

# Disable Logging

## 1. Overview

### Disable Logging

- *disable logs in builds with 1-Click*

#### Is this a tool for you?

- Does it seem like a huge task to remove all calls to Debug.Log in release builds?
- Are you trying to use "Debug.isDebugBuild" or "Scripting Define Symbols" to check for debug?
- Do you want to output specific logs with color to stand out from the rest logs?

If you have met any of the above issues, Disable Logging is the tool for you.

#### Features:

- Enable/Disable logs in builds easily.
- Outputs the specific logs with the color you like.

## 2. Quick Start

### IMPORTANT:

*Please, always remove previous version before updating! It will let you avoid different compatibility issues and errors.*

Import the plugin, then read the following instructions.

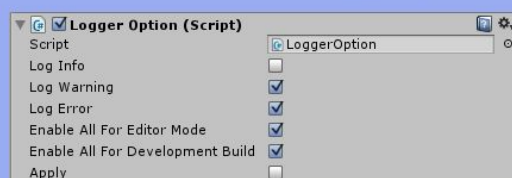
### 2.1 How to enable/disable logs in builds?

If you are working on a new project, do like this:

#### How to enable/disable logs in builds? (1/2)

If you are working on a new project,  
just do the following steps:

1. Call **Logger.LogXXX()** instead of Debug.LogXXX().
2. Don't forget to add the "**using DisableLogging;**" to the scripts which call the Logger.LogXXX().
3. Attach the "**LoggerOption**" script to a game object.
4. Set the options of "**LoggerOption**".

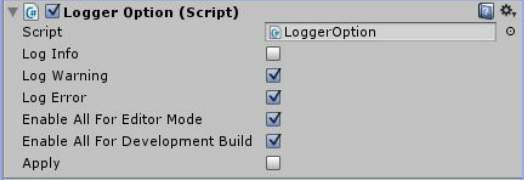


If you are working on an existing project, do like this:

**How to enable/disable logs in builds? (2/2)**

**If you are working on an existing project, just do the following steps:**

1. Replace "Debug.Log" with "**Logger.Log**".  
(In MonoDevelop, use the menu item: Search -> Replace in Files...)
2. Batch add "**using DisableLogging;**" to the scripts.  
(In Unity, use the menu item: Window -> Disable Logging -> Insert...)
3. Attach the "**LoggerOption**" script to a game object.
4. Set the options of "**LoggerOption**".



### LoggerOption (Script)

#### - Log Info

This option can enable/disable "Logger.Log()"

#### - Log Warning

This option can enable/disable "Logger.LogWarning()"

#### - Log Error

This option can enable/disable "Logger.LogError()" and "Logger.LogException()"

#### - Enable All For Editor Mode

If you choose this, all the logs (info, warning, error, exception) will be outputted in the unity editor mode. Ignore settings of "Log Info", "Log Warning" and "Log Error".

#### - Enable All For Development Build

If you choose this, all the logs (info, warning, error, exception) will be outputted in the development build. Ignore settings of "Log Info", "Log Warning" and "Log Error".

#### - Apply

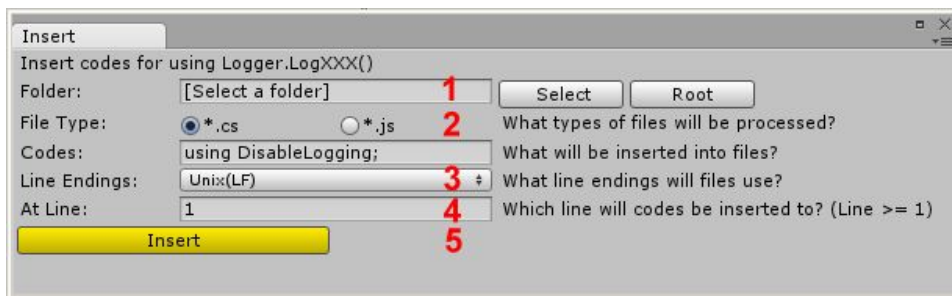
If you click this, all the settings will take effect immediately in the unity editor mode.

## 2.2 How to use the insert tool?

If you call Logger.LogXXX() in your script, you should add "using DisableLogging;" to that script.

We recommend you use the insert tool to batch add "using DisableLogging;" to your scripts.

(In Unity, Window -> Disable Logging -> Insert...)



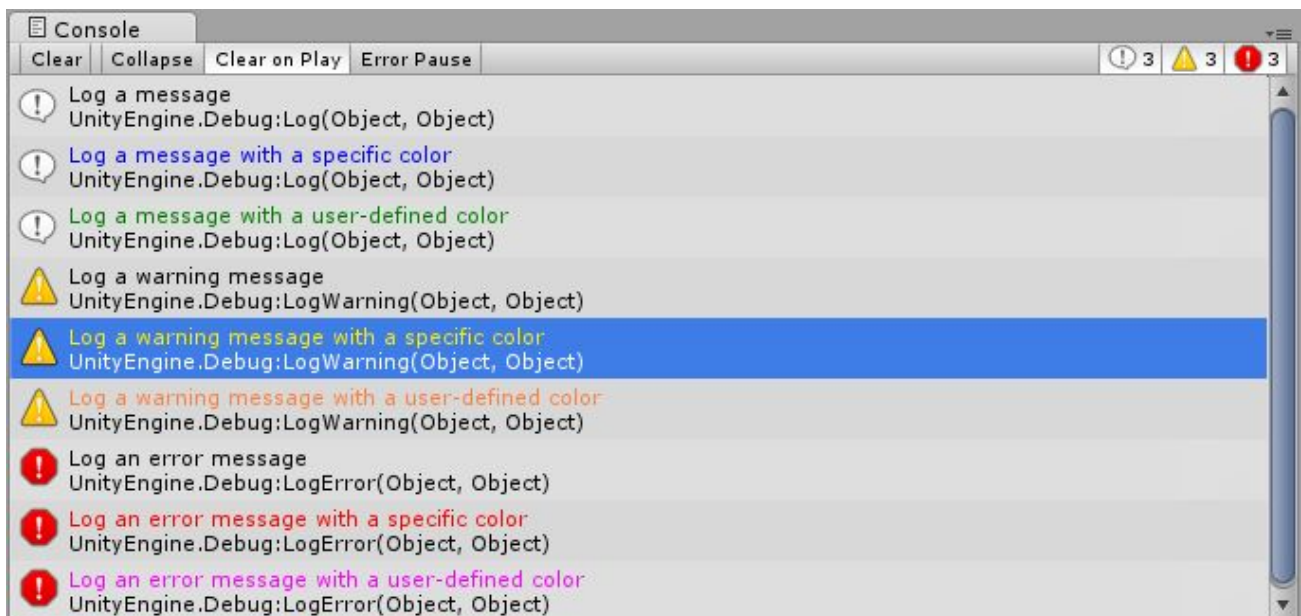
#### Steps:

- (1) Select a folder in the project view, and click the “Select” button to use the selected folder as the target folder. Or click the “Root” button to use the “Assets” folder as the target folder.
- (2) Select the type of files you want to process.
- (3) Select the line endings you want to use. We recommend you choose the same line endings of your files.
- (4) Set the line where codes will be inserted to. (Default is line 1, which means codes will be added to the top of the script.)
- (5) Click “Insert” button.

## 2.3 How to output colored logs?

This plugin provides many simple APIs, which can help you output the colored logs easily.

#### Example effects (in Unity Console):



#### Example codes:

// Examples of `Logger.Log()`

```
Logger.Log("Log a message");
```

```
Logger.Log("Log a message with a specific color ", Color.blue);
```

```
Logger.Log("Log a message with a user-defined color", new Color(0f, 0.5f, 0f));
```

// Examples of `Logger.LogWarning()`

```
Logger.LogWarning("Log a warning message");
```

```
Logger.LogWarning("Log a warning message with a specific color", Color.yellow);  
Logger.LogWarning("Log a warning message with a user-defined color", new Color(1f, 0.5f, 0.25f));
```

```
// Examples of Logger.LogError()
```

```
Logger.LogError("Log an error message");  
Logger.LogError("Log an error message with a specific color", Color.red);  
Logger.LogError("Log an error message with a user-defined color", new Color(1f, 0f, 1f));
```

```
// Examples of Logger.LogException()
```

```
Logger.LogException(new System.Exception("Log an exception message"));  
Logger.LogException(new System.Exception("Log an exception message with a specific color"),  
Color.red);  
Logger.LogException(new System.Exception("Log an exception message with a user-defined color"),  
new Color(1f, 0f, 1f));
```

### 3. Support

Thank you for using Disable Logging!

I hope you will find this plugin is helpful and it will save some of your priceless time!

Please, leave your reviews at the Asset Store page and feel free to drop me bug reports, feature suggestions and other thoughts on the forum or via email!

#### Disable Logging Links:

[Asset Store](#) | [Website](#) | [Docs](#) | [Unity Forum thread](#)

Email: [support@tiinoo.com](mailto:support@tiinoo.com)