NAME
> **git−lfs−clone** – Efficiently clone a LFS−enabled repository

SYNOPSIS
> **git lfs clone** [git clone options] *repository* [*directory*]

DESCRIPTION
> Clone an LFS enabled Git repository more efficiently by disabling LFS during the git clone, then perform-ing a 'git lfs pull' directly afterwards.
>
> 'git lfs clone' also installs all of the repo−level hooks (.git/hooks) that LFS requires to operate. If **−−sepa-rate−git−dir** is given, the hooks will be installed there.
>
> This is faster than a regular 'git clone' because that will download LFS content using the smudge filter, which is executed individually per file in the working copy. This is relatively inefficient compared to the batch mode and parallel downloads performed by 'git lfs pull'.

OPTIONS
> All options supported by 'git clone'
>
> **−I** *paths* **−−include=***paths*
>> See *INCLUDE AND EXCLUDE*
>
> **−X** *paths* **−−exclude=***paths*
>> See *INCLUDE AND EXCLUDE*
>
> **−−skip−repo**
>> Skip installing repo−level hooks (.git/hooks) that LFS requires. Disabled by default.

INCLUDE AND EXCLUDE
> You can configure Git LFS to only fetch objects to satisfy references in certain paths of the repo, and/or to exclude certain paths of the repo, to reduce the time you spend downloading things you do not use.
>
> In lfsconfig, set lfs.fetchinclude and lfs.fetchexclude to comma−separated lists of paths to include/exclude in the fetch (wildcard matching as per gitignore). Only paths which are matched by fetchinclude and not matched by fetchexclude will have objects fetched for them.
>
> Note that using the command−line options **−I** and **−X** override the respective configuration settings.

SEE ALSO
> git−clone(1), git−lfs−pull(1).
>
> Part of the git−lfs(1) suite.