

Fast Script Reload

Tool will allow you to iterate quicker on your code. You simply go into play mode, make a change to any file and it'll be compiled on the fly and hot-reloaded in your running play-mode session.

Getting started

1. Press play
2. Make code change
3. See results

Limitations

There are some limitation due to the method taken

No IL2CPP support

Asset runs based on specific .NET functionality, IL2CPP builds will not be supported. Although as this is development workflow aid you can build your APK with Mono backend (android) and change later.

Adding new fields

- sometimes when adding new fields you'll get some odd behaviour, this is generally happening when adding values before existing ones (in code) eg

```
[SerializeField] FunctionLibrary.FunctionName function; //existing

[SerializeField] private int _dynamicallyAddedField; //dynamically added

[SerializeField] private int _testIterationCounter = 1; //existing
```

This is likely down to the approach taken by asset in which it'll add [jmp] instruction to old class, seems like modifying structure (as in new methods / fields) can create issues.

What can help is moving new fields to be declared after all existing variables - ideally do not add new fields.

eg

```
[SerializeField] FunctionLibrary.FunctionName function; //existing

[SerializeField] private int _testIterationCounter = 1; //existing

[SerializeField] private int _dynamicallyAddedField; //dynamically added
```

- dynamically added fields will not run inline initializer, eg `private int _dynamicallyAddedField; = 1` will not initialize to 1, you have to do that in `OnScriptHotReload()` method
- dynamically added fields will only show in editor after full reload

Networked Version

- add info about broadcast and option to directly specify IP,
- add basic info about fw

Auto-save

- make sure to turn off auto save for files in editor (otherwise changes will be picked up)
- tool will also batch changes and execute new compile every 3 seconds (which can be configured in settings)

Performance

Performance should be on par with your standard code. The only hit comes at change time when compilation happens.