

**NAME**

git-credential-cache – Helper to temporarily store passwords in memory

**SYNOPSIS**

```
git config credential.helper 'cache [<options>']
```

**DESCRIPTION**

This command caches credentials in memory for use by future Git programs. The stored credentials never touch the disk, and are forgotten after a configurable timeout. The cache is accessible over a Unix domain socket, restricted to the current user by filesystem permissions.

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**

`--timeout <seconds>`

Number of seconds to cache credentials (default: 900).

`--socket <path>`

Use **<path>** to contact a running cache daemon (or start a new cache daemon if one is not started).

Defaults to `$XDG_CACHE_HOME/git/credential/socket` unless `~/.git-credential-cache/` exists in which case `~/.git-credential-cache/socket` is used instead. If your home directory is on a network-mounted filesystem, you may need to change this to a local filesystem. You must specify an absolute path.

**CONTROLLING THE DAEMON**

If you would like the daemon to exit early, forgetting all cached credentials before their timeout, you can issue an **exit** action:

```
git credential-cache exit
```

**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 cache
$ git push http://example.com/repo.git
Username: <type your username>
Password: <type your password>
```

```
[work for 5 more minutes]
$ git push http://example.com/repo.git
[your credentials are used automatically]
```

You can provide options via the `credential.helper` configuration variable (this example drops the cache time to 5 minutes):

```
$ git config credential.helper 'cache --timeout=300'
```

**GIT**

Part of the **git(1)** suite