NAME

pwd_mkdb - generate the password databases

SYNOPSIS

pwd_mkdb [**-CiNp**] [**-d** *directory*] [**-s** *cachesize*] [**-u** *username*] *file*

DESCRIPTION

The **pwd_mkdb** utility creates db(3) style secure and insecure databases for the specified file. These databases are then installed into */etc/spwd.db* and */etc/pwd.db* respectively. The file is installed into */etc/master.passwd*. The file must be in the correct format (see passwd(5)). It is important to note that the format used in this system is different from the historic Version 7 style format.

The options are as follows:

-C Check if the password file is in the correct format. Do not change, add, or remove any files.

-d *directory*
Store databases into specified destination directory instead of */etc*.

-i Ignore locking failure of the *master.passwd* file. This option is intended to be used to build password files in the release process over NFS where no contention can happen. A non-default directory must also be specified with the **-d** option for locking to be ignored. Other use of this option is strongly discouraged.

-N Tell **pwd_mkdb** to exit with an error if it cannot obtain a lock on the file. By default, we block waiting for a lock on the source file. The lock is held through the rebuilding of the database.

-p Create a Version 7 style password file and install it into */etc/passwd*.

-s *cachesize*
Specify in megabytes the size of the memory cache used by the hashing library. On systems with a large user base, a small cache size can lead to prohibitively long database file rebuild times. As a rough guide, the memory usage of **pwd_mkdb** in megabytes will be a little bit more than twice the figure specified here. The default is 2 megabytes.

-u *username*
Only update the record for the specified user. Utilities that operate on a single user can use this option to avoid the overhead of rebuilding the entire database.

The two databases differ in that the secure version contains the user's encrypted password and the

insecure version has an asterisk (''*'')

The databases are used by the C library password routines (see getpwent(3)).

The **pwd_mkdb** utility exits zero on success, non-zero on failure.

ENVIRONMENT

If the PW_SCAN_BIG_IDS environment variable is set, **pwd_mkdb** will suppress the warning messages that are normally generated for large user and group IDs. Such IDs can cause serious problems with software that makes assumptions about the values of IDs.

FILES

/etc/pwd.db	The insecure password database file.
/etc/pwd.db.tmp	A temporary file.
/etc/spwd.db	The secure password database file.
/etc/spwd.db.tmp	A temporary file.
/etc/master.passwd	The current password file.
/etc/passwd	A Version 7 format password file.

EXAMPLES

Regenerate the password database after manually editing or replacing the password file:

```
/usr/sbin/pwd_mkdb -p /etc/master.passwd
```

COMPATIBILITY

Previous versions of the system had a program similar to **pwd_mkdb**, mkpasswd(8), which built dbm(3) style databases for the password file but depended on the calling programs to install them. The program was renamed in order that previous users of the program not be surprised by the changes in functionality.

SEE ALSO

chpass(1), passwd(1), db(3), getpwent(3), passwd(5), vipw(8)

BUGS

Because of the necessity for atomic update of the password files, **pwd_mkdb** uses rename(2) to install them. This, however, requires that the file specified on the command line live on the same file system as the */etc* directory.

There are the obvious races with multiple people running **pwd_mkdb** on different password files at the same time. The front-ends to **pwd_mkdb**, chpass(1), passwd(1) and vipw(8), handle the locking

necessary to avoid this problem.