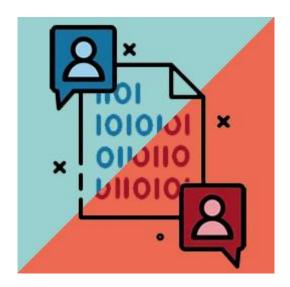# GIT LFS LOCKS

Effective plugin for conflict protection. You don't need to write a message to the entire team that you are working on the scene to avoid conflicts. Git LFS Locks plugin will take care of it.

Changelog

1.0.2

- Added lock suggestions popup for not locked assets
- Fixed reopening git locks window after every git lock process
- Fixed serialization repository URL per project
- Fixed scene view notification about scene lock status

1.0.1

- Added macOS support.
- Changed DISABLE_GIT_LOCKS define symbol to disable whole git lock feature.
- Fixed crash while building in batch mode.

Package Dependencies

- Editor Coroutines (com.unity.editorcoroutines) 1.0.0
- Newtonsoft JSON (com.unity.nuget.newtonsoft-json) 3.0.2

What is Git LFS Locks plugin?

The plugin adds to unity support for Git LFS Locks, which you can lock editing for other repository users of selected files from the repository directly from Unity.
Thanks to this solution, you can avoid conflicts on binary files such as scenes, prefabs and some assets. This plugin works great when more people are working on a project and we want to avoid difficult to solve file conflicts. In unity, we have a live preview in a dedicated editor window of what files are locked and by who.

Additionally, each locked assets and scenes have information about locks in the header of the inspector. Scenes that are the most conflict-sensitive have a special way of alerting by displaying notification about locks on Scene View as soon as the scene is opened. When trying to save changes to a locked file, a popup message is displayed that someone is working on this file and is locked. When we close the editor, a notification will be displayed about the locks we have so that we do not accidentally end work with the locked files. The plugin gives you some additional shortcuts with which you can directly from unity discard, stash, checkout changes on selected files.

API documentation is available at the link https://krelistudio.pl/git-lfs-locks/.

Key Features of Git LFS Locks package

- Main feature is to prevent conflicts on your scenes, prefabs and other assets while you are working on a project in a team.
- Live preview what is locked.
- Discard changes inside Unity.
- Shortcut to open file history inside Unity.
- Notifications to warn users if a scene or asset are locked by someone else.
- You do not need to write a message to the entire team that you are working on the scene to avoid conflicts. Git LFS Locks plugin will take care of it.

What is Git, Git LFS and Git LFS Locks?

1. What is Git?

Git is a free and open source distributed version control system designed to handle everything from small to very large projects with speed and efficiency. More about Git https://git-scm.com/

2. What is Git LFS?

An open source Git extension for versioning large files. Git Large File Storage (LFS) replaces large files such as audio samples, videos, datasets, and graphics with text pointers inside Git, while storing the file contents on a remote server like GitHub.com or GitHub Enterprise. More about Git LFS https://git-lfs.github.com/

3. What is Git LFS Locks?

File Locking lets developers lock files they are updating to prevent other users from updating them at the same time. Concurrent edits in Git repositories will lead to merge conflicts, which are very difficult to resolve in large binary files. More about Git LFS Locks https://github.com/git-lfs/git-lfs/wiki/File-Locking

Kreli Studio

How to setup Git LFS Locks plugin?

1. Setup Git

To install Git on your computer you have to navigate to Git installer
https://git-scm.com/downloads and download the latest version. Once the installer has
started, follow the instructions as provided in the Git Setup wizard screen until the
installation is complete.

2. Setup Git LFS

To install Git LFS you have to navigate to https://github.com/git-lfs/git-lfs/releases then
find a list of latest version released assets and download it. Once the installer has started,
follow the instructions as provided in the Git LFS Setup wizard screen until the installation
is complete. More about installation process Git LFS can be find here
https://github.com/git-lfs/git-lfs#getting-started

3. Setup Git LFS Locks plugin

Git LFS Locks plugin requires installed Git, Git LFS extension and Unity project to be git
repository. About package dependencies read #Package Dependencies section. Plugin does
not need any other installation or preparation process to work.

How to use Git LFS Locks plugin?

1. How open Git locks window

Select from the toolbar "Tools/Git/Git Locks Window".
Lock/unlock any asset and the Git locks window will be automatically open next to the hierarchy tab.

2. How to lock or unlock assets?

Click the right mouse button on the asset in the project tab and then select "Git/Lock" or "Git/Unlock".
Open the Git locks window and click the right mouse button on any locked asset and then select unlock.

3. How can I force unlock not my locks?

First you must have permissions setup on repository to do this operation.
Open the Git locks window and click the right mouse button on any locked asset and then select force unlock.

4. How can I check which assets are locked and by who?

Locked assets have that information in inspector header.
Open the Git locks window and all displayed assets there are locked then select one of them and detailed information will be displayed on the bottom of the window.
Lock notification will be displayed in the scene view right after you open the locked scene.

5. How can I discard asset changes directly from unity?

Click right mouse button on asset in project tab and then select "Git/Discard Changes".
Open the Git locks window and click the right mouse button on any locked asset and then select discard changes.

6. How can I stash asset changes directly from unity?

Click right mouse button on asset in project tab and then select "Git/Stash Changes".
Open the Git locks window and click the right mouse button on any locked asset and then select stash changes.

7. How can I LFS checkout asset directly from unity?

Click right mouse button on asset in project tab and then select "Git/LFS Checkout".
Open the Git locks window and click the right mouse button on any locked asset and then select LFS checkout.

8. How can I display asset history directly from unity?

First open the Git locks window and click on the gear icon. Set the repository url and then use one of the instructions below.
Click right mouse button on asset in project tab and then select "Git/Show File History".
Open the Git locks window and click the right mouse button on any locked asset and then select show file history.
Select asset and click left mouse button on authors label in inspector header.

9. How can I disable Git LFS Locks features locally?

Open Player Settings and add new scripting define symbol "DISABLE_GIT_LOCKS"

10. How can I write my own Git features depending on the Git LFS Locks plugin?

Whole magic is inside GitCommands class. Use the "ExecuteGitCommandCoroutine" method to invoke any Git command.

11. Where can I find API documentation?

https://krelistudio.pl/git-lfs-locks/

12. Git LFS Locks doesn't work. What should I check first?

Make sure your project is a git repository.
Make sure you have installed latest version of Git and Git LFS.
Make sure your project is open in Unity 2021.3.2 or higher.
Make sure your project has require packages. Read #Package Dependencies section for more information.
Make sure your git account has all require permissions to repository.

13. Should I disable Git LFS Locks feature for CI builds?

I recommend disable Git LFS Locks feature for CI builds and while running Unity in batch mode because sometimes async git process and opening unity windows can crash.
You can disable Git LFS Locks feature for CI builds by define additional scripting define in "BuildPlayerOptions.extraScriptingDefines". More info:
https://docs.unity3d.com/ScriptReference/BuildPlayerOptions-extraScriptingDefines.html

14. Can I disable popup about lock suggestions for modified assets?

Yes, you can toggle it by menu "Tools/Git/Lock Suggestions Popup".