



Full credit is given to the above companies including the Operating System (OS) that this PDF file was generated!

Rocky Enterprise Linux 9.2 Manual Pages on command 'git-credential-store.1'

\$ man git-credential-store.1

GIT-CREDENTIAL-STO(1) Git Manual GIT-CREDENTIAL-STO(1)

NAME

git-credential-store - Helper to store credentials on disk

SYNOPSIS

git config credential.helper 'store [<options>']

DESCRIPTION

Note

Using this helper will store your passwords unencrypted on disk, protected only by filesystem permissions. If this is not an acceptable security tradeoff, try git-credential-cache(1), or find a helper that integrates with secure storage provided by your operating system.

This command stores credentials indefinitely on disk for use by future Git programs.

You probably don't want to invoke this command directly; it is meant to be used as a credential helper by other parts of git. See gitcredentials(7) or EXAMPLES below.

OPTIONS

--file=<path>

Use <path> to lookup and store credentials. The file will have its filesystem permissions set to prevent other users on the system from reading it, but will not be encrypted or otherwise protected. If not specified, credentials will be searched for from ~/.git-credentials and \$XDG_CONFIG_HOME/git/credentials, and credentials will be written to ~/.git-credentials if it exists, or \$XDG_CONFIG_HOME/git/credentials if it exists and the former does not. See also the section called ?FILES?.

FILES

If not set explicitly with `--file`, there are two files where `git-credential-store` will search for credentials in order of precedence:

`~/.git-credentials`

User-specific credentials file.

`$XDG_CONFIG_HOME/git/credentials`

Second user-specific credentials file. If `$XDG_CONFIG_HOME` is not set or empty, `$HOME/.config/git/credentials` will be used. Any credentials stored in this file will not be used if `~/.git-credentials` has a matching credential as well. It is a good idea not to create this file if you sometimes use older versions of Git that do not support it.

For credential lookups, the files are read in the order given above, with the first matching credential found taking precedence over credentials found in files further down the list.

Credential storage will by default write to the first existing file in the list. If none of these files exist, `~/.git-credentials` will be created and written to.

When erasing credentials, matching credentials will be erased from all files.

EXAMPLES

The point of this helper is to reduce the number of times you must type your username or password. For example:

```
$ git config credential.helper store
```

```
$ git push http://example.com/repo.git
```

```
Username: <type your username>
```

```
Password: <type your password>
```

```
[several days later]
```

```
$ git push http://example.com/repo.git
```

```
[your credentials are used automatically]
```

STORAGE FORMAT

The `.git-credentials` file is stored in plaintext. Each credential is stored on its own line as a URL like:

```
https://user:pass@example.com
```

No other kinds of lines (e.g. empty lines or comment lines) are allowed in the file, even though some may be silently ignored. Do not view or edit the file with editors.

When Git needs authentication for a particular URL context, `credential-store` will consider

that context a pattern to match against each entry in the credentials file. If the protocol, hostname, and username (if we already have one) match, then the password is returned to Git. See the discussion of configuration in gitcredentials(7) for more information.

GIT

Part of the git(1) suite

Git 2.34.1

07/07/2023

GIT-CREDENTIAL-STO(1)