

Consistent

Line Endings (Free)

By Nightfox

Consistent

Line Endings^{FREE}

Code. Normalize. Relax.



Auto-fix your line ending styles, depending on your preference (Win, Unix, Mac).

NOTE:

Please create a backup of your project before using this asset.

What is Consistent Line Endings?

Consistent Line Endings automatically normalizes all inconsistent line-endings in your script and text files, allowing you to develop without dealing with annoying warnings every time you open them up.

What else does it do?

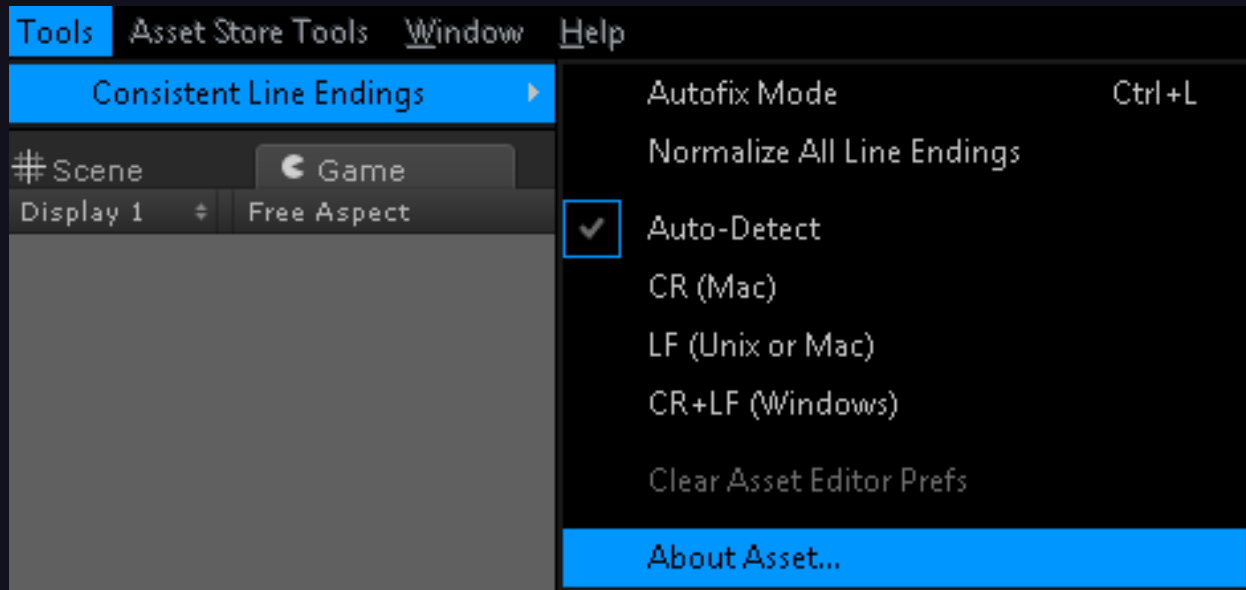
Apart from normalizing your line-endings, you can also choose to convert them into OS-specific styles (CL, LF, CR+LF).

How can we use it?

You can start normalizing inconsistent line-endings in your files in two ways:

1. Through the Tools menu (Auto + Normalize)
2. Through the Assets menu (Normalize)

Through the Tools menu:



Consistent Line Endings Tools Menu

Go to Tools > Consistent Line Endings.

Choose Autofix Mode if you want to normalize automatically. You can toggle this mode by clicking **Ctrl + L**. Once this is on, line endings of files will be automatically normalized after every script / file changes. To turn this mode off, simply toggle it again.

On the other hand, if you just want to normalize once, choose Normalize All Line Endings.

You can also choose what line ending style you want.

It is recommended that you stick to the style your current platform uses. If you're unsure what style your OS uses, you could pick Auto-Detect so that this plugin automatically handles it for you.

However, if you want a custom style, you can select from the 3 available line endings:

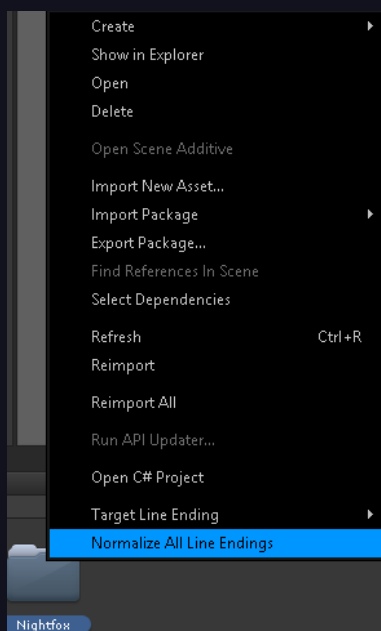
CR: Old Macs

LF: Linux and New Macs

CRLF: Windows

Afterwards, you may want to clear any Editor prefs this asset uses, which resides in the registry, .plist, or /etc folder depending on your current platform. To do this, simply select Clear Asset Editor Prefs.

Through Assets menu:



Simply right-click on your Project view, and locate Normalize All Line Endings at the bottom of the menu.

You can also click Target Line Ending and choose the desired line ending style you want, or if you're not sure, just select, Auto-Detect.

This is just the same as the Tools menu, but this has the advantage of being more accessible, which is just by right-clicking on Project view.

How does it work?

Consistent Line Endings checks for every supported script / text file in the project directory and detects the frequency of inconsistencies of line-endings present. It then verifies this data and normalizes all files with inconsistent line endings.

If the file doesn't match with the specified line-ending style you have set, then the file is converted as necessary to the target end-of-line style.

When kept on and set to operate automatically, Consistent Line Endings periodically checks for inconsistencies after every time scripts have been finished compiling.

When this setting is used, the plugin keeps hooked at the callback after scripts have been reloaded, thus fully utilizing the potential to eliminate any annoying warnings in Unity and in your External Script Editor (such as Visual Studio) every time you create a new script or make some changes that contains mixed ending styles.

Alternatively, once set to Off, you need to click Normalize All Line Endings to manually normalize them. This is useful if you just want to normalize line endings once.

Consistent Line Endings also handles your scripts / text files safely and properly. The normalization / conversion process will be terminated once any issues have been detected, to ensure your files never gets any undesirable changes.

Any possible issues with this asset?

Through a series of testing, there are barely any errors you can get from this asset. Either way, it's recommended that you create a backup of your project before using this plugin, just in case.

Possible issues:

ExecuteMenuItem target for Consistent Line Endings does not exist

You might get this error if you click toggle **Autofix Mode** too fast. Give it at least half a second after compiling, and everything should work as usual.

Shader errors and warnings

You might have this issue if you convert shader and compute files to CR (\r). Make sure if you're going to convert line endings of shaders, you are selecting either LF, or CRLF. To solve this issue, simply convert back to LF or CRLF.

...To view more about possible issues and solutions, please check the [Debugging Document](#).

That's all, and I'm glad you have read it up to here. I hope this asset helps you in your projects.

Getting inconsistent line ending warnings every time can really be annoying and can hinder one's workflow, no matter how insignificant it may seem at first.

After I have created this plugin, I have decided to share a Lite version (which works for most parts) to everyone who's having the same dilemma. If this had helped you, please don't hesitate to give this a review. 😊

On the other hand, if you have any suggestions, problems / bug reports, you can contact me at: <mailto:midnightfoxdev@gmail.com>