

NAME

pw - create, remove, modify & display system users and groups

SYNOPSIS

```
pw [-R rootdir] [-V etcdir] useradd [-n] name [-mNoPq] [-C config] [-c comment] [-d homedir]
[-e accexpdate] [-G grouplist] [-g group] [-H fd] [-h fd] [-k skeldir] [-L class] [-M mode]
[-p passexpdate] [-s shell] [-u uid] [-w passmethod] [-Y [-y nispasswd]]

pw [-R rootdir] [-V etcdir] useradd -D [-q] [-b basehome] [-C config] [-e accexpdays] [-G grouplist]
[-g group] [-i mingid,maxgid] [-k skeldir] [-M mode] [-p passexpdays] [-s shell] [-u minuid,maxuid]
[-w passmethod] [-Y [-y nispasswd]]

pw [-R rootdir] [-V etcdir] userdel [-n] name[-u] uid [-r] [-Y [-y nispasswd]]

pw [-R rootdir] [-V etcdir] usermod [-n] name|uid [-u newuid] | -u uid [-mNPq] [-C config]
[-c comment] [-d homedir] [-e accexpdate] [-k skeldir] [-G grouplist] [-g group] [-H fd] [-h fd]
[-L class] [-l newname] [-M mode] [-p passexpdate] [-s shell] [-w passmethod] [-Y [-y nispasswd]]

pw [-R rootdir] [-V etcdir] usershow [-n] name[-u] uid [-7aFP]

pw [-R rootdir] [-V etcdir] usernext [-q] [-C config]

pw [-R rootdir] [-V etcdir] groupadd [-n] name [-oNPqY] [-C config] [-g gid] [-H fd] [-h fd]
[-M members]

pw [-R rootdir] [-V etcdir] groupdel [-n] name[-g] gid [-Y]

pw [-R rootdir] [-V etcdir] groupmod [-n] name|gid [-g newgid] | -g gid [-NPqY] [-C config]
[-d oldmembers] [-H fd] [-h fd] [-l newname] [-M members] [-m newmembers]

pw [-R rootdir] [-V etcdir] groupshow [-n] name[-g] gid [-aFP]

pw [-R rootdir] [-V etcdir] groupnext [-C config] [-q]

pw [-R rootdir] [-V etcdir] lock [-n] name[-u] uid [-q] [-C config]

pw [-R rootdir] [-V etcdir] unlock [-n] name[-u] uid [-q] [-C config]
```

DESCRIPTION

The **pw** utility is a command-line based editor for the system *user* and *group* files, allowing the superuser an easy to use and standardized way of adding, modifying and removing users and groups. Note that **pw** only operates on the local user and group files. NIS users and groups must be maintained on the NIS server. The **pw** utility handles updating the passwd(5), master.passwd(5), group(5) and the secure and insecure password database files, and must be run as root.

The first one or two keywords provided to **pw** on the command line provide the context for the remainder of the arguments. The keywords **user** and **group** may be combined with **add**, **del**, **mod**, **show**, or **next** in any order. (For example, **showuser**, **usershow**, **show user**, and **user show** all mean the same thing.) This flexibility is useful for interactive scripts calling **pw** for user and group database manipulation. Following these keywords, the user or group name or numeric id may be optionally specified as an alternative to using the **-n name**, **-u uid**, **-g gid** options.

The following flags are common to most or all modes of operation:

- R** *rootdir* Specifies an alternate root directory within which **pw** will operate. Any paths specified will be relative to *rootdir*.
- V** *etcdir* Set an alternate location for the password, group, and configuration files. Can be used to maintain a user/group database in an alternate location. If this switch is specified, the system */etc/pw.conf* will not be sourced for default configuration data, but the file *pw.conf* in the specified directory will be used instead (or none, if it does not exist). The **-C** flag may be used to override this behaviour. As an exception to the general rule where options must follow the operation type, the **-V** flag must be used on the command line before the operation keyword.
- C** *config* By default, **pw** reads the file */etc/pw.conf* to obtain policy information on how new user accounts and groups are to be created. The **-C** option specifies a different configuration file. While most of the contents of the configuration file may be overridden via command-line options, it may be more convenient to keep standard information in a configuration file.
- q** Use of this option causes **pw** to suppress error messages, which may be useful in interactive environments where it is preferable to interpret status codes returned by **pw** rather than messing up a carefully formatted display.
- N** This option is available in **add** and **modify** operations, and tells **pw** to output the result of the operation without updating the user or group databases. You may use the **-P** option to switch between standard passwd and readable formats.
- Y** Using this option with any of the update modes causes **pw** to run `make(1)` after changing to the directory */var/yp*. This is intended to allow automatic updating of NIS database files. If separate passwd and group files are being used by NIS, then use the **-y** *nispaswd* option to specify the location of the NIS passwd database so that **pw** will concurrently update it with the system password databases.

USER OPTIONS

The following options apply to the **useradd** and **usermod** commands:

- [-n]** *name* Required unless **-u** *uid* is given. Specify the user/account name. In the case of **usermod** can be a *uid*.
- u** *uid* Required if *name* is not given. Specify the user/account numeric id. In the case of

usermod if paired with *name*, changes the numeric id of the named user/account.

Usually, only one of these options is required, as the account name will imply the uid, or vice versa. However, there are times when both are needed. For example, when changing the uid of an existing user with **usermod**, or overriding the default uid when creating a new account with **useradd**. To automatically allocate the uid to a new user with **useradd**, then do *not* use the **-u** option. Either the account or userid can also be provided immediately after the **useradd**, **userdel**, **usermod**, or **usershow** keywords on the command line without using the **-n** or **-u** options.

- c comment** This field sets the contents of the passwd GECOS field, which normally contains up to four comma-separated fields containing the user's full name, office or location, and work and home phone numbers. These sub-fields are used by convention only, however, and are optional. If this field is to contain spaces, the comment must be enclosed in double quotes `""`. Avoid using commas in this field as these are used as sub-field separators, and the colon `:` character also cannot be used as this is the field separator for the passwd file itself.
- d homedir** This option sets the account's home directory. Normally, this is only used if the home directory is to be different from the default determined from `/etc/pw.conf` - normally `/home` with the account name as a subdirectory.
- e accexpdate**
Set the account's expiration date. Format of the date is either a UNIX time in decimal, or a date in `'dd-mmm-yy[yy]'` format, where `dd` is the day, `mmm` is the month, either in numeric or alphabetic format (`'Jan'`, `'Feb'`, etc) and year is either a two or four digit year. This option also accepts a relative date in the form `'+n[mhdwoy]'` where `'n'` is a decimal, octal (leading 0) or hexadecimal (leading 0x) digit followed by the number of Minutes, Hours, Days, Weeks, Months or Years from the current date at which the expiration date is to be set.
- p passexpdate**
Set the account's password expiration date. This field is similar to the account expiration date option, except that it applies to forced password changes. This is set in the same manner as the **-e** option.
- g group** Set the account's primary group to the given group. *group* may be defined by either its name or group number.
- G grouplist** Set secondary group memberships for an account. *grouplist* is a comma, space, or tab-

separated list of group names or group numbers. The user is added to the groups specified in *grouplist*, and removed from all groups not specified. The current login session is not affected by group membership changes, which only take effect when the user reconnects. Note: do not add a user to their primary group with *grouplist*.

-L class This option sets the login class for the user being created. See `login.conf(5)` and `passwd(5)` for more information on user login classes.

-m This option instructs **pw** to attempt to create the user's home directory. While primarily useful when adding a new account with **useradd**, this may also be of use when moving an existing user's home directory elsewhere on the file system. The new home directory is populated with the contents of the *skeleton* directory, which typically contains a set of shell configuration files that the user may personalize to taste. Files in this directory are usually named *dot.<config>* where the *dot* prefix will be stripped. When **-m** is used on an account with **usermod**, existing configuration files in the user's home directory are *not* overwritten from the skeleton files.

When a user's home directory is created, it will by default be a subdirectory of the *basehome* directory as specified by the **-b** option, bearing the name of the new account. This can be overridden by the **-d** option on the command line, if desired.

-M mode Create the user's home directory with the specified *mode*, modified by the current `umask(2)`. If omitted, it is derived from the parent process' `umask(2)`. This option is only useful in combination with the **-m** flag.

-k skeldir Set the *skeleton* directory, from which basic startup and configuration files are copied when the user's home directory is created. This option only has meaning when used with the **-d** or **-m** flags.

-s shell Set or changes the user's login shell to *shell*. If the path to the shell program is omitted, **pw** searches the *shellpath* specified in `/etc/pw.conf` and fills it in as appropriate. Note that unless you have a specific reason to do so, you should avoid specifying the path - this will allow **pw** to validate that the program exists and is executable. Specifying a full path (or supplying a blank "" shell) avoids this check and allows for such entries as */nonexistent* that should be set for accounts not intended for interactive login.

-h fd This option provides a special interface by which interactive scripts can set an account password using **pw**. Because the command line and environment are fundamentally insecure mechanisms by which programs can accept information, **pw** will only allow setting of account and group passwords via a file descriptor (usually a pipe between an

interactive script and the program). *sh*, *bash*, *ksh* and *perl* all possess mechanisms by which this can be done. Alternatively, **pw** will prompt for the user's password if **-h 0** is given, nominating *stdin* as the file descriptor on which to read the password. Note that this password will be read only once and is intended for use by a script rather than for interactive use. If you wish to have new password confirmation along the lines of `passwd(1)`, this must be implemented as part of an interactive script that calls **pw**.

If a value of '-' is given as the argument *fd*, then the password will be set to '*', rendering the account inaccessible via password-based login.

-H *fd* Read an encrypted password string from the specified file descriptor. This is like **-h**, but the password should be supplied already encrypted in a form suitable for writing directly to the password database. See `openssl-passwd(1)` and `crypt(3)` for more details about generating an encrypted password hash.

It is possible to use **useradd** to create a new account that duplicates an existing user id. While this is normally considered an error and will be rejected, the **-o** option overrides the check for duplicates and allows the duplication of the user id. This may be useful if you allow the same user to login under different contexts (different group allocations, different home directory, different shell) while providing basically the same permissions for access to the user's files in each account.

The **useradd** command also has the ability to set new user and group defaults by using the **-D** option. Instead of adding a new user, **pw** writes a new set of defaults to its configuration file, `/etc/pw.conf`. When using the **-D** option, you must not use either **-n name** or **-u uid** or an error will result. Use of **-D** changes the meaning of several command line switches in the *useradd* command. These are:

-D Set default values in `/etc/pw.conf` configuration file, or a different named configuration file if the **-C config** option is used.

-b *basehome*

Set the root directory in which user home directories are created. The default value for this is `/home`, but it may be set elsewhere as desired.

-e *accexpdays*

Set the default account expiration period in days. When **-D** is used, the *accexpdays* argument is interpreted differently. It must be numeric and represents the number of days after creation that the account expires. A value of 0 suppresses automatic calculation of the expiry date.

-p *passexpdays*

Set the default password expiration period in days. When **-D** is used, the *passexpdays* argument is interpreted differently. It must be numeric and represents the number of days after creation that the account expires. A value of 0 suppresses automatic calculation of the expiry date.

- g group** Set the default group for new users. If a blank group is specified using **-g ""**, then new users will be allocated their own private primary group with the same name as their login name. If a group is supplied, either its name or uid may be given as an argument.
- G grouplist** Set the default groups in which new users are granted membership. This is a separate set of groups from the primary group. Avoid nominating the same group as both primary and extra groups. In other words, these extra groups determine membership in groups *other than* the primary group. *grouplist* is a comma-separated list of group names or ids, and are always stored in */etc/pw.conf* by their symbolic names.
- L class** This option sets the default login class for new users.
- k skeldir** Set the default *skeleton* directory, from which prototype shell and other initialization files are copied when **pw** creates a user's home directory. See description of **-k** for naming conventions of these files.
- u minuid,maxuid, -i mingid,maxgid**
Set the minimum and maximum user and group ids allocated for new accounts and groups created by **pw**. The default values for each is 1000 minimum and 32000 maximum. *minuid* and *maxuid* are both numbers, where max must be greater than min, and both must be between 0 and 32767 (the same applies to *mingid* and *maxgid*). In general, user and group ids less than 100 are reserved for use by the system, and numbers greater than 32000 may also be reserved for special purposes (used by some system daemons).
- w passmethod**
The **-w** option selects the default method used to set passwords for newly created user accounts. *passmethod* is one of:

- no** disable login on newly created accounts
- yes** force the password to be the account name
- none** force a blank password
- random** generate a random password

The **random** or **no** methods are the most secure; in the former case, **pw** generates a password and prints it to stdout, which is suitable when users are issued passwords rather

than being allowed to select their own (possibly poorly chosen) password. The **no** method requires that the superuser use `passwd(1)` to render the account accessible with a password.

-y path This sets the pathname of the database used by NIS if you are not sharing the information from `/etc/master.passwd` directly with NIS. You should only set this option for NIS servers.

The **userdel** command has three distinct options. The **-n name** and **-u uid** options have already been covered above. The additional option is:

-r This tells **pw** to remove the user's home directory and all of its contents. The **pw** utility errs on the side of caution when removing files from the system. Firstly, it will not do so if the uid of the account being removed is also used by another account on the system, and the "home" directory in the password file is a valid path that commences with the character `'/'`. Secondly, it will only remove files and directories that are actually owned by the user, or symbolic links owned by anyone under the user's home directory. Finally, after deleting all contents owned by the user only empty directories will be removed. If any additional cleanup work is required, this is left to the administrator.

Mail spool files and `crontab(5)` files are always removed when an account is deleted as these are unconditionally attached to the user name. Jobs queued for processing by `at(1)` are also removed if the user's uid is unique and not also used by another account on the system.

The **usermod** command adds one additional option:

-l newname This option allows changing of an existing account name to *newname*. The new name must not already exist, and any attempt to duplicate an existing account name will be rejected.

The **usershow** command allows viewing of an account in one of two formats. By default, the format is identical to the format used in `/etc/master.passwd` with the password field replaced with a `'*'`. If the **-P** option is used, then **pw** outputs the account details in a more human readable form. If the **-7** option is used, the account details are shown in v7 format. The **-a** option lists all users currently on file. Using **-F** forces **pw** to print the details of an account even if it does not exist.

The command **usernext** returns the next available user and group ids separated by a colon. This is normally of interest only to interactive scripts or front-ends that use **pw**.

GROUP OPTIONS

The **-C** and **-q** options (explained at the start of the previous section) are available with the group

manipulation commands. Other common options to all group-related commands are:

[-n] name Required unless **-g gid** is given. Specify the group name. In the case of **groupmod** can be a gid.

-g gid Required if *name* is not given. Specify the group numeric id. In the case of **groupmod** if paired with *name*, changes the numeric id of the named group.

As with the account name and id fields, you will usually only need to supply one of these, as the group name implies the uid and vice versa. You will only need to use both when setting a specific group id against a new group or when changing the uid of an existing group.

-M memberlist This option provides an alternative way to add existing users to a new group (in **groupadd**) or replace an existing membership list (in **groupmod**). *memberlist* is a comma separated list of valid and existing user names or uids.

-m newmembers Similar to **-M**, this option allows the *addition* of existing users to a group without replacing the existing list of members. Login names or user ids may be used, and duplicate users are silently eliminated.

-d oldmembers Similar to **-M**, this option allows the *deletion* of existing users from a group without replacing the existing list of members. Login names or user ids may be used, and duplicate users are silently eliminated.

groupadd also has a **-o** option that allows allocation of an existing group id to a new group. The default action is to reject an attempt to add a group, and this option overrides the check for duplicate group ids. There is rarely any need to duplicate a group id.

The **groupmod** command adds one additional option:

-l newname This option allows changing of an existing group name to *newname*. The new name must not already exist, and any attempt to duplicate an existing group name will be rejected.

Options for **groupshow** are the same as for **usershow**, with the **-g gid** replacing **-u uid** to specify the group id. The **-7** option does not apply to the **groupshow** command.

The command **groupnext** returns the next available group id on standard output.

USER LOCKING

The **pw** utility supports a simple password locking mechanism for users; it works by prepending the string `*LOCKED*` to the beginning of the password field in `master.passwd(5)` to prevent successful authentication.

The **lock** and **unlock** commands take a user name or uid of the account to lock or unlock, respectively. The **-V**, **-C**, and **-q** options as described above are accepted by these commands.

NOTES

For a summary of options available with each command, you can use

```
pw [command] help
```

For example,

```
pw useradd help
```

lists all available options for the **useradd** operation.

The **pw** utility allows 8-bit characters in the `passwd` GECOS field (user's full name, office, work and home phone number subfields), but disallows them in user login and group names. Use 8-bit characters with caution, as connection to the Internet will require that your mail transport program supports 8BITMIME, and will convert headers containing 8-bit characters to 7-bit quoted-printable format. `sendmail(8)` does support this. Use of 8-bit characters in the GECOS field should be used in conjunction with the user's default locale and character set and should not be implemented without their use. Using 8-bit characters may also affect other programs that transmit the contents of the GECOS field over the Internet, such as `finger(8)`, and a small number of TCP/IP clients, such as IRC, where full names specified in the `passwd` file may be used by default.

The **pw** utility writes a log to the `/var/log/userlog` file when actions such as user or group additions or deletions occur. The location of this logfile can be changed in `pw.conf(5)`.

FILES

<i>/etc/master.passwd</i>	The user database
<i>/etc/passwd</i>	A Version 7 format password file
<i>/etc/login.conf</i>	The user capabilities database
<i>/etc/group</i>	The group database
<i>/etc/pw.conf</i>	Pw default options file
<i>/var/log/userlog</i>	User/group modification logfile

EXAMPLES

Add new user Glurmo Smith (`gsmith`). A `gsmith` login group is created if not already present. The login shell is set to `cs(1)`. A new home directory at `/home/gsmith` is created if it does not already exist.

Finally, a random password is generated and displayed:

```
pw useradd -n gsmith -c "Glurmo Smith" -s csh -m -w random
```

Delete the gsmith user and their home directory, including contents.

```
pw userdel -n gsmith -r
```

Add the existing user jsmith to the wheel group, in addition to the other groups jsmith is already a member of.

```
pw groupmod wheel -m jsmith
```

Generate random password and show it in both plain text and encrypted form not modifying any database.

```
pw usermod nobody -Nw random
```

EXIT STATUS

The **pw** utility returns EXIT_SUCCESS on successful operation, otherwise **pw** returns one of the following exit codes defined by sysexit(3) as follows:

EX_USAGE

- Command line syntax errors (invalid keyword, unknown option).

EX_NOPERM

- Attempting to run one of the update modes as non-root.

EX_OSERR

- Memory allocation error.
- Read error from password file descriptor.

EX_DATAERR

- Bad or invalid data provided or missing on the command line or via the password file descriptor.
- Attempted to remove, rename root account or change its uid.

EX_OSFILE

- Skeleton directory is invalid or does not exist.
- Base home directory is invalid or does not exist.
- Invalid or non-existent shell specified.

EX_NOUSER

- ⊕ User, user id, group or group id specified does not exist.
- ⊕ User or group recorded, added, or modified unexpectedly disappeared.

EX_SOFTWARE

- ⊕ No more group or user ids available within specified range.

EX_IOERR

- ⊕ Unable to rewrite configuration file.
- ⊕ Error updating group or user database files.
- ⊕ Update error for passwd or group database files.

EX_CONFIG

- ⊕ No base home directory configured.

SEE ALSO

chpass(1), passwd(1), umask(2), group(5), login.conf(5), passwd(5), pw.conf(5), pwd_mkdb(8), vipw(8)

HISTORY

The **pw** utility was written to mimic many of the options used in the SYSV *shadow* support suite, but is modified for passwd and group fields specific to the 4.4BSD operating system, and combines all of the major elements into a single command.