Overview

SmartNS v1.3.0

SmartNS is a simple editor extension that monitors the creation of C# scripts, and automatically adds a namespace to the scripts. The default behavior is for the namespace to mirror the physical path of the C# script within the unity project. For example, if you create a new script within a directory named ‘Assets/Code/Enemies`, SmartNS will add the declaration `namespace Assets.Code.Enemies {}` to the script, wrapping the content of the class.

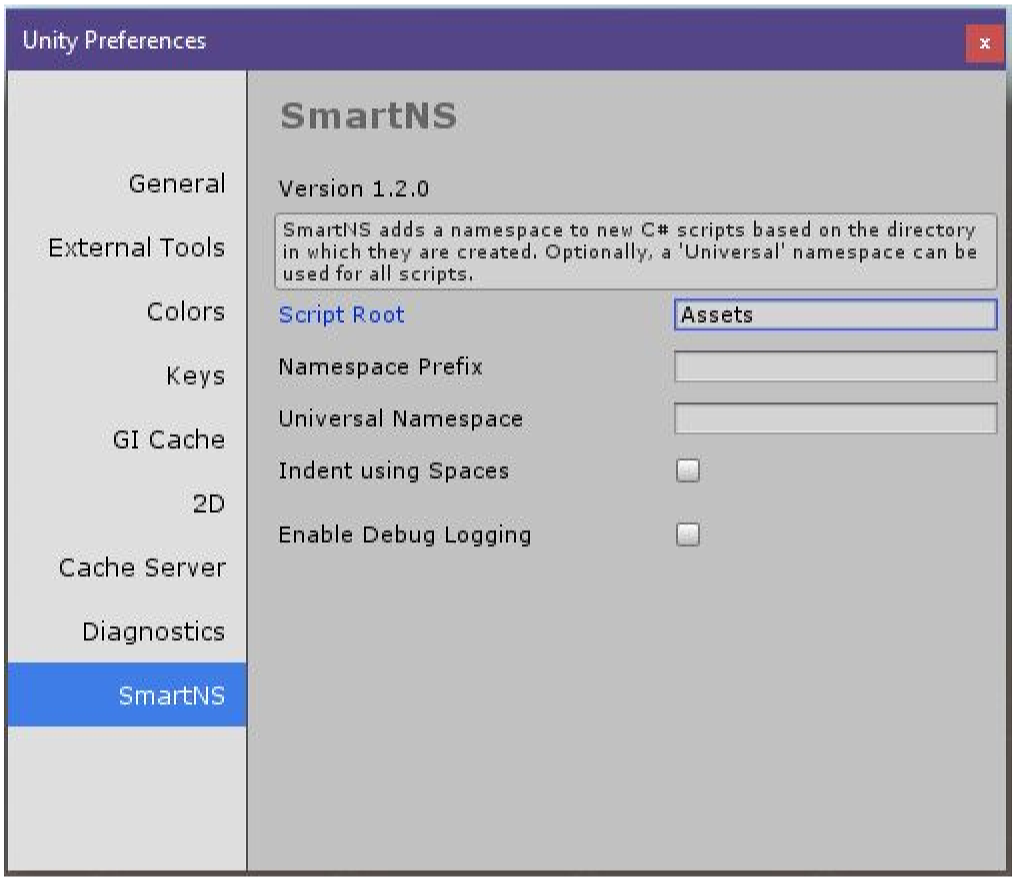
This behavior works whether you create a script via `Create -> C# Script`, or using other script creation methods, such as adding C# unit tests.

This package also adds a section to the the Unity Preferences menu to control optional behavior.

Setup

In Unity, simply import the package via `Assets -> Import Package -> Custom Package…`, or install the package from the Asset Store.

Options



SmartNS

Version 1.2.0

SmartNS adds a namespace to new C# scripts based on the directory in which they are created. Optionally, a 'Universal' namespace can be used for all scripts.

Script Root: Assets

Namespace Prefix

Universal Namespace

Indent using Spaces ☐

Enable Debug Logging ☐

After installing SmartNS, you can modify the behavior by adjust the preferences under `Edit -> Preferences…`. All preferences are project-specific, based on the “Company Name” and “Product Name” in the Player Settings. Changing either of these values will cause SmartNS preferences to reset, and you should restart Unity before adjusting the preferences again.

Preference options are:

- **Script Root**: Whatever you enter here will be stripped from the namespace. Since many projects keep all of their assets under a folder named Assets, it’s generally preferable for “Assets” not to be included in the namespaces. If you keep all scripts in a directory named “Assets/Code”, you might want to set this Script Root value to “Assets.Code” to remove that from the namespaces.
- **Namespace Prefix**: This is a value that will be prepended to all created namespaces, for example to put your company name into the namespace.
- **Universal Namespace**: This overrides the “smart” directory-based namespace generation, and instead uses the entered namespace in all cases. For example, if you set this to “MyUniversalNamespace”, then every script created will use that namespace.

- **Indent using Spaces**: The default behavior is to wrap the `` `public class…` `` declaration in the namespace declaration, and to indent all lines between the opening and closing braces using tabs. If this is checked, spaces will be used instead.
- **Enable Debug Logging**: This writes some log information to the console when scripts are created.