 Oct 1999 12:35:33 PM WST, + 0 msec,
header,36,2,system booted,na,Mon 18 Oct 1999 12:35:33 PM WST, +
  330000749 msec
text,booting kernel
header,86,2,su,,Mon 18 Oct 1999 12:57:30 PM WST, + 800005198 msec
subject,winsor,root,staff,winsor,staff,491,377,0 0 paperbark
text,success for user root
return,success,0
header,36,2,system booted,na,Mon 18 Oct 1999 01:02:47 PM WST, +
  329999144 msec
text,booting kernel
header,86,2,su,,Mon 18 Oct 1999 01:05:55 PM WST, + 919999045 msec
subject,winsor,root,staff,winsor,staff,492,376,0 0 paperbark
text,success for user root
return,success,0
header,36,2,system booted,na,Mon 18 Oct 1999 04:04:27 PM WST, +
  339999508 msec
text,booting kernel
header,86,2,su,,Mon 18 Oct 1999 04:08:56 PM WST, + 440004800 msec
subject,winsor,root,staff,winsor,staff,494,376,0 0 paperbark
text,success for user root
return,success,0
file,Mon 18 Oct 1999 04:08:56 PM WST, + 0 msec,
```

The following example views the audit data for February 16, 2000, for user bcalkins.

```
# auditreduce -d 20000216 -u bcalkins | praudit
file,Wed 31 Dec 1969 07:00 PM EST, + 0 msec
file,Wed 16 Feb 2000 11:41 AM EST, + 0 msec
#
```

If you are combining all the audit trail files into one large file, then deleting the original files could be desirable to prevent the records from appearing twice, as shown in the following example.

```
# auditreduce -V -d /etc/security/audit/combined/all
auditreduce: command line error - invalid date/time format - not all
  digits (/etc/security/audit/combined/all).
#
```

The following example prints what user winsor did on October 18, 1999. The output is piped through the praudit command to display it in a human-readable form to the standard output.

```
# auditreduce -d 19991018 -u winsor | praudit
file,Mon 18 Oct 1999 12:57:30 PM WST, + 0 msec,
header,86,2,su,,Mon 18 Oct 1999 12:57:30 PM WST, + 800005198 msec
subject,winsor,root,staff,winsor,staff,491,377,0 0 paperbark
text,success for user root
return,success,0
header,86,2,su,,Mon 18 Oct 1999 01:05:55 PM WST, + 919999045 msec
```

```
subject,winsor,root,staff,winsor,staff,492,376,0 0 paperbark
text,success for user root
return,success,0
header,86,2,su,,Mon 18 Oct 1999 04:08:56 PM WST, + 440004800 msec
subject,winsor,root,staff,winsor,staff,494,376,0 0 paperbark
text,success for user root
return,success,0
file,Mon 18 Oct 1999 04:08:56 PM WST, + 0 msec,
#
```

The above example may produce a large volume of data if user `winsor` has been busy. To restrict the output, the following example uses the `-c` option to select records from a specified class.

```
# auditreduce -d 19991018 -u winsor -c lo | praudit
file,Mon 18 Oct 1999 12:57:30 PM WST, + 0 msec,
header,86,2,su,,Mon 18 Oct 1999 12:57:30 PM WST, + 800005198 msec
subject,winsor,root,staff,winsor,staff,491,377,0 0 paperbark
text,success for user root
return,success,0
header,86,2,su,,Mon 18 Oct 1999 01:05:55 PM WST, + 919999045 msec
subject,winsor,root,staff,winsor,staff,492,376,0 0 paperbark
text,success for user root
return,success,0
header,86,2,su,,Mon 18 Oct 1999 04:08:56 PM WST, + 440004800 msec
subject,winsor,root,staff,winsor,staff,494,376,0 0 paperbark
text,success for user root
return,success,0
file,Mon 18 Oct 1999 04:08:56 PM WST, + 0 msec,
#
```

The following example shows `milner`'s login/logout activity for April 13, 14, and 15. The results are saved to a file in the current working directory. Note that the name of the output file has `milnerlo` as the suffix, with the appropriate timestamp prefixes. Note that the long form of the name is used for the `-c` option.

```
% auditreduce -a 19880413 -b +3d -u milner -c login_logout -o milnerlo
```

To follow `milner`'s movement about the file system on April 13, 14, and 15, you could view the `chdir` record types. Note that to get the same time range as the above example, you needed to specify the `-b` *`time`* as the day after our range because `19880416` defaults to midnight of that day, and records before that fall on 0415, the end-day of the range.

```
% auditreduce -a 19880413 -b 19880416 -u milner -m AUE_CHDIR | praudit
```

In the following example, the audit records are being collected in summary form (the login/logout records only). The records are written to a summary file in a directory different from the normal audit root to prevent the selected records from existing twice in the audit root.

```
% auditreduce -d 19880330 -c lo -o /etc/security/audit_summary/logins
```

If activity for user ID 9944 has been observed but that user is not known to the system administrator, then the following example searches the entire audit trail for any records generated by that user. auditreduce queries the system about the current validity of ID 9944 and prints a warning message if it is not currently active.

```
% auditreduce -o /etc/security/audit_suspect/user9944 -u 9944
```

Files

/etc/security/audit/*server*/files/*

Location where audit trails are stored.

Attributes

See attributes(5) for descriptions of the following attributes.

Attribute Type	**Attribute Value**
Availability	SUNWcsu

See Also

bsmconv(1M),praudit(1M),audit.log(4),audit_class(4),audit_control(4), group(4), hosts(4), passwd(4), attributes(5)

Diagnostics

auditreduce prints error messages for command-line errors and then exits. If fatal errors occur during the run, auditreduce prints an explanatory message and exits. In this case, the output file may be in an inconsistent state (no trailer or partially written record) and auditreduce prints a warning message before exiting. Successful invocation returns 0 and unsuccessful invocation returns 1.

Because auditreduce may be processing a large number of input files, it is possible to exceed the machinewide limit on open files. If this happens, auditreduce prints a message to that effect, gives information on how many file there are, and exits.

If auditreduce prints a record's timestamp in a diagnostic message, that time is in local time. However, when file names are displayed, their timestamps are in GMT.

Bugs

Conjunction, disjunction, negation, and grouping of record selection options should be allowed.

audit_startup — Audit Subsystem Initialization Script

Synopsis

/etc/security/audit_startup

Description

Use the audit_startup script to initialize the audit subsystem before the audit daemon is started. You can configure this script, which currently consists of a series of auditconfig(1M) commands to set the system default policy and download the initial event to class mapping.

> **Note —** The functionality described in this manual page is available only if the Basic Security Module (BSM) has been enabled. See bsmconv(1M) for more information.

See Also

auditconfig(1M), auditd(1M), bsmconv(1M), attributes(5)

auditstat — Display Kernel Audit Statistics

Synopsis

```
/usr/sbin/auditstat [-c count][-h numlines][-i interval][-n][-v]
```

Description

Use the auditstat command to display kernel audit statistics, as shown in the following example.

```
# auditstat
 gen nona kern  aud  ctl  enq wrtn wblk rblk drop  tot  mem
   2    1    0    1    0    2    2    0    2    0    0   48
#
```

> **Note —** The functionality described in this manual page is available only if the Basic Security Module (BSM) has been enabled. See bsmconv(1M) for more information.

The following list describes, in alphabetical order, the fields in the auditstat output.

`aud` The total number of audit records processed by the audit(2) system call.

`ctl` This field is obsolete.

`drop` The total number of audit records that have been dropped. Records are dropped according to the kernel audit policy. See auditon(2), AUDIT_CNT policy, for details.

`enq` The total number of audit records put on the kernel audit queue.

`gen` The total number of audit records that have been constructed (not the number written).

`kern` The total number of audit records produced by user processes (as a result of system calls).

`mem` The total number of kilobytes of memory currently in use by the kernel audit module.

`nona` The total number of nonattributable audit records that have been constructed. These are audit records that are not attributable to any particular user.

`rblk` The total number of times that `auditsvc`(2) has blocked, waiting to process audit data.

`tot` The total number of kilobytes of audit data written to the audit trail.

`wblk` The total number of times that user processes blocked on the audit queue at the high watermark.

`wrtn` The total number of audit records written. The difference between `enq` and `wrtn` is the number of outstanding audit records on the audit queue that have not been written.

Options

`-c` *`count`* Display the statistics a total of *`count`* times. If *`count`* is equal to zero, statistics are displayed indefinitely. You must specify a time interval.

`-h` *`numlines`* Display a header for every *`numlines`* of statistics printed. The default is to display the header every 20 lines. If *`numlines`* is equal to 0, never display the header.

`-i` *`interval`* Display the statistics every interval where *`interval`* is the number of seconds to sleep between each collection.

`-n` Display the number of kernel audit events currently configured.

`-v` Display the version number of the kernel audit module software.

Exit Status

`0` Success.

`1` Failure.

Attributes

See `attributes`(5) for descriptions of the following attributes.

Attribute Type	**Attribute Value**
Availability	`SUNWcsu`

See Also

`auditconfig(1M), praudit(1M), bsmconv(1M), audit(2), auditon(2), auditsvc(2), attributes(5)`

audit_warn — Audit Daemon Warning Script

Synopsis

```
/etc/security/audit_warn [option [arguments]]
```

Description

The audit_warn script processes warning or error messages from the audit daemon. When a problem is encountered, the audit daemon, auditd(1M) calls audit_warn with the appropriate arguments. The *option* argument specifies the error type.

You can specify a list of mail recipients to be notified when an audit_warn situation arises, by defining a mail alias called audit_warn in aliases(4). The users that make up the audit_warn alias are typically the audit and root users.

Note — The functionality described in this manual page is available only if the Basic Security Module (BSM) has been enabled. See bsmconv(1M) for more information.

Options

allhard *count*

Indicate that the hard limit for all file systems has been exceeded *count* times. The default action for this option is to send mail to the audit_warn alias only if the count is 1 and to write a message to the machine console every time. It is recommended that mail not be sent every time because you could saturate the file system that contains the mail spool directory.

allsoft

Indicate that the soft limit for all file systems has been exceeded. The default action for this option is to send mail to the audit_warn alias and to write a message to the machine console.

auditoff

Indicate that someone other than the audit daemon changed the system audit state to something other than AUC_AUDITING. The audit daemon has exited in this case. The default action for this option is to send mail to the audit_warn alias and to write a message to the machine console.

ebusy

Indicate that the audit daemon is already running. The default action for this option is to send mail to the audit_warn alias and to write a message to the machine console.

getacdir *count*

Indicate that there is a problem getting the directory list from audit_control(4). The audit daemon hangs in a sleep loop until the file is fixed. The default action for this option is to send mail to the audit_warn alias only if count is 1 and to write a message to the machine console every time. It is recommended that mail not be sent every time because you could saturate the file system that contains the mail spool directory.

`hard`*`filename`*

Indicate that the hard limit for the file has been exceeded. The default action for this option is to send mail to the `audit_warn` alias and to write a message to the machine console.

`nostart` Indicate that auditing could not be started. The default action for this option is to send mail to the `audit_warn` alias and to write a message to the machine console. Some administrators may prefer to modify `audit_warn` to reboot the system when this error occurs.

`postsigterm` Indicate that an error occurred during the orderly shutdown of the audit daemon. The default action for this option is to send mail to the `audit_warn` alias and to write a message to the machine console.

`soft`*`filename`*

Indicate that the soft limit for *`filename`* has been exceeded. The default action for this option is to send mail to the `audit_warn` alias and to write a message to the machine console.

`tmpfile` Indicate that the temporary audit file already exists, indicating a fatal error. The default action for this option is to send mail to the `audit_warn` alias and to write a message to the machine console.

Attributes

See `attributes`(5) for descriptions of the following attributes.

Attribute Type	**Attribute Value**
Availability	SUNWcsr

See Also

`audit(1M), auditd(1M), bsmconv(1M), aliases(4), audit.log(4), audit_control(4), attributes(5)`

automount — Install Automatic Mount Points

Synopsis

```
/usr/sbin/automount [-t duration][-v]
```

Description

The `automount` command installs `autofs` mount points and associates an automount map with each mount point. The `automount` command is run automatically at boot time by the /etc/rc2.d/S74autofs script. If you make changes to the automounting maps, you may want to run the command manually to update the automounter.

The autofs file-system monitors try to access directories within the file system and notify the automountd(1M) daemon. The daemon uses the map to locate a file system, which it then mounts at the point of reference within the autofs file system. You can assign a map to an autofs mount by using an entry in the /etc/auto_master map or by using a direct map.

If the file system is not accessed within an appropriate interval (five minutes by default), the automountd daemon unmounts the file system.

The file /etc/auto_master determines the locations of all autofs mount points. By default, this file contains the following four entries.

```
#
# Master map for automounter
#
+auto_master
/net  -hosts -nosuid
/home  auto_home
/xfn  -xfn
```

The +auto_master entry is a reference to an external NIS or NIS+ master map. If such a map exists, then its entries are read as if they occurred in place of the +auto_master entry. The remaining entries in the master file specify a directory on which an autofs mount is made followed by the automounter map to be associated with it. You can supply optional mount options as an optional third field in the each entry. These options are used for any entries in the map that do not specify mount options explicitly. You usually run the automount command without arguments. It compares the entries in /etc/auto_master with the current list of autofs mounts in /etc/mnttab and adds, removes, or updates autofs mounts to bring /etc/mnttab up to date with the /etc/auto_master file.

At boot time automount installs all autofs mounts from the master map. You can subsequently run it to install autofs mounts for new entries in the master map or the direct map or to perform unmounts for entries that have been removed from these maps.

Notes

autofs mount points must not be hierarchically related. automount does not allow an autofs mount point to be created within another autofs mount.

Because each direct map entry results in a new autofs mount, such maps should be kept short.

You can modify entries in both direct and indirect maps at any time. The new information is used when automountd next uses the map entry to do a mount.

New entries added to a master map or direct map are not useful until the automount command is run to install them as new autofs mount points. New entries added to an indirect map can be used immediately.

Starting with the Solaris 2.6 release, a listing (see ls(1)) of the autofs directory associated with an indirect map shows all potential mountable entries. The attributes associated with the potential mountable entries are temporary. The real file-system attributes are only shown once the file system has been mounted.

You can assign default mount options to an entire map when you specify them as an optional third field in the master map. These options apply only to map entries that have no mount options. Note that map entities with options override the default options because, at this time, the options do not concatenate. The concatenation feature is planned for a future release.

The Network Information Service (NIS) was formerly known as Sun Yellow Pages (YP). The functionality of the two remains the same.

Options

`-t` *`duration`* Specify a duration, in seconds, that a file system is to remain mounted when not in use. The default is 10 minutes.

`-v` Notify of `autofs` mounts, unmounts, or other non-essential information.

Usage

Map Entry Format

A simple map entry (mapping) takes the following form

```
key [-mount-options] location...
```

where *`key`* is the full path name of the directory to mount when used in a direct map, or the simple name of a subdirectory in an indirect map. *`mount-options`* is a comma-separated list of mount options, and *`location`* specifies a file system from which the directory can be mounted. In the case of a simple NFS mount, the options that can be used are as specified in `mount_nfs`(1M), and *`location`* takes the following form

```
host:pathname
```

where *`host`* is the name of the host from which to mount the file system, and *`pathname`* is the absolute path name of the directory to mount.

Options to other file systems are documented on the other `mount_*` reference manual pages, for example, `mount_cachefs`(1M).

Replicated File Systems

You can specify multiple location fields for replicated NFS file systems, in which case `automount` and the kernel each try to use that information to increase availability. If the read-only flag is set in the map entry, `automount` mounts a list of locations that the kernel can use, sorted by several criteria. When a server does not respond, the kernel switches to an alternate server. The sort ordering of `automount` determines how the next server is chosen. If the read-only flag is not set, `automount` mounts the best single location, chosen by the same sort ordering, and new servers are chosen only when an unmount has been possible and a remount is done. Servers on the same local subnet are given the strongest preference, and servers on the local net are given the second strongest preference. Among servers equally far away, response times determine the order if no weighting factors (see below) are used.

If the list includes server locations using both the NFS Version 2 Protocol and the NFS Version 3 Protocol, `automount` chooses only a subset of the server locations on the list, so that all entries are the same protocol. It chooses servers with the NFS Version 3 Protocol as long as an NFS Version 2 Protocol server on a local subnet is not ignored. See the *NFS Administration Guide* for additional details.

If each location in the list shares the same path name, then a single location can be used with a comma-separated list of host names.

```
hostname,hostname...:pathname
```

You can weight requests for a server by appending the weighting factor as an integer in parentheses to the server name. Servers without a weighting are assumed to have a value of zero (most likely to be selected). Progressively higher values decrease the chance of being selected. In the following example, hosts alpha and bravo have the highest priority. Host delta has the lowest.

```
man -ro alpha,bravo,charlie(1),delta(4):/usr/man
```

Server proximity takes priority in the selection process. In the example above, if the server delta is on the same network segment as the client but the others are on different network segments, then the weighting value is ignored and delta is selected. The weighting has effect only when selecting between servers with the same network proximity.

In cases where each server has a different export point, the weighting can still be applied. For example,

```
man -ro alpha:/usr/man bravo,charlie(1):/usr/share/man
     delta(3):/export/man
```

You can continue a mapping across input lines by escaping the newline with a backslash (\). Comments begin with a number sign (#) and end at the subsequent newline.

Map Key Substitution

The ampersand (&) character is expanded to the value of the key field for the entry in which it occurs. In the following example, the & expands to jane.

```
jane sparcserver:/home/&
```

Wildcard Key

The asterisk (*) character, when supplied as the key field, is recognized as the catch-all entry. Such an entry matches any key not previously matched. For instance, if the following entry appeared in the indirect map for /config

```
* &:/export/config/&
```

it would allow automatic mounts in /config of any remote file system whose location could be specified as

hostname:/export/config/*hostname*

Variable Substitution

You can use client-specific variables within an automount map. For example, if $HOST appeared within a map, automount would expand it to its current value for the client's host name. The following variables are supported.

ARCH The application architecture name. The architecture name is derived from the output of uname -m, for example, sun4 on a sun4u system.

CPU The processor type output of uname -p. For example, sparc.

HOST The host-name output of uname -n. For example, castle.

OSNAME The OS name output of uname -s. For example, SunOS.

OSREL The OS release name output of uname -r. For example 2.5, 2.5.1, 2.6, 5.7, 5.8.

OSVERS	The OS version output of `uname -v`. For example, `Beta`.
NATISA	The native instruction set architecture output of `isainfo -n`. For example, `sparcv9`.

If you need to protect a reference from affixed characters, surround the variable name with curly braces ({}).

Multiple Mounts

A multiple mount entry takes the following form.

```
key [-mount-options][[mountpoint][-mount-options] location...]...
```

The initial [*mountpoint*] is optional for the first mount and mandatory for all subsequent mounts. The optional *mountpoint* is taken as a path name relative to the directory named by *key*. If you omit *mountpoint* in the first occurrence, a mount point of / (root) is implied.

Given the following entry in the indirect map for `/src`, all offsets must exist on the server under `beta.automount` automatically mounts `/src/beta`, `/src/beta/1.0`, and `/src/beta/1.0/man`, as needed, from either `svr1` or `svr2`, whichever host is nearest and responds first.

```
beta -ro \
       /          svr1,svr2:/export/src/beta\
       /1.0       svr1,svr2:/export/src/beta/1.0\
       /1.0/man   svr1,svr2:/export/src/beta/1.0/man
```

The backslash at the end of each line tells the automounter to consider the entire entry as one line, and it makes the entry easier to read. The last entry line does not have a backslash because it ends the sequence.

Other File-System Types

The automounter assumes NFS mounts as a default file-system type. You can describe other file-system types with the *fstype* mount option. You can combine other mount options specific to this file-system type with the *fstype* option. The location field must contain information specific to the file-system type. If the location field begins with a slash, you must prepend a colon character. The following example mounts a CD file system.

```
cdrom -fstype=hsfs,ro :/dev/sr0
```

The following example performs an `autofs` mount.

```
src -fstype=autofs auto_src
```

Note — Use the above procedure only if you are not using Volume Manager.

Mounts using CacheFS are most useful when applied to an entire map as map defaults. The following entry in the master map describes cached home directory mounts. It assumes the default location of the cache directory, `/cache`.

```
/home auto_home -fstype=cachefs,backfstype=nfs
```

See “Notes” on page 114 for information on option inheritance.

Indirect Maps

An indirect map enables you to specify mappings for the subdirectories you want to mount under the directory indicated on the command line. In an indirect map, each key consists of a simple name that refers to one or more file systems that are to be mounted as needed. The `auto_home` map is a good example of an indirect map that mounts a resource from a single server.

Direct Maps

Entries in a direct map are associated directly with `autofs` mount points. Each key is the full path name of an `autofs` mount point. The direct map as a whole is not associated with any single directory.

Included Maps

You can include the contents of another map within a map with an entry of the form +*mapname*.

If *mapname* begins with a slash, it is assumed to be the path name of a local file. Otherwise, the location of the map is determined by the policy of the nameservice switch according to the entry for the automounter in `/etc/nsswitch.conf`, such as

```
automount: files nis
```

If the nameservice is `files`, then the name is assumed to be that of a local file in `/etc`. If the key being searched for is not found in the included map, the search continues with the next entry.

Special Maps

Three special maps are available.

```
-hosts
-xfn
-null
```

The `-hosts` map is used with the `/net` directory and assumes that the map key is the host name of an NFS server. The `automountd` daemon dynamically constructs a map entry from the server's list of exported file systems. For example, a reference to `/net/hermes/usr` would initiate an automatic mount of all exported file systems from `hermes` that are mountable by the client. References to a directory under `/net/hermes` refers to the corresponding directory relative to `hermes` root.

The `-xfn` map is used to mount the initial context of the Federated Naming Service (FNS) namespace under the `/xfn` directory. For more information on FNS, see `fns`(5), `fns_initial_context`(5), `fns_policies`(5), and the *Federated Naming Service Guide*.

The `-null` map, when indicated on the command line, cancels a previous map for the directory indicated. This map is most useful in the `/etc/auto_master` for cancelling entries that would otherwise be inherited from the `+auto_master` include entry. For `-null` entries to be effective, they must be inserted before the included map entry.

Executable Maps

Local maps that have the execute bit set in their file permissions are executed by the automounter and provided with a key to be looked up as an argument. The executable map is expected to return the content of an automounter map entry on its standard output or no output if the entry cannot be determined. You cannot make a direct map executable.

Configuration and the auto_master Map

When initiated without arguments, automount consults the master map for a list of autofs mount points and their maps. It mounts any autofs mounts that are not already mounted and unmounts autofs mounts that have been removed from the master map or direct map.

The master map is assumed to be called auto_master, and its location is determined by the nameservice switch policy. Normally, the master map is located initially as a local file /etc/auto_master.

Browsing

Starting with the Solaris 2.6 release, browsing of indirect maps is supported. This feature enables all of the potential mount points to be visible, regardless of whether they are mounted. You can add the -nobrowse option to any indirect autofs map to disable browsing, as shown in the following example.

```
/net -hosts -nosuid,nobrowse
/home auto_home
```

In this case, any host names would be visible in /net only after they are mounted, but all potential mount points would be visible under /home. The -browse option enables browsability of autofs file systems. It is the default for all indirect maps.

Exit Status

0	Successful completion.
1	An error occurred.

Files

/etc/auto_master

Master automount map.

/etc/auto_home

Map to support automounted home directories.

/etc/nsswitch.conf

The nameservice switch configuration file.

Attributes

See attributes(5) for descriptions of the following attributes.

Attribute Type	**Attribute Value**
Availability	SUNWcsu

See Also

isainfo(1), ls(1), uname(1), automountd(1M), mount(1M), mount_cachefs(1M), mount_nfs(1M), attributes(5), fns(5), fns_initial_context(5), fns_policies(5), nfssec(5)

NFS Administration Guide

automountd — Mount/Unmount Daemon for autofs

Synopsis

```
/usr/lib/fs/autofs/automountd [-Tvn][-D name=value]
```

Description

The automountd daemon controls automounting activities. It is an RPC server that answers file-system mount and unmount requests from the autofs file system. automountd uses local files or nameservice maps to locate file systems to be mounted. These maps are described with the automount(1M) command.

The automountd daemon is automatically invoked in run level 2 from the /etc/rc2.d/S74autofs run control script.

Options

-T	Expand each RPC call and display it on the standard output.
-v	Log status messages to the console.
-n	Turn off browsing for all autofs mount points. This option overrides the -browse autofs map option on the local host.
-D*name*=*value*	Assign *value* to the indicated automount map-substitution variable. You cannot use these assignments to substitute variables in the master map auto_master.

Usage

See largefile(5) for the description of the behavior of automountd when encountering files greater than or equal to 2 Gbytes (2**31 bytes).

Files

/etc/auto_master

Master map for automounter.

Attributes

See attributes(5) for descriptions of the following attributes.

Attribute Type	**Attribute Value**
Availability	SUNWcsu

See Also

automount(1M), attributes(5), largefile(5)

autopush — Configure Lists of Automatically Pushed STREAMS Modules

Synopsis

```
/usr/sbin/autopush -f filename
/usr/sbin/autopush -g -M major -m minor
/usr/sbin/autopush -r -M major -m minor
```

Description

In the SunOS 4.x release, the `streamtab` structure enabled a driver to specify that certain STREAMS modules be pushed when the device was opened. In the Solaris 8 operating environment, you use the `autopush` command to specify when a STREAMS module is pushed. If required, you can run `autopush` at driver installation. The `/etc/inittab` file contains an entry that automatically runs `autopush` at bootup.

The `autopush` command configures the list of modules to be automatically pushed onto the stream when a device is opened. You can also use it to remove a previous setting or get information on a setting. The `autopush` command configures the list of modules for a STREAMS device. It automatically pushes a prespecified list of modules from the `/etc/iu.ap` autopush configuration file onto the stream when the STREAMS device is opened and the device is not already open.

The following example shows an `/etc/iu.ap` configuration file.

```
# /dev/console and /dev/contty autopush setup
#
#      major   minor lastminor  modules

        wc      0       0       ldterm ttcompat
        zs      0       1       ldterm ttcompat
        ptsl    0       15      ldterm ttcompat
```

The first line configures a single minor device whose major name is `wc` and whose minor numbers start and end at `0`, creating only one minor number. The `ldterm` and `ttcompat` modules are automatically pushed. The second line configures the `zs` driver whose minor device numbers are `0` and `1`, and automatically pushes the same modules. The last line configures the `pts1` driver whose minor device numbers are from `0` to `15`, and automatically pushes the same modules.

Options

`-f` *`filename`* Set up the autopush configuration for each driver according to the information stored in *`filename`*. An autopush file consists of lines of four or more fields, separated by spaces, as shown below.

`major minor last-minor module1 module2... module8` `[`*`anchor`*`]` New!

The first field is a string that specifies the major device name, as listed in the `/kernel/drv` directory. The next two fields are integers that specify the minor device number and *`last-minor`* device number. The fields following represent the names of modules. If *minor* is -1, then all minor devices of a major driver specified by *major* are configured and the value for *last-minor* is ignored. If *last-minor* is 0, then only a single minor device is configured. To configure a range of minor devices for a particular major, *minor* must be less than *last-minor*.

The remaining fields list the names of modules to be automatically pushed onto the stream when opened, along with the position of an optional anchor. The maximum number of modules that can be pushed is eight. The modules are pushed in the order you specify them. The optional special character sequence [*anchor*] puts a STREAMS anchor on the stream at the module previously specified in the list. Specifying more than one anchor or putting an anchor first in the list results in an error.

A non-zero exit status indicates that one or more of the lines in the specified file failed to complete successfully.

`-g` Get the current configuration setting of a particular major and minor device number specified with the -M and -m options and display the autopush modules associated with it. Return the starting minor device number if the request corresponds to a setting of a range (as described with the -f option).

`-M` *major* Specify the major device number.

`-m` *minor* Specify the minor device number.

`-r` Remove the previous configuration setting of the particular major and minor device number specified with the -M and -m options. If the values of *major* and *minor* correspond to a previously established setting of a range of minor devices, where *minor* matches the first minor device number in the range, remove the configuration for the entire range.

Examples

The following example gets the current configuration settings for the major device 29 and minor device 0 and displays the autopush modules associated with them for the character-special device /dev/term/a.

```
# autopush -g -M 29 -m 0
     Major      Minor  Lastminor        Modules
        29          0         63        ldterm ttcompat
#
```

Exit Status

0 Successful completion.

non-zero An error occurred.

Files

`/etc/iu.ap` Contains a prespecified list of modules that can be pushed onto the streams device if the STREAM is not already open.

Attributes

See `attributes`(5) for descriptions of the following attributes.

Attribute Type	**Attribute Value**
Availability	`SUNWcsu`

See Also

`bdconfig(1M)`, `ttymon(1M)`, `attributes(5)`, `sad(7D)`, `streamio(7I)`, `ldterm(7M)`, `ttcompat(7M)`

STREAMS Programming Guide