pass Unix password manager

CONCEPTS

Each password is one GPG-encrypted file under ~/.password-store; the folder tree is your organisation.

Plain text typed or generated
↓ `pass insert`
GPG file name.gpg in the store
↓ `pass git push`
Git remote optional backup

The first line of a file is the password; put extra data (user, URL) on later lines.

Back up your GPG secret key: without it the store cannot be decrypted.

SETUP

`pass init` *`<gpg-id>`*
Create the store for a GPG key

`pass init -p` *`<subfolder>`* *`<gpg-id>`*
Use a different key for a folder

`pass git init`
Turn the store into a git repo

`pass git remote add origin` *`<url>`*
Add a remote to the store

`PASSWORD_STORE_DIR=`*`<dir>`* **`pass`**
Use another store location

`PASSWORD_STORE_ENABLE_EXTENSIONS=true`
Enable user extensions

BROWSE

`pass`
List all entries as a tree

`pass ls` *`<folder>`*
List one folder

`pass find` *`<name>`*
List entries whose name matches

`pass grep` *`<text>`*
Search inside decrypted entries

`pass grep -i` *`<text>`*
Search ignoring case

READ

`pass show` *`<name>`*
Print an entry

`pass` *`<name>`*
Print an entry (shortcut)

`pass -c` *`<name>`*
Copy the first line to the clipboard

`pass -c`*`<N>`* *`<name>`*
Copy line N instead

`pass -q` *`<name>`*
Show as a QR code

`PASSWORD_STORE_CLIP_TIME=`*`<sec>`*
Change the clipboard timeout

ADD AND EDIT

`pass insert` *`<name>`*
Add an entry, typed twice hidden

`pass insert -m` *`<name>`*
Add a multi-line entry

`pass insert -e` *`<name>`*
Add an entry with visible input

`pass insert -f` *`<name>`*
Overwrite without asking

`pass edit` *`<name>`*
Edit an entry in $EDITOR

`pass generate` *`<name>`* *`<length>`*
Generate a random password

`pass generate -n` *`<name>`* *`<length>`*
Generate without symbols

`pass generate -c` *`<name>`* *`<length>`*
Generate and copy, do not print

`pass generate -i` *`<name>`* *`<length>`*
Replace only the first line

`pass generate -f` *`<name>`* *`<length>`*
Overwrite without asking

`PASSWORD_STORE_GENERATED_LENGTH=`*`<n>`*
Default generated length

ORGANISE

`pass mv` *`<old>`* *`<new>`*
Rename or move an entry

ORGANISE (cont.)

`pass cp` *`<old>`* *`<new>`*
Copy an entry

`pass rm` *`<name>`*
Delete an entry

`pass rm -r` *`<folder>`*
Delete a folder

`pass mv` *`<name>`* *`<folder>`***`/`**
Move into a folder

GIT SYNC

`pass git log`
Show store history

`pass git push`
Push the store

`pass git pull`
Pull changes from the remote

`pass git status`
Check for uncommitted changes

`pass git clone` *`<url>`* **`~/.password-store`**
Restore on a new machine

TEAMS AND SECURITY

`pass init` *`<id1>`* *`<id2>`*
Encrypt for several recipients

Entry names are visible in the file tree and git history; only the contents are encrypted.

RECIPES

First setup

```
gpg --list-secret-keys --keyid-format long
pass init <gpg-id>
pass git init
pass generate email/work 24
```

Sync to a new machine

```
gpg --import <secret-key-file>
pass git clone <url> ~/.password-store
pass -c email/work
```

Store extra fields

```
pass insert -m site/example
# line 1: password
# line 2: user: me@example.com
```

Rotate a password

```
pass generate -i site/example 24
pass git push
```