EasyExcel

Locke

How To Use

1. Editor a xlsx

Editor a xlsx with the format below:

	A	B	C	D	E	F	G
1	ID	Name	HP	Attack	Speed	Items	Words
2	int	string	int	int	float	int[]	string[]
3	0		0	0	0	0	
4	1001	Goblin	20	3	2.8	1,2,3,4,5	a,b,c,d3
5	1002	Orc	80	4	2	1	dddd
6	1003	Shaman	100	5	1	1,2,3,4	a,b
7	1004	Ogre	1000	50	5.5		
8	1005	Bandit	120	25	2	1,2,3	
9	2001	Brigand	160	10	4.2		
10	2002	Marauder	200	18	3.6		
11	2003	Goblin	20	3	2.8	1	
12	2004	Orc	80	4	2		

The first row is the **name** of variables.
The second row is the **type** of variables.
The third row is **default values**.
For more details, you can refer to the xlsx files in EasyExcel/Example/ExcelFiles.
The table below show the formats of types and default values.

Type	Default Value
int	0 or any other int value
int[]	0 or any other int value
float	0 or any other float value
float[]	0 or any other float value
string	empty of any string without outside quotes
string[]	empty of any string without outside quotes

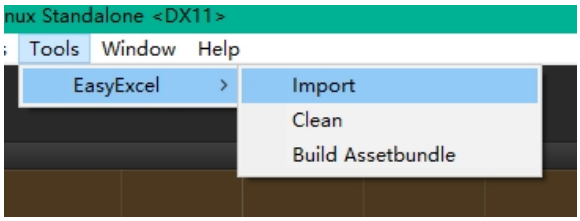
2. Set paths for generating

Set your following two paths in EasyExcel/Config.cs

CSharpPath	It is where generated code will be. It is relative to your Unity project root, for example “Assets/EasyExcel/Example/AutoGenCode/”.
AssetPath	It is where generated assets will be. It is relative to your Unity project root, for example “Assets/EasyExcel/Example/AutoGenAsset/”.

3. Import xlsx files

Click menu Tools->EasyExcel->Import



You will see a dialog for selecting the folder where your xlsx files are. Browser to your folder and click ok, then the xlsx files will be imported to your project.
When the process is done, you will see .cs files in CSharpPath and .asset files in AssetPath.

4. Set resource mode

(1) **LoadFromAssetbundle**

Before running, you need to Set LoadFromAssetbundle. Set true if you want to load the generated assets from assetbundle. Set false if you want to load them from Resources folder.

If true, AssetPath must be in one of your project's Resources folders,
and the option AssetbudleName below will be applied as the name of the assetbundle,
the option AssetbundlePath will be the output folder.

If false, AssetPath should be outside of all the Resources folders, and the two option AssetbudleName and AssetbundlePath will not be used.

(2) **AssetBundleName**

The name of the assetbundle. You can change it to your own, or leave it by default.

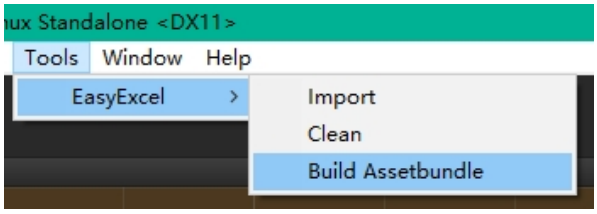
(3) **AssetbundlePath**

The output folder of the assetbundle. You can change it to your own, or leave it by default.

(4) Build Assetbundle

If LoadFromAssetbundle is false, ignore this step.

If LoadFromAssetbundle is true, you need to build assetbundle first:



5.Open the example scene and play.

If false, AssetPath should not be in any of the Resources folders, the two option AssetbudleName and AssetbundlePath will not be used.

Config.cs

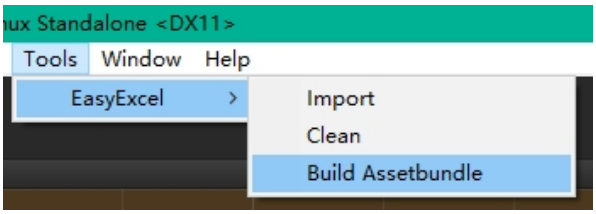
The table below shows the details of Config.cs

Name	Description
CSharpPath	This is where the generated csharp files will be.
AssetPath	This is where the generated ScriptableObject files will be.
LoadFromAssetbundle	If true, the ScriptableObjects should be built into assetbundle before running. The AssetbudleName below will be applied as the name of the assetbundle. The AssetbundlePath below will be the output folder. If false, make sure AssetPath is somewhere in Resources folder, for example "Assets/Resources/aaa/bbb", and the two options AssetbudleName and AssetbundlePath will not be used.
AssetbudleName	Assetbudle name of generated ScriptableObject files.
AssetbundlePath	Assetbundle path

RowDataClassNamePostfix	Postfix of generated RowData classes. For example Item.xlsx corresponds to Item + RowDataClassNamePostfix.
DataTableClassNamePostfix	Postfix of generated RowData classes. For example Item.xlsx corresponds to Item + DataTableClassNamePostfix.
AssetFileExtension	Extension of generated ScriptableObject files. For example Item.xlsx corresponds to Item + AssetFileExtension.
NAME_ROW_INDEX	This row in a excel sheet is Name. Used for importing xlsx.
TYPE_ROW_INDEX	This row in a excel sheet is Type. Used for importing xlsx.
DATA_START_INDEX	This row in a excel sheet is where real data starts. Used for importing xlsx.

Tools Menu

Tools menu of EasyExcel are under Tools->EasyExcel.



1.Import:

Select a folder of xlsx files and import them as ScriptableObjects.

2.Clean

Delete all the files and folders of Config.CSharpPath and Config.AssetPath.

3.Build Assetbundle

Build the assetbundle of all the generated ScriptableObjects, to folder Config.AssetbudlePath with assetbundle name Config.AssetbudleName.

API Examples

1.Initialize and load all data

This should be put where your game initialize.

```
DataTableManager manager = new DataTableManager();
manager.Load();
```

2.Find a ItemRowData by id

```
// Get ItemRowData with id 1001
var itm = dataTableManager.Get<ItemRowData>(1001);
Debug.Log(itm.Description);
```

3.Find ItemRowData list

```
// Get RoleRowData list
Dictionary<int, RowData> dic = dataTableManager.GetList<RoleRowData>();
foreach (var item in dic.Values)
{
    RoleRowData np = item as RoleRowData;
    Debug.Log(np.Icon);
}
```