

# Bear Data Editor

List, Search, and edit objects with an easy-to-access alternative to the Project view. Working with Bear Data Editor enables frictionless editing of your content from a single place as your project grows.

Any Class inheriting from ScriptableObject and MonoBehaviour can use the [BearDataEditor] attribute (available in the CollisionBear.BearDataEditor namespace) to be exposed by the editor window. The editor can display any scriptable object, or prefab with a MonoBehaviour component, of a given type. The editor finds them in the project folder regardless of location and sorts the results by name. Support for several open editor instances simultaneously is also available, along with filtering among the objects based on their names.

Next to every asset is a folder icon that reveals its location in the Unity project folder.

## Getting started

First, you need to get your hands on a copy of the editor. We support a few options.

### Unity Asset store

[Preview Field](#)

### Unity Package

The editor extension can be added to Unity's package manager from 'Add package from git URL'

<https://github.com/CollisionBear/BearDataEditor.git>

## Manual download

You need to put the Bear Data Editor content inside your Unity project's Asset folder.

<https://github.com/CollisionBear/BearDataEditor/releases/download/1.0.0/BearDataEditor-1.0.0.unitypackage>

## Example

Decorate a class with the attribute. By default, the class's name will be displayed.

```
using CollisionBear.BearDataEditor;
[BearDataEditor]
class TestClass: ScriptableObject {}
or
[CollisionBear.BearDataEditor.BearDataEditor]
class TestClass: ScriptableObject {}
```

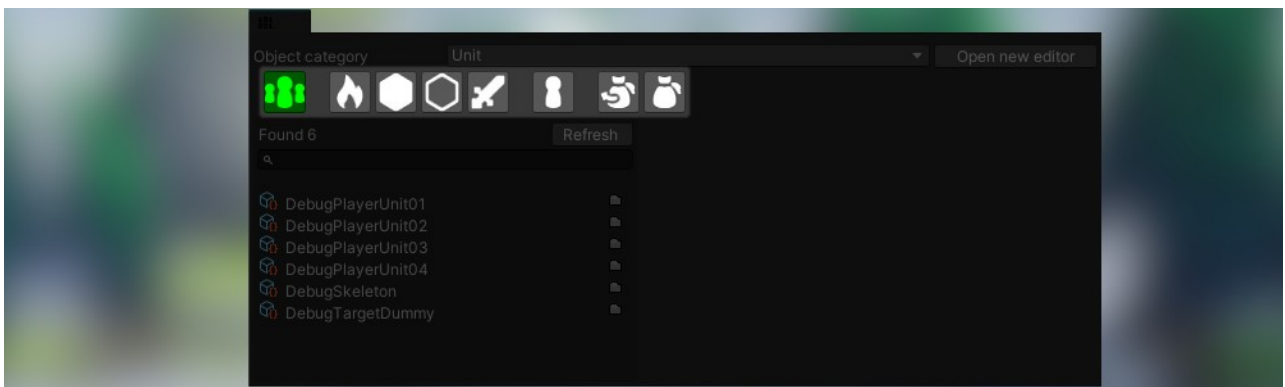
If you want to display another name for your class in the editor, enter a DisplayName for it in the attribute.

```
using CollisionBear.BearDataEditor;
[BearDataEditor(DisplayName = "Some other name")]
class TestClass: ScriptableObject {}
```

## Icon bar & Hotkeys

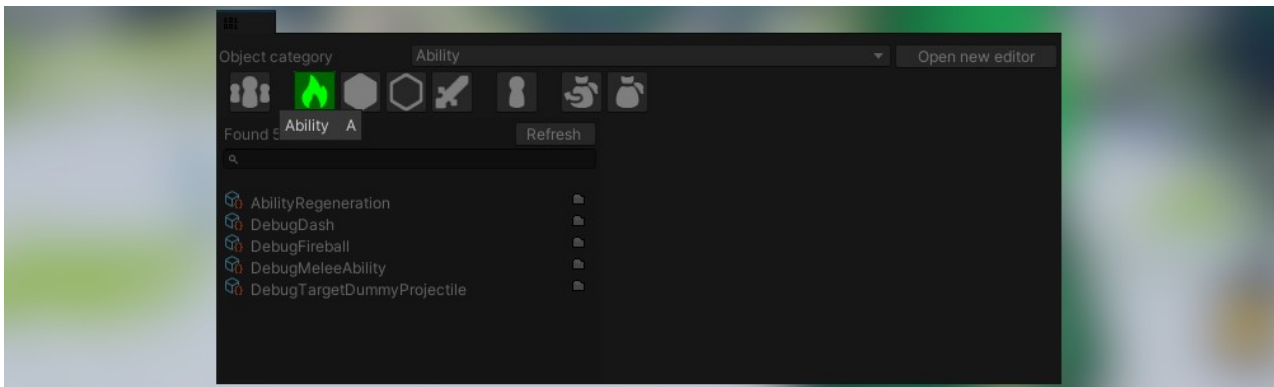
The BearDataEditor also supports icons. Set the UseIcon field to true, and the IconPath to some resource (Uses [Unity's resource folder](#));

```
using CollisionBear.BearDataEditor;
[BearDataEditor(UseIcon = true, IconPath = "pawn")]
class Unit: ScriptableObject {}
```



Set the HotKey to enable a hot key for this type. (Note! Only works when the editor window is focused)

```
using CollisionBear.BearDataEditor;
[BearDataEditor(UseIcon = true, IconPath = "fire", HotKey = KeyCode.A)]
class Ability: ScriptableObject {}
```



## License

This project is released as Open Source under a [MIT license]  
(<https://opensource.org/licenses/MIT>).