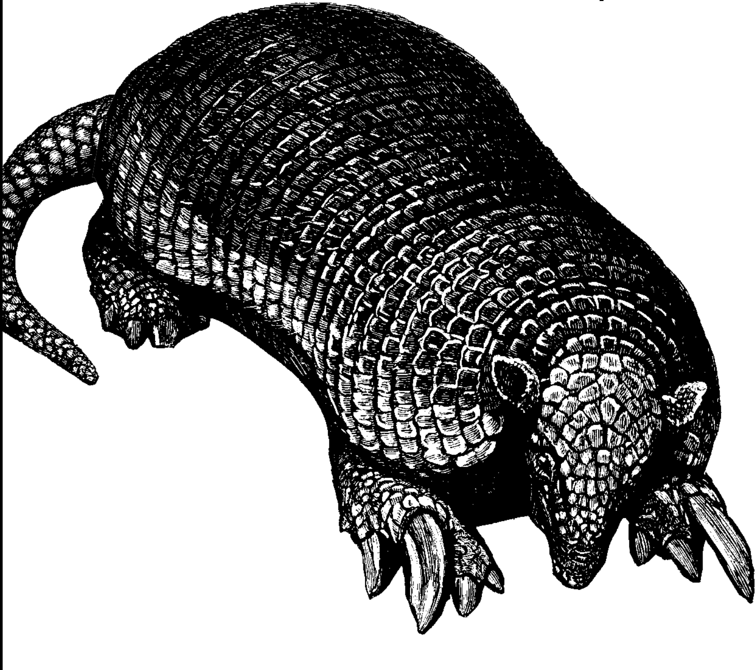
Commands and File Formats

Essential System Administration

Pocket Reference



O'REILLY®

Æleen Frisch

Essential System Administration

Pocket Reference

Æleen Frisch

O'REILLY®

Beijing · Cambridge · Farnham · Köln · Paris · Sebastopol · Taipei · Tokyo

User Accounts

/etc/passwd

The password file

username:*x*:*UID*:*GID*:*user-info*:*home-dir*:*login-shell*

username User account login name (generally limited to 8 characters).

x Traditional password field, set to a fixed character (usually x) when a shadow password file is in use. AIX uses an exclamation point (!), and FreeBSD uses an asterisk (*).

UID The user identification number.

GID The user's primary group membership.

user-info Conventionally, contains the user's full name and, possibly, other job-related information (e.g., office location or phone number). Up to five comma-separated subfields may be defined.

home-dir The user's home directory.

login-shell The program used as the command interpreter for this user. On most systems, the */etc/shells* file lists the full pathnames of valid shell programs; on AIX systems, valid shells are listed in the shells field in the usw stanza of */etc/security/login.cfg*:

```
shells = /bin/sh, /bin/csh, …
```

Shadow password files

/etc/shadow (Linux and Solaris)

user:*pwd*:*changed*:*minlife*:*maxlife*:*warn*:*inactive*:*expires*:

user Username (as in */etc/passwd*).

pwd Encoded password.

changed Last password change (Unix date format[*]).

minlife Minimum number of days a user must keep a new password.

maxlife Maximum password lifetime, in days.

warn Days to warn in advance of an upcoming password expiration.

inactive Days after password expiration that the account will be disabled.

expires Date the account expires (Unix date format).

/etc/master.passwd (FreeBSD)

```
user:pwd:UID:GID:class:pwd-expire:acct-expire:
user-info:…
```

This file, which serves as both *passwd* and *shadow* files, uses three additional fields between the GID and user information fields:

pwd Encoded password.

class User class (see page 37).

pwd-expire Password expiration date (Unix date format).

acct-expire Account expiration date (Unix date format).

/etc/security/passwd (AIX)

Encoded passwords are stored in the `password` field.

/etc/group

The group file

```
name:*:GID:additional-members
```

name Group name.

* Placeholder character for vestigial group password. Some systems use `!` or `x` in this field. Linux uses group passwords.

GID Group ID number.

[*] Unix systems often store dates as the number of seconds (or milliseconds) since midnight on 1/1/1970.

adtl-members	List of group members in addition to those having the group in the GID field of their password file entry.

The HP-UX /etc/logingroup file

If present, this file has the sam syntax as */etc/group*. The user lists in this file determine each user's initial login group.

/etc/gshadow

The Linux shadow group file

```
name:pwd:group-admins:additional-users
```

name	Group names, as in */etc/group*.
pwd	Encoded group password, controls who can use the `newgrp` command with this group.
group-admins	Group administrators: can change the group password and member lists.
adtl-users	List of additional group members (usually the same as */etc/group*).

gpasswd

```
gpasswd group                                Change group password
gpasswd [options] user[,user…] group    Modify group files
```

Change password or add/remove group members and/or administrators.

Options

`-a|-d`	Add/remove users from *group*'s member list in both files.
`-M`	Specify the complete additional-members list (both files).
`-A`	Specify the complete group-administrator list.
`-R|-r`	Disable/remove group password, allowing no one/everyone to use `newgrp` with the group.

FreeBSD user classes

User classes allow account attributes and login environment settings to be applied to many accounts. Classes are independent of Unix groups. They are defined in */etc/login.conf*:

```
class-name:\
   :attribute=value:\
   :attribute=value:\
   ...
```

Important attributes are discussed in the next section.

cap_mkdb (FreeBSD)

```
cap_mkdb -v file
```

Recreate the database that corresponds to *file* (often */etc/login.conf*).

Solaris Projects

projadd
projmod

```
projadd [options] name
projmod [options] name
```

Add or modify a project.

Options

`-c` *string*	Project description.
`-p` *n*	Set project ID to *n*.
`-U` *user*[,*user*]	Place user(s) into project.
`-G` *group*[,*group*]	Place group(s) into project.

projdel

```
projdel name
```

Remove a project.

projects

```
projects -v [user]
```

List projects (or projects of which a user is a member), with descriptions.

newtask

```
newtask -p name
```

Change current project to project *name*.

User Account Management Commands

useradd, usermod, userdel (HP-UX, Linux, Solaris)

```
useradd [options] username     Add a user account.
useradd -D options             Set new account defaults.
usermod options username       Modify a user account.
userdel [-r] username          Remove a user account.
```

These commands add, modify, and remove user accounts. `useradd -D` sets new default values for accounts subsequently created.

Options for useradd and usermod

`-u` *uid* UID (defaults to the next highest unused UID).

`-g` *group* Primary group.

`-G` *groups* Comma-separated list of secondary groups.

`-d` *dir* Full path to home directory (defaults to *current-base-dir/username*).

`-s` *shell* Full path to login shell.

`-c` *full-name*
Full name (user information field).

`-m` Create home directory and copy standard initialization files to it.

`-k` *`dir`* *(*`useradd` *only)*
: Skeleton directory that contains default initialization files (*/etc/skel* is the default). This directory is valid only with `-m`.

`-e` *`yyyy-mm-dd`*
: Account expiration date (default is none).

`-f` *`n`*
: Disable account if inactive for *n* days.

`-p` *`encoded-pwd`* *(Linux)*
: Encoded password (used when importing user accounts from another Unix system's password file).

`-D` *(*`useradd` *only)*
: Set defaults with `-f`, `-e`, `-g`, and `-b` (and `-s` under Linux).

`-b` *`path`* *(*`useradd` *only)*
: Base for home directories (valid only with `-D`).

adduser, chpass, rmuser (FreeBSD)

```
adduser [-s|-v]          Add user account.
chpass user              Modify user account.
rmuser user              Remove user account.
```

`adduser` adds a new user account via a series of prompts (some can also be set via options; see the man page). `-s` requests brief prompts, and `-v` requests verbose ones. `chpass` modifies the specified account via an editor session. `rmuser` removes the specified user account (also via prompts).

adduser defaults: /etc/adduser.conf

`defaultpasswd = yes|no`
: Whether to require passwords for user accounts.

`dotdir = "`*`path`*`"`
: Skeleton directory (defaults to */usr/share/skel*).

`home = "/`*`path`*`"`
: Home directory root (defaults to */home*).

`defaultshell = "`*`name`*`"`
: Login shell (defaults to `tcsh`)

`defaultgroup = `*`group`*`|USER`
: Default group. `USER` requests that a user-private group be created.

`defaultclass = "`*`class-name`*`"`
: Default user class (default is no assigned class).

`uid_start = "`*`n`*`"`
: Lowest UID assigned.

mkuser, chuser, rmuser (AIX)

```
mkuser attribute=value … user
chuser attribute=value … user
rmuser [-p] user
```

Create/modify/remove a user account. The `-p` option to `rmuser` removes the account stanzas from all configurations files, not just the password file.

User account attributes

See also page 44 for account expiration and pages 48–50 for password aging attributes.

`id=`*`n`*
: UID number.

`prgp=`*`group`*
: Primary group.

`groups=`*`list`*
: Group memberships (should include the primary group).

`gecos="`*`full name`*`"`
: User information password file field.

`shell=`*`path`*
: Login shell.

`home=`*`path`*
: Home directory.

login=true|false
: Whether local logins are allowed.

rlogin=true|false
: Whether remote logins are allowed.

daemon=true|false
: Whether user can use cron or the system resource controller.

logintimes=*list*
: Valid login times.

ttys=*list*
: Valid TTY locations.

loginretries=*n*
: Number of login failures after which to lock account.

expire=*MMDDhhmmYY*
: Account expiration date and time.

su=true|false
: Whether other users can su to this account.

sugroups=*list*
: Groups allowed to su to this account.

admin=true|false
: Whether account is an administrative account.

admgroups=*list*
: Groups that the account administers.

umask=*mask*
: Initial umask value.

User Account Attributes

Locking and unlocking a user account

AIX:	chuser account_locked=true|false *user*.
FreeBSD:	chpass -e *user* (use the account expiration date).
HP-UX:	passwd -l *user* (to lock); edit */etc/passwd* to unlock.
Linux:	passwd -l|-u *user*.
Solaris:	passwd -l *user* (to lock); edit */etc/shadow* to unlock.

User account resource limits

AIX: */etc/security/limits*:

```
cpu = seconds
nofiles = n                               Number of open files
fsize, core, data, rss, stack bytes     –1=no limit
```

FreeBSD: */etc/login.conf*:

```
:cputime=seconds:
:maxproc=n:
:openfiles=n:
:priority=nice#:
coredumpsize, datasize, filesize, memoryuse,
memorylocked, sbsize, stacksize bytes|unlimited
```

System-wide initialization files

/etc/profile

Bourne shell, Korn shell, and `bash`. Under Red Hat Linux, the scripts in */etc/profile.d* are also executed.

/etc/csh.cshrc, /etc/csh.login, and /etc/csh.logout

Enhanced C shell (`tcsh`).

/etc/environment and /etc/security/environ (AIX)

Additional sources of environment variable definitions.

/etc/login.conf (FreeBSD)

The `setenv` entry sets environment variables for a class:

```
:setenv=VAR=value[,VAR=value...]:
```

/etc/login.defs (Linux)

Users'/root paths are set via `ENV_PATH` and `ENV_ROOTPATH`.

User Authentication andLogin Controls

Login message files

/etc/motd

Message-of-the-day, displayed after a successful login.

/etc/issue

Pre-login message (HP-UX, Linux, Solaris).

Login process controls

AIX: */etc/security/user*:

`logintimes = ALL|`*`time`*`[,`*`time`*`]`

time is of the form `[!][`*`d`*`[-`*`d`*`]]:`*`start-end`*, where d is the day number (Monday=1). *start* and *end* are four-digit (24-hour) times. An initial `!` functions as a negator.

`ttys = ALL|`*`list`*	*Omit "/dev".*
`loginretries = `*`n`*	*Lock account after n failures.*

/etc/security/login.cfg:

`logintimes` (as in */etc/security/user*)

`logindisable = `*`n`*	*Disable port after n failures.*
`logininterval = `*`seconds`*	*Reset failure count.*
`logindelay = `*`secs`*	*Delay increase per attempt.*
`loginreenable = `*`secs`*	*Unlock locked port.*

FreeBSD: */etc/login.access*:

`+|-:`*`who`*`:`*`origins`*

The initial character, `+` or `-`, grants or denies access, respectively. *who* is a list of user and/or groups to whom the entry applies. *origins* is a list of TTYs, host names/addresses, domain names, and/or subnet addresses. Either list can include the keywords `ALL`, `EXCEPT`, and/or `LOCAL`.

/etc/login.conf:

`:requirehome:`	*Forbid login if home doesn't exist.*

`:times.allow|deny=`*`time`*`[,`*`time`*`...]:`

time format: `[`*`dd`*`[`*`dd`*`…]]`*`start-end`*, where *dd* is a weekday (`Mo`, `Tu`, etc.). *start* and *end* are four-digit (24-hour) times.

`:ttys.`*`xxx`*`=`*`list-of-names`*`:`	*Omit "/dev."*
`:hosts.`*`xxx`*`=`*`host-list`*`:`	*Wildcards ok in list.*
`:`*`yyy`*`time=`*`seconds`*`:`	*Connect time limits.*

yyy can be `day`, `week`, `month` or `session`.

Linux: */etc/login.defs*:

`LOGIN_RETRIES `*`n`*	*Maximum login failures.*
`LOGIN_TIMEOUT `*`secs`*	*Delay between attempts.*
`DEFAULT_HOME yes|no`	*Allow login if home doesn't exist?*

Solaris: */etc/default/login*:

`TIMEOUT=`*`secs`*	*Login attempt timeout.*

`SLEEPTIME=`*`secs`*	*Delay between attempts.*
`SYSLOG=yes|no`	*Log root logins and all failures?*

Account expiration date

AIX:	`chuser expires=`*`MMDDhhwwYY`*
FreeBSD:	`chpass -e "`*`mon day year`*`"`; also, `expireperiod=`*`n`*`w|d` (days/weeks) in */etc/login.conf* (per user class)
Linux:	`chage -E` *`YYYY-MM-DD`*

PAM

PAM is a freely available user authentication facility, currently provided by FreeBSD, HP-UX, Linux, and Solaris.

PAM configuration files: /etc/pam.d

Configuration files are named for the command to which they apply.* The `other` configuration provides defaults for PAM-aware services without their own configuration.

```
entry-type  result-action  module [args]
```

The various entry types are:

`auth`	Specifies procedures for user authentication.
`account`	Set user account attributes and apply account controls.
`password`	Used when a password change is required.
`session`	Configures logging to the `syslog` facility.

The group of entries of a particular type are processed in turn and form a *stack*. The aggregate results of all modules in the stack determine whether access is granted.

Result action keywords

`sufficient`
: Skip any remaining modules in the stack if this module grants access (return authentication success).

* An alternate configuration file, */etc/pam.conf*, can also be used, but the files in */etc/pam.d* take precedence.

requisite
: If the module denies access, return authentication failure and skip any remaining modules in the stack.

required
: The module must grant access for authentication to succeed.

optional
: Results are used only when no other module is deterministic.

Important PAM modules

FreeBSD, HP-UX, and Solaris provide only a few PAM modules, but all modules are open source and can usually be built for these systems. Applicable stacks are in parentheses.

pam_deny *(account, auth, passwd, session)*
pam_permit *(account, auth, passwd, session)*
: Always return failure and success, respectively. Stack either one with pam_warn for logging.

pam_warn *(account, auth, passwd, session)*
: Log information about the calling user and host to syslog.

pam_unix *(account, auth, passwd, session)*
: Verify (account), check aging for (account), or change (password) user passwords.[*] Important options are md5 (use MD5-encoded passwords), shadow (a shadow password file is in use), and try_first_pass (don't prompt for a password if another module has already done so).

pam_cracklib *(passwd)*
: Password triviality checking (stack with pam_unix). The module checks proposed passwords against the words in its dictionary file, */usr/lib/cracklib_dict*, and against the user's previous passwords stored in */etc/security/opasswd*.

: Other options include minlen=*n* (sets a minimum password length of *n*-1 using other default settings; see the man page) and difok=*n* (sets the number of characters in new password that must not be present in the old password, defaults to 10).

pam_limits *(session)*
: Sets user resource limits, as specified in */etc/security/limits.conf*:

[*] There are also several other password checking modules. This is the most common.

```
user-or-group   hard|soft   resource   limit-value
```

The most important resources are `as` (maximum address space), `core` (maximum core file size), `cpu` (CPU time, in minutes), `fsize` (maximum file size), `nofile` (maximum open files), `data`, and `stack` (maximum stack size). All sizes are expressed in kilobytes.

`pam_listfile` *(auth)*

Allow or deny access based on items listed in an external file. Options include:

`sense=allow|deny`	*Grant/deny access.*
`file=`*`path`*	*Location of file.*
`item=`*`what`*	*What the file contains.*

what is one of `user`, `group`, `rhost`, `ruser`, `tty`, or `shell`.

`pam_mkhomedir` *(session)*

Creates the user's home directory if it does not already exist, copying files from */etc/skel* to the new directory.

`pam_nologin` *(auth)*

Prevents non-*root* logins if the file */etc/nologin* exists; the contents of the file are displayed to the user.

`pam_rootok` *(auth)*

Allows *root* access without a password.

`pam_securetty` *(auth)*

Limits *root* access to terminals listed in */etc/securetty*.

`pam_time` *(account)*

Restricts access by time of day, based on user, group, tty, and/or shell, via the configuration in */etc/security/time.conf*:

```
commands; ttys; users; times
```

Each field holds one or more items, joined with `|` (`OR`) or `&` (`AND`); `!` indicates exclusion. *Times* syntax: *`DDstart-end`*; *DD* is one of `Mo`, `Tu`, `We`, `Th`, `Fr`, `Sa`, `Su`, `Wk`, `Wd` (weekday/end), or `Al` (all), and *start* and *end* are four-digit 24-hour times.

Asterisks are wildcards (only one bare wildcard is allowed within the first three fields).

`pam_wheel` *(auth)*

Designed for the `su` facility, this module denies *root* access to users who aren't members of a specified group. Options include:

`group=`*name*	*Applicable group (defaults to GID 0).*
`deny`	*Deny access.*

Solaris PAM modules

`pam_projects` *(account, auth, passwd, session)*
Succeeds if the user belongs to a project and fails otherwise.

`pam_dial_auth` *(account, auth, passwd, session)*
Authenticates dialup logins, using the configuration files */etc/dialup* and */etc/d_passwd* (see the following section).

`pam_roles` *(account, auth, passwd, session)*
Autheticates role changes (see page 5).

Solaris and HP-UX dialup passwords

/etc/dialups
Contains a list of special files that correspond to serial lines on which to accept dialup sessions.

/etc/d_passwd
Per-shell dialup passwords. The entry format is as follows:

```
shell-path:encoded-password:
```

Under HP-UX, use `passwd -F` to modify this file:

```
passwd -F /etc/d_passwd /shell-path
```

Password Selection and Aging

Password lifetimes

Minimum password lifetime

AIX:	`chuser minage=`*wks*	HP-UX:	`passwd -n` *days*
Linux:	`chage -m` *days*	Solaris:	`passwd -n` *days*

Maximum password lifetime

AIX:	`chuser maxage=`*wks*	HP-UX:	`passwd -x` *days*
Linux:	`chage -M` *days*	Solaris:	`passwd -x` *days*
FreeBSD:	`passwordtime=`*days*`d` in */etc/login.conf* (per class)		

Warning period (in advance of upcoming password expiration)

AIX:	`chuser pwdwarntime=`*days*

HP-UX: `passwd -w` *days*   Linux: `chage -W` *days*
Solaris: `passwd -w` *days*
FreeBSD: `warnpassword=`*days*`d` in */etc/login.conf* (user class)

Inactivity period (before account is disabled after password expires)

AIX: `chuser maxexpired=`*days*
Linux: `chage -I` *days*

Set last password change date

FreeBSD: `chpass` (interactive)
Linux: `chage -d` *YYYY-MM-DD* (or local date format)

View current settings

AIX: `lsuser -f`   HP-UX: `passwd -s`
Linux: `chage -l`   Solaris: `passwd -s`

Default settings

AIX: In the `default` stanza of */etc/security/user*

FreeBSD: Settings for the appropriate user class in */etc/login.conf* (or the `default` class)

Linux: */etc/login.defs*:
`PASS_MAX_DAYS` *days*
`PASS_MIN_DAYS` *days*
`PASS_WARN_AGE` *days*
`PASS_MIN_LEN` *n*
`PASS_MAX_LEN` *n*   *Encode only this many password characters.*

Solaris: */etc/default/passwd*:
`MAXWEEKS=`*weeks*
`MINWEEKS=`*weeks*
`WARNWEEKS=`*weeks*
`PASSLENGTH=`*n*   *Minimum password length.*

Password selection triviality checks

AIX account attributes

`minalpha` Minimum number of alphabetic characters.

`minother` Minimum number of nonalphabetic characters.

`mindiff` Minimum number of characters not present in old password.

`maxrepeats`
Maximum number of times any character can appear.

`minlen` Actual minimum password length:

```
minimum(minlen, minalpha+minother)
```

Note that only the first 8 password characters are used.

`dictionlist=`*`file`*`[,`*`file`*`]`
List of files that contain unacceptable passwords.

Linux

Configure via PAM modules (described previously in the "Important PAM modules" section).

FreeBSD: /etc/login.conf

`:minpasswordlen=`*`n`*`:`
Minimum password length.

`:passwd_format=md5:`
Use MD5 encoding (enables passwords > 8 characters).

`:mixpasswordcase=true:`
Disallow all lowercase passwords.

Password history lists

History lists prevent users from reselecting previous passwords.

AIX user account attributes

`histexpire=`*`weeks`*
Time until an old password can be reused (maximum is 260 weeks).

`histsize=`*`n`*
Number of old passwords to remember (maximum is 50).

HP-UX: /etc/default/security

`PASSWORD_HISTORY_DEPTH=`*`n`*
Remember `n` passwords (maximum is 10).

Forcing a password change

AIX:	`pwdadm -f ADMCHG`	FreeBSD:	`chpass` (interactive)
HP-UX:	`passwd -f`	Solaris:	`passwd -f`
Linux:	`chage -d 0` (add `-M 999` if not using aging)		