

Project

Create

- Folder
- C# Script
- Javascript
- Shader
- Compute Shader
- Prefab
- Audio Mixer
- Material
- Lens Flare
- Render Texture
- Lightmap Parameters
- Animator Controller
- Animation
- Animator Override Controller
- Avatar Mask
- Physic Material
- Physics2D Material
- GUI Skin
- Custom Font
- Shader Variant Collection
- Legacy

Inspector

CSharpCustomAsset

Open

Script: CSharpCustomAsset

Position: X 0 Y 0 Z 0

Name:

**Custom Asset**

Inspector

CustomAssetGenerator.config

Open

Auto Generate ☒

Change values and re-generate assembly

Logs ☒

Custom Assets

Assemblies Types

Assets/Create/Custom Assets/ Assembly-CSharp

Assets/Create/Custom Assets/ Editor

Assembly-CSharp-Editor

Assembly-UnityScript-Editor

Assets/Create/Custom Assets/ Firstpass

Assembly-Boo

Assets/Create/Custom Assets/ User Assemblies

TestLibrary

Re-generate

Custom Assets

- Generate Assembly
- Generate Config Asset
- Open Config Asset
- Create Custom Asset
- CSharpCustomAsset (Assembly-CSharp)
- CubeCustomAsset
- SphereCustomAsset
- Editor
- Firstpass
- User Assemblies

**Config Asset**

CSharpCustomAsset

CubeCustomAsset

SphereCustomAsset

CSharpCustomAsset

CSharpCustomAsset (Assembly-CSharp-Editor)

JSCustomAsset

JSCustomAsset



# CUSTOM ASSET GENERATOR

Unity gives you an option to store serializable data container classes into custom asset files. But you have to always write extra code for any of your classes that inherit [ScriptableObject](#) to generate custom asset of that class. This tool solves this problem.



Import [CustomAssetGenerator.dll](#) and go to "Assets/Create/Custom Assets/Generate Config Asset" main menu path to generate configuration asset.

Change values at configuration asset and re-generate *CustomAssets* assembly by re-generate button or "Assets/Create/Custom Assets/Generate Assembly" main menu path.



[CustomAssetGenerator.dll](#)

or



[CustomAssetGenerator.unitypackage](#)

Go to [Website](#).

Support [Website](#).



**Custom Asset Generator**

by Samet Kurumahmut

# VERSION HISTORY

Version 1.3.1

Sep 18, 2016

## Features

- Added “Create Custom Asset” menu item for selected *custom asset* scripts.
- Create custom assets through *configuration asset*.
- Added “Ping all Custom Assets” button to *configuration asset*.
- Added buttons to create custom assets, ping them, and ping *custom asset* scripts to the header of the *reorderable lists*.
- Added search bar.

Jump to the search bar: “Ctrl + F” hot-key on *Windows* / “Command + F” hot-key on *OS X*.

Close the search bar: “Esc” hot-key.

- Added version updater for next versions.

## Changes

- Sorting type lists by *assembly qualified name* when they have same *full name*.

## Fixes

- *Fixed:* Duplicate issues when script changes after *reorderable list* changes.
- *Fixed:* Undo issues after sorting alphabetically a list.
- *Fixed:* Not re-sorting while adding item to alphabetically sorted list.
- *Fixed:* Disabling auto generate when *Unity* re-instantiates the configuration object instance.

### Features

- Menu items can be created according to the types grouped from different assemblies.
- *Custom Asset Generator* tries to add the new assembly to “*Unity Assemblies*”, “*User Assemblies*”, and “*Newly Added*” list in order. On the other hand, it tries to add the new type to “*Unity Types*”, “*User Types*”, and “*Newly Added*” in order.
- Changing assemblies/types tool-bar tabs can be achieved by using the same keys with *Google Chrome* tab shortcuts.

*Jump to the next tab:* “**Ctrl + Tab**” or “**Ctrl + PgDn**” hot-key on *Windows* / “**Command + Option + Right Arrow**” hot-key on *OS X*.

*Jump to the previous tab:* “**Ctrl + Shift + Tab**” or “**Ctrl + PgUp**” hot-key on *Windows* / “**Command + Option + Left Arrow**” hot-key on *OS X*.

- Added ping buttons for all assemblies and types. *Show In Explorer/Reveal In Finder* with *Alt/Option* key.
- Added *type* and *className* overloads to *CreateAsset()* and *CreateAssetAtPath()* methods for public API *CustomAssetGenerator.Utils.ScriptableObjectUtil* type.
- Added *context* Object for log messages.

### Changes

- When any *custom asset* couldn't found, printing warning log messages instead of error.
- Added space between lists at *configuration asset*.

### Fixes

- *Fixed:* Processing nested *ScriptableObject* types.
- *Fixed:* Re-generating when already added script has changed.

- *Fixed:* “Cleaning up leaked objects in scene” warning on *configuration asset*.

## Version 1.2.0

Apr 21, 2016

### Features

- Added “*Open Config Asset*” menu item.
- Added an icon to differentiate active *configuration asset* from passive ones.
- Added contact links to the *configuration asset*.

### Changes

- *Custom asset* generation methods in *ScriptableObjectUtil* returns newly generated *custom asset* instead of void.
- Importing generated *CustomAssets* assembly using *AssetDatabase.ImportAsset()* instead of *AssetDatabase.Refresh()*.

### Fixes

- Will not delete previous *CustomAssets* if an error occurs while generating a new assembly.
- *Fixed:* *Unauthorized access exception* issue while removing previous *CustomAssets* assembly.

## Version 1.1.0

Sep 21, 2016

### Features

- *Custom Asset Generator* location is truly independent anymore. Previously, *Custom Asset Generator* files must be located at “CustomAssetGenerator/Editor/”. But you're now free to put it anywhere. Nevertheless, locating under “CustomAssetGenerator/Editor/” path is a good way to differentiate *Custom Asset Generator* files from your files.

### Changes

- Generate button text now can present “Generate” or “Re-generate” by looking *CustomAssets* assembly location that's in same directory with *CustomAssetGenerator* assembly. Previously, it was only fixed “Re-generate”.

### Fixes

- Configuration asset would create empty a *ReorderableList* when user had no *Unity* or *User* assembly.
- *ScriptableObjectUtil.CreateAssetAtPath()* method would create custom assets at fixed path. Now, it creates them at given path.

## Version 1.0.0

Sep 7, 2015

### Features

- Generating custom assets of any of your classes that inherit *ScriptableObject* without extra coding.
- Creating generic menu by re-orderable list
- Sort alphabetically
- Auto generate
- Enabling/Disabling logs