
Auto
Singleton

Contents

Namespace

All Auto Singleton classes are in the **AutoSingleton** namespace.

Add this at the top of your scripts to access them:

```
using AutoSingleton;
```

Creating a Singleton

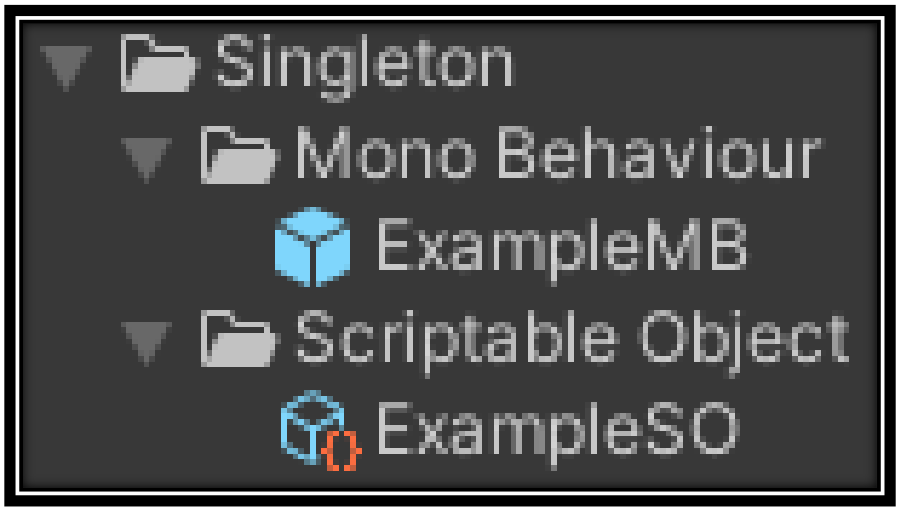
Basic

To create a singleton, simply add the **Singleton** attribute over a class derived from **MonoBehaviour** or **ScriptableObject**:

```
[Singleton]
public class ExampleMB : MonoBehaviour
```

```
[Singleton]
public class ExampleSO : ScriptableObject
```

An instance of it will be created automatically in the project **Assets** folder.



You can freely rename, move, and modify it. If you delete the singleton component from a prefab asset, a new one will be created.

Advanced

The **Singleton** attribute have different fields that you can set:

```
[Singleton(
    inherited = true,
    displayName = "Example Display Name",
    folderPath = "Folder/SubFolder"
)]
```

inherited
If true, this attribute will also be applied to any derived classes.

displayName
Choose the name that the singleton instance will have in **Assets** (not inherited).

folderPath
Choose where the singleton instance will be created, relative to the **Assets/** folder (inherited).

Retrieving a Singleton

Single Instance

To access a singleton from scripts, all you have to do is use the **Singleton<T>** class **Instance** property:

```
ExampleMB singletonInstance = Singleton<ExampleMB>.Instance;
```

Multiple Instances

You can also get multiple singletons derived from a parent class using **Instances**:

```
IReadOnlyList<ParentMB> allMBs = Singleton<ParentMB>.Instances;
```

You can even do it using an interface that your singletons implement:

```
var allImplementer = Singleton<IInterface>.Instances;
```

Selecting an Instance

Using the **Instance** property when multiple singleton instances correspond to the chosen template parameter of **Singleton<T>** will throw an **Exception**.

You can use the **SelectInstance** method to choose which singleton to get when calling **Instance**.

This lets you have one piece of code define what singleton to use, and all your other scripts adapt by using the chosen instance.

```
// In one script...
Singleton<ParentMB>.SelectInstance((pmb) =>
{
    return pmb.enabled;
});

// And in another...
Singleton<ParentMB>.Instance.DoSomething();
```

Find Method

Both the generic **Singleton<T>** and non-generic **Singleton** classes have a **Find** method.

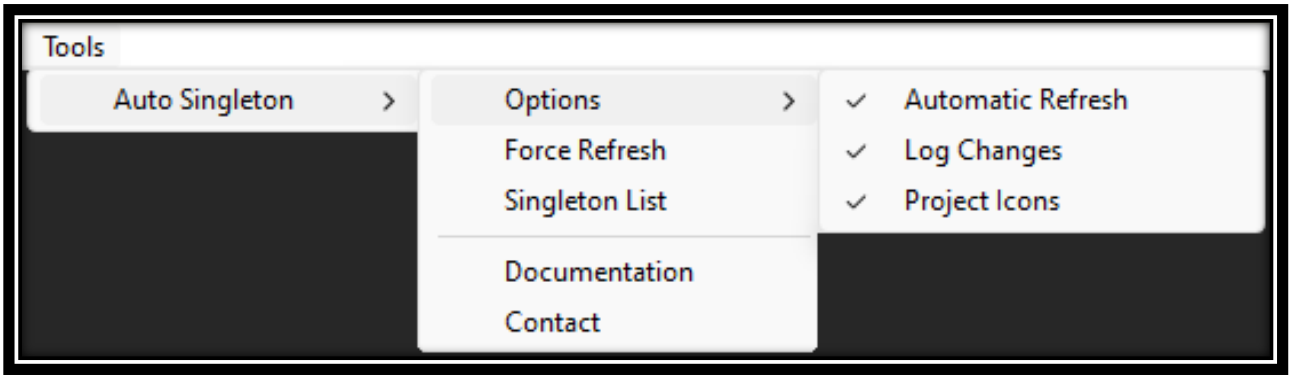
Using it, you can get every singleton that matches the given predicate (a function that returns either true or false).

```
Object[] match = Singleton.Find((singleton) =>
{
    return singleton.name.StartsWith("a");
});
```

This method will instantiate a new array on every call, you should cache the return value when possible.

Top bar menu

In **Tools/Auto Singleton** you will find different menu item:



Options/Automatic Refresh
If enabled, the singletons will be created and deleted automatically after every successful compilation.

Options/Log Changes
If enabled, we will log in the console every singleton changes.

Options/Project Icons
If enabled, we will display an icon on singleton instances in the **Project** (Assets) window.

Force Refresh
Trigger the creation and deletion of singletons according to the **Singleton** attribute usage.

Singleton List
Open in the inspector the list of all singletons, handy to easily find an instance in the **Assets** folder. It lets you disable any singleton, preventing them from being instantiated at runtime.

Documentation
Open this pdf.

Contact
Open a web page giving you ways to contact the tool creator.

Demo

The demo is a simple Tic Tac Toe game made following an MVC pattern.

It showcases 4 singletons, check them out to see how to use the tool in different scenarios.

It is recommended to delete the demo folder once you've explored it, to avoid instantiating useless singletons and to remove unused assets.

Final words

Thank you for getting this tool!

If you have any questions, discover a bug, or have a feature idea, please don't hesitate to contact me at justetools@gmail.com.

If you enjoy **Auto Singleton**, please consider leaving a review. Your feedback helps improve the tool!