

Performance Tuning

Refreshing

There are a couple of possible reasons which may result in slower refreshing:

- **Insufficient heap size:** try to increase the [maximum memory limit](#).
- **Network Shares:** make sure, your repository and Git's HOME directory is located on a local drive.
- **Symlinks:** try to open the repository from its real location. When opening a repository over a symlink (Windows, too!), refreshing may be slower.
- **Background processes (since version 6):** in the **Preferences**, section **Background Commands**, try to disable all options related to **R efreshing and Local and Remote Changes**.
- **Big Repositories:** if SmartGit is close to the maximum memory limit, refreshing may become very slow. Usually, the pre-configured memory limit is sufficient, however for large repositories it may be too low. Try to increase the limit according to [these instructions](#).
- **.git/index timestamps wrong:** sometimes `.git/index` timestamps may be incorrectly set/rounded, resulting in a content comparison for all your working tree files. A possible solution may be to delete `.git/index` and then invoke `git reset --hard` from the command line.

Warning

Be sure that you have no *local* nor *staged* changes, before doing so!

If you encounter wrong `.git/index` timestamps frequently, please contact smartgit@syntevo.com

If an invocation of `git status` is *rather* slow, too, you may also want to try the following:

- **Switch off ctime change detection:** invoke `git config core.trustctime false` in your repository root to ignore the files' ctimes changes when scanning for modified files. For details, refer to the [git-config man page](#).
- **Inefficient .gitignore:** if you have a large amount of ignored files, be sure they will be located in a directory which itself is marked as ignored, not just the contained files. Also, be sure that there is no tracked (versioned) file contained in this directory. If you are encountering this problem, too, please comment and vote for [Topic #630](#).

Example

If many of your ignored files are located directly in the `out/`-directory, following `.gitignore` line will be *efficient*:

```
out/
```

while this line will be *inefficient*:

```
out/*
```

Furthermore, if there is just a single tracked file located in `out/` or one of its subdirectories, the ignore-processing will be *inefficient*.

Many Refs

Large amounts of tags (or branches) may slow down Logs, but also the Working Tree window. You can tell SmartGit to exclude certain tags or remote branches from loading by using [Java regular expressions](#) with following options in `.git/config`:

```
[smartgit "refLimit"]
  tagIncludeRegex = smart.*
tagExcludeRegex = .*-I\\d{8}-\\d{4}
remoteBranchIncludeRegex =
remoteBranchExcludeRegex =
```

If an include regular expression matches the tag/remote branch name, it will be included, no matter whether the excluding pattern matches.

It is recommended to invoke a `git pack-refs --all` after doing so, to have the changes take effect in SmartGit.

Log

In addition to the above mentioned problems:

- If opening the Log takes a long time, make sure that you deselect as many **Branches** as possible, in the ideal case only leaving **HEAD** selected. Once reopening the Log, it should be faster.

If nothing of the above helps, shutdown SmartGit, get rid of all `log.txt*`-files from the Settings directory (see About dialog), restart SmartGit, capture the slowness and send resulting `log.txt*`-files to smartgit@syntevo.com .

Miscellaneous

- We have received user reports that SmartGit's user interface may be slow if **SuperPuTTY** version 1.0.4.6 (or lower) is **running**. Especially, closing SmartGit may take several seconds.