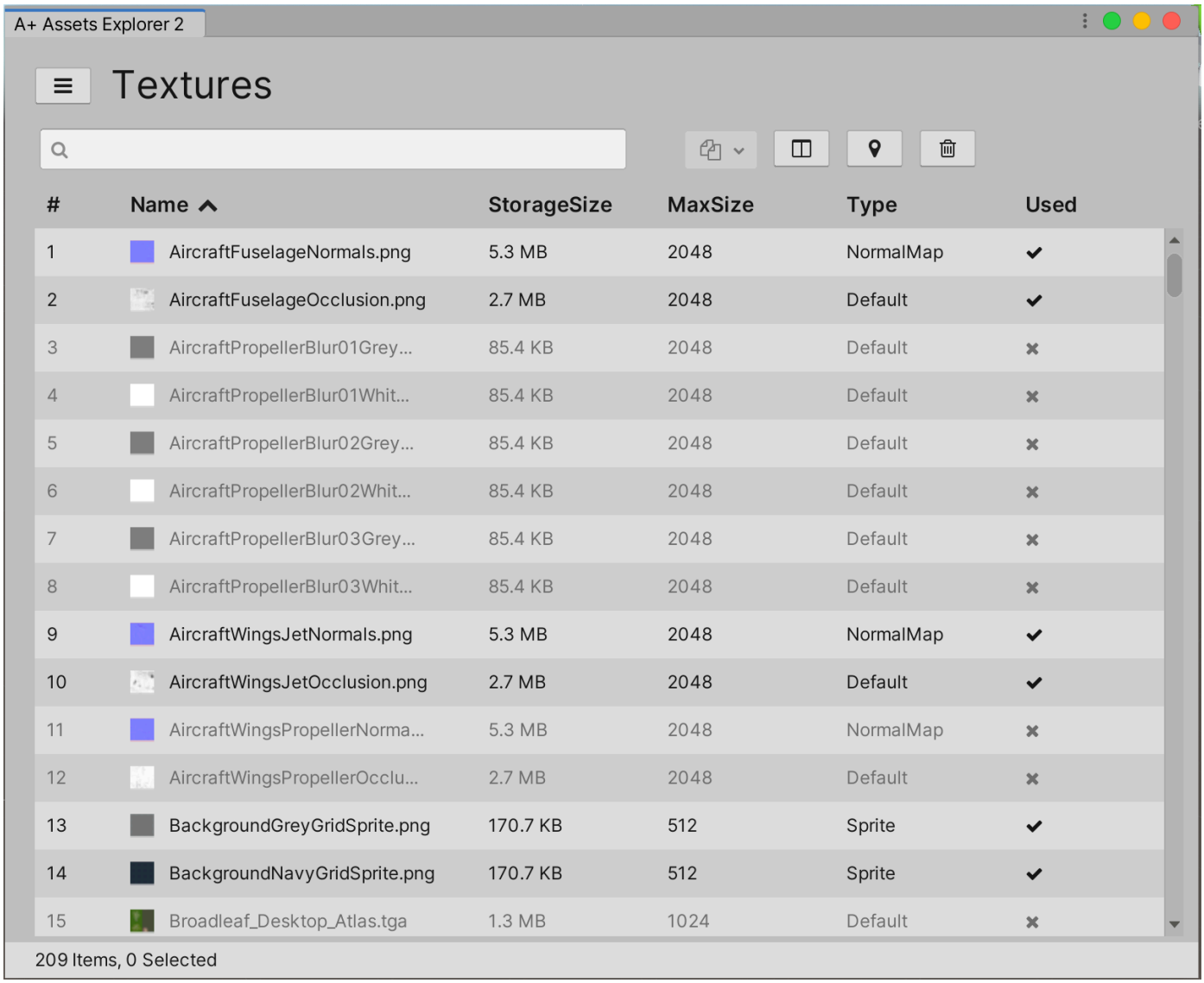
A+ Assets Explorer 2

Open Assets Explorer

Select the **Assets Explorer** menu item from the **Tools** -> **A+ Assets Explorer 2** menu in Unity Editor

Assets Table UI



Search Assets

Search input. We can search assets via this control by typing condictions string into input and then press `ENTER` on the keyboard. It deault to seach asset name if there no seach field specific.

Search Condition Format

Searching in Assets Explorer supports multiple asset properties. For each property, the search condition format is

> [AssetPropertyDataHeader] : [>|<] [SearchValue]

For the above format:

- `[AssetPropertyDataHeader]` is the table header in Assets Explorer
- `:` is Required, it connects data header and its value

- `[>|<]` is not needed for `string` type property. For `number` type property, if it's not specified, it will default to `>`
- `[SearchValue]` is the property value we want to search. For `string` type, if you have serveral keywords to search, use `|` to combine them

NOTE: if there are no valid search format, the input string will trust as the search vaule of **Name** property of assets.

Samples:

- Below are samples for searching for textures:

> 1. `Name:icon` means searching the textures whose name **contains** 'icon' chars
> 2. `StorageSize:>1024` means searching the textures whose storage size is larger than 1024 KB
> 3. `MaxSize:<2048` means searching the textures whose import parameter MaxSize is less than 2048

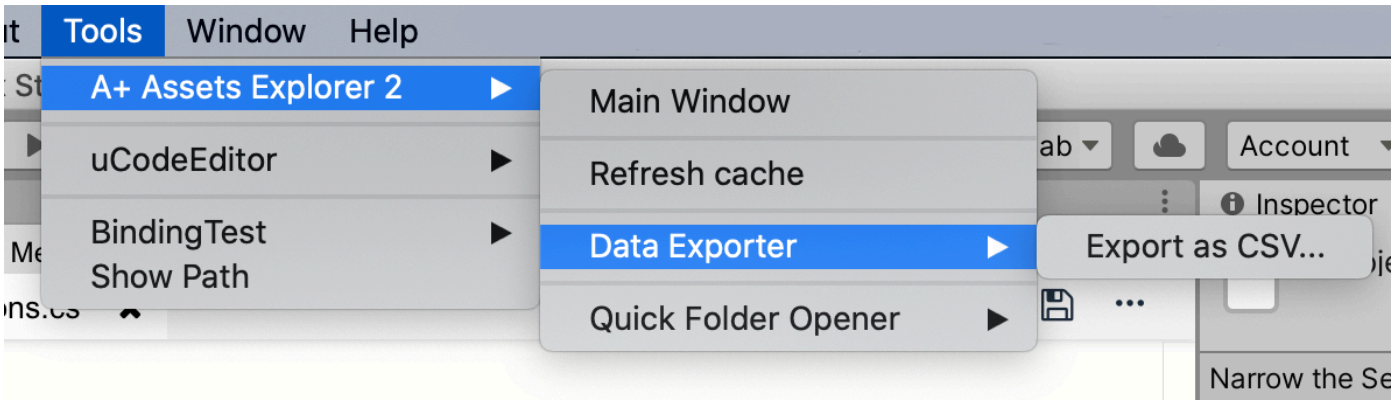
- For multiple conditions, each search condition will always be **`AND`** logic. Take textures search for example:

> `StorageSize:>1000 KB MaxSize:>1024`
>
> *means:*
>
> search the texture with storage size is larger than 1000KB **AND** import parameter MaxSize is larger than 1024

- The multipe values sample:

> `Name: icon | body` means search the texture whose name **contains** 'icon' **OR** 'body'

Copy Actions Dropdown



Copy the name or path of selcted assets in table.

Change table Headers

Open the table header settings winwdow.

Locate Assets In Project window

Click this button, A+ Assets Explorer will select the selected assets in Unity Project Window

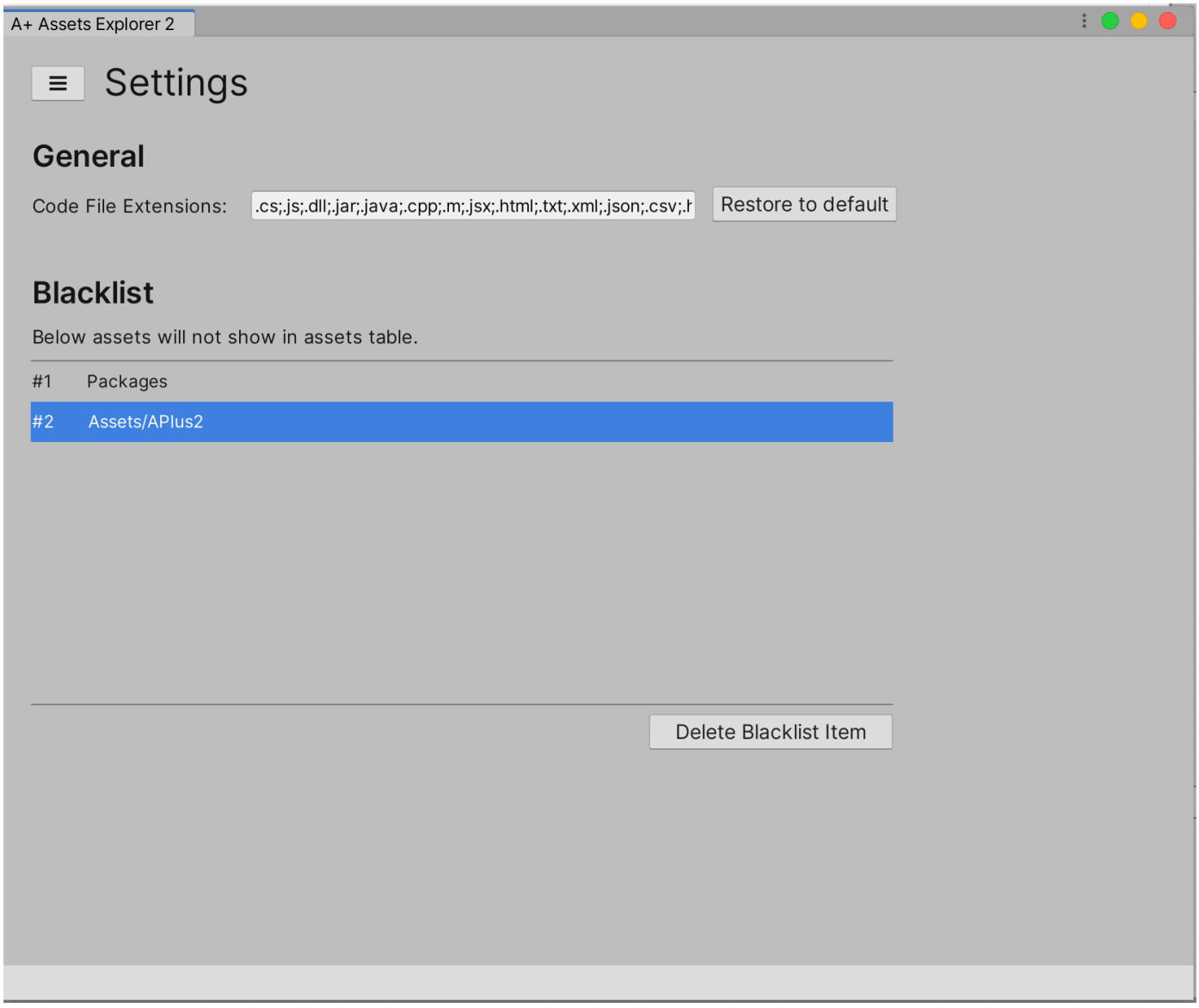
Delete Assets

Delete selected assets by this button. Delete assets action cannot undo.

Themes

Supports Unity Person or Professional theme.

Settings



General

Code File Extensions

The extension of files that assets explorer consider it whether its code file or not.

Blacklist

Both of assets and folders can be added into blacklist:

- If assets in blacklist, it will hide from assets data table view.
- If folder in blacklist, all of assets under the folder hide from assets data table view.

Add Assets / folders into Blacklist

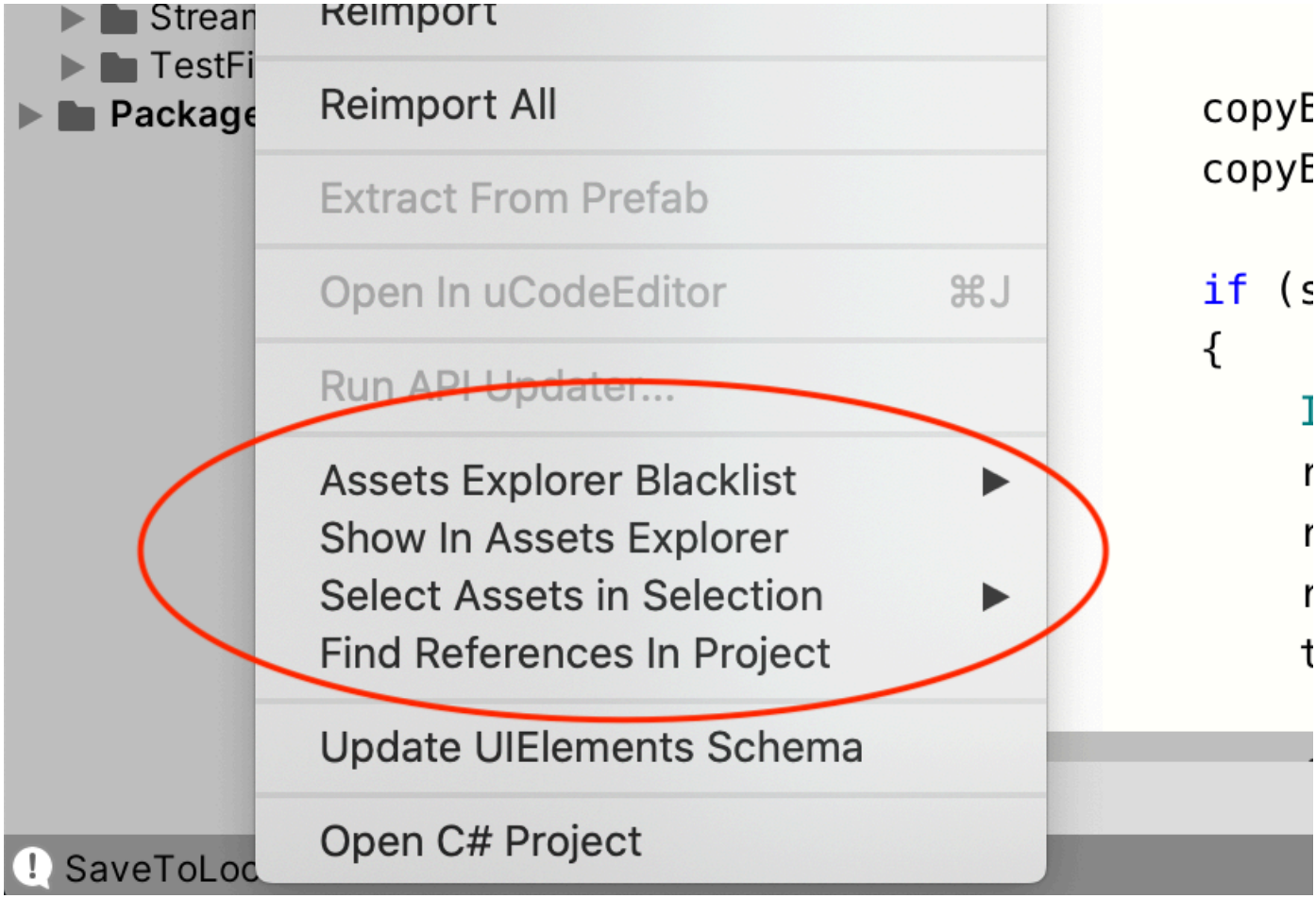
Select the assets/folder in Unity Project Window, and then right click to choose **Assets Explorer Blacklist** -> **Add**

Remove Assets / folders from Blacklist

Two ways to remove from blacklist:

- Select the assets/folder in Unity Project Window, and then right click to choose **Assets Explorer Blacklist** -> **Remove**
- In Assets Explorer UI, click the `Delete Blacklist Item` button on the right corner of UI

Context Menu



Assets Explorer Blacklist

Add selected assets/folders to or remove selected assets/folders from blacklist cache.

Show In Assets Explorer

Show the selected assets in Assets Explorer. The menu item is disabled when one of below two conditions matched:

1. There is no Assets Explorer Window
2. Multiple type assets are selected. For example, two assets selected, one is a texture and the other is a model

Select Assets In Selection

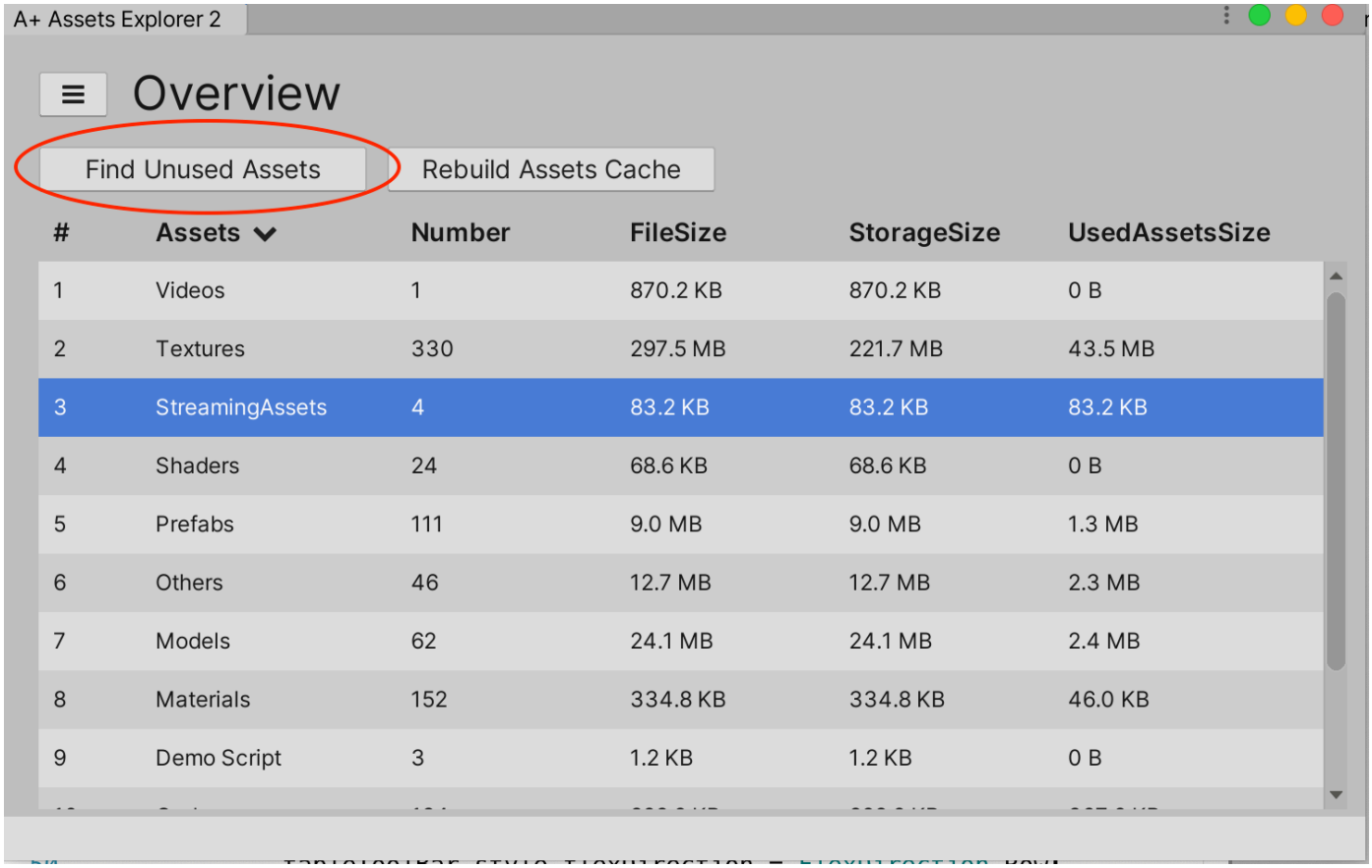
Select the assets by type in current seleced objects in Project Window.

Find References In Project

Find the references assets of the selected assets, similar to the **Find Preferences In Scene**.

Find Unused Assets

Click `Find Unused Assets` button on `Overview` tab to trigger a build to find unused assets.



What assets are Unused?

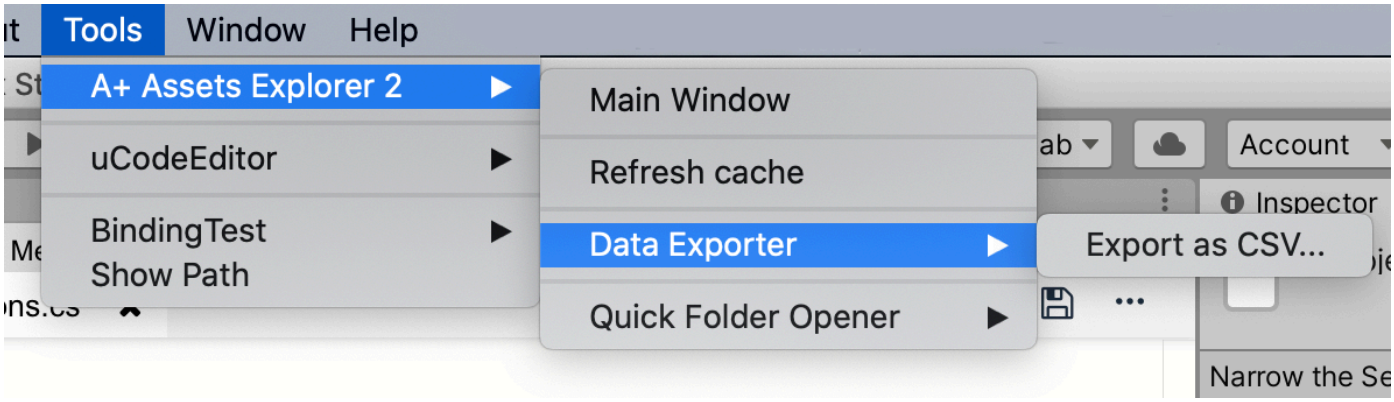
When the assets satisfy one of below condictions are **Used**, others are **Unused**:

- In **Resources** folder
- In **StreamingAssets** folder
- Has **AssetBundle** assigned
- The Assets are the dependencies of an AssetBundle
- In **Unity Build Report**. Because of this one, we have to trigger a build to get the unused file

How Assets Explorer show the Used/Unused assets?

- When A+ Assets Explorer does not know whehter assets are used or not, `Used` column displays ? mark
- If asset is used, it will display black color nd `Used` column display ✓ mark
- if asset is unused, it will display grey color and `Used` column display × mark

Data Exporter



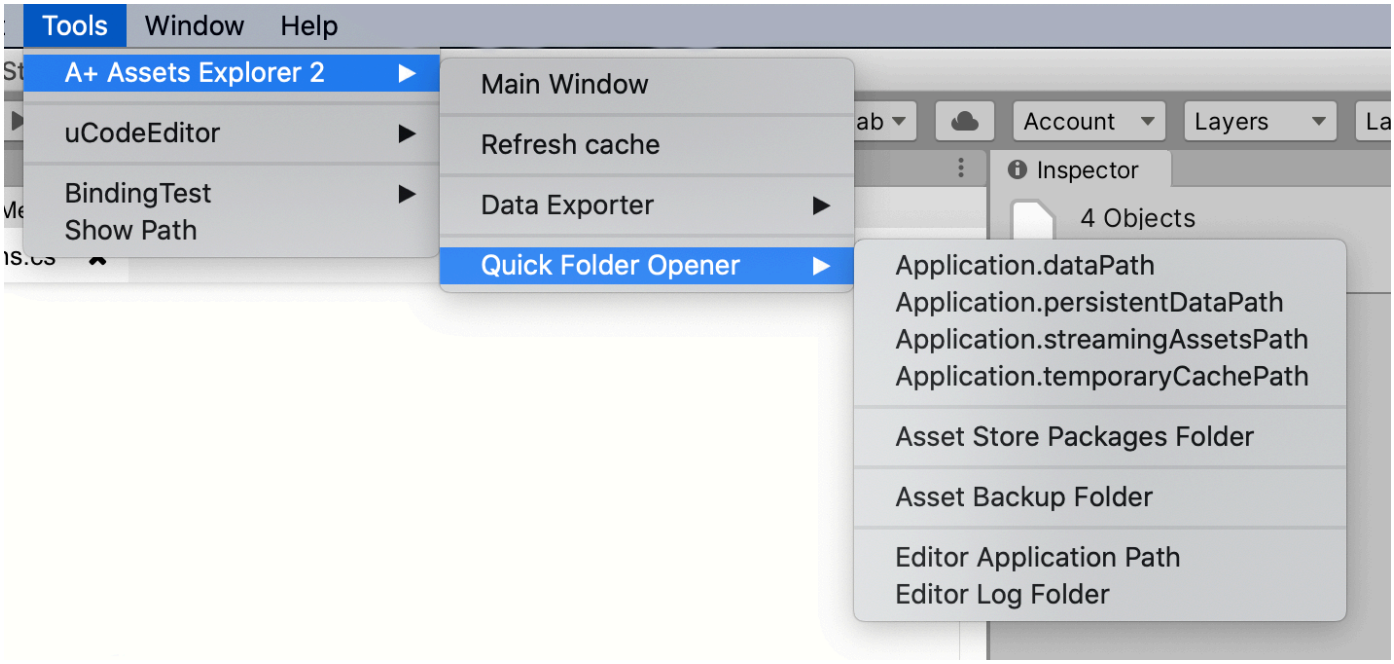
Export assets data to a local file. Select menu **Tools** -> **A+ Assets Explorer 2** -> **Data Exporter**, currently only CSV is supported.

Export to CSV

Select **Tools** -> **A+ Assets Explorer 2** -> **Data Exporter** -> **Export as CSV...** to save all data to local csv files.

Quick Folder Opener

Select Menu **Tools** -> **A+ Assets Explorer 2** -> **Quick Folder Opener** and select folder you want to open.



Supported Folders:

The Quick Folder Opener can following path in both of Windows and macOS.

- Application.dataPath
- Application.persistentDataPath
- Application.streamingAssetsPath
- Application.temporaryCachePath
- Asset Store Packages Folder
- Asset Backup Folder (the folder that Assets Explorer back files to when deleting assets)
- Editor Application Path

- Editor Log Folder

Addressables support

By default, Find unused assets for addressables support is turn off. If we want to turn it on,

1. We need to install **Addressables package** from Package Manager first,
2. And then install the **AddressablesIntegration.unitypackage**.
3. Refresh cache

Now, addressables will trust as used assets.

Table Headers

Below headers are supported by A+ Assets Explorer 2 currently. If you need more, welcome to send email to me.

Textures

Headers	data type	Comments
Name	string	File name of texture file
FileSize	number	The file size of texture file
StorageSize	number	Storage size of texture file
MaxSize	number	Max size of texture
TextureFormat	string	Texture format
ReadWrite	boolean	ReadWrite enable or not
MipMap	boolean	MipMap enabled or not
Type	string	Texture type
WidthInPixel	number	Width of texture in pixel
HeighInPixel	number	Height of texture in pixel
Width	number	Width of texture
Height	number	Height of texture
File Path	string	The file path of the texture file
Unused	boolean	Used in the game or not
Id	string	The file guid of assets or file guid with local id of nested assets
Hash	String	The hash of the asset file

Models

Headers	data type	Comments
Name	string	File name of texture file
FileSize	number	The file size of model file
VertexCount	number	Vertex count of model
Tris	number	Tris count of model
ScaleFactor	number	Scale factor of model
MeshCompress	boolean	Compress mesh or not
OptimizeMesh	boolean	Optimize mesh
ReadWrite	boolean	Is readable or not
ImportBlendShapes	boolean	Should Unity import BlendShapes
GenerateColliders	boolean	Should Unity generate mesh colliders for all meshes
LightmapToUV2	boolean	Generate lightmap UVs to UV2
SwapUVs	boolean	Swaps the 2 UV channels in meshes. Use if your diffuse texture use UVs from lightmap
Path	string	The file path of the model file
Used	boolean	Used in the game or not
Id	string	The file guid of assets or file guid with local id of nested assets
Hash	String	The hash of the asset file

Animations

Headers	data type	Comments
Name	string	Name of animation
In File	string	In which file
Length	number	Frame Length
FPS	number	Frame per second
LoopTime	boolean	Loop animation or not
LoopPose	boolean	Loop pose or not
CycleOffset	number	Cycle Offset
Path	string	The file path of animation file
Used	boolean	Used in the game or not
Id	string	The file guid of assets or file guid with local id of nested assets
Hash	String	The hash of the asset file

Audios

Headers	data type	Comments
Name	string	File name of audio file
ImportedSize	number	The size after imported
FileSize	number	The file size of audio file
Ratio	number	Compress ratio
Frequency	number	Audio frequency
Compress Format	string	Compress format of audio
Duration	number	Duration of the audio
Quality	number	Quality of the audio
Background	boolean	Run in background
path	string	The file path of the audio file
Used	boolean	Used in the game or not
Id	string	The file guid of assets or file guid with local id of nested assets
Hash	String	The hash of the asset file

Movies

Headers	data type	Comments
Name	string	File name of movie file
Approx	number	Approx size of the movie
Texture Size	number	Texture size of the movie
Quality	number	Quality of the movie
Duration	number	The duration of the movie
Path	string	The file path of movies file
Used	boolean	Used in the game or not
Id	string	The file guid of assets or file guid with local id of nested assets
Hash	String	The hash of the asset file

Materials

Headers	data type	Comments
Name	string	File name of material file
FileSize	number	File size of material file
Type	string	The type of material: Material or Physics Material
Shader	string	The shader name that material used
Path	string	The file path of material file
Used	boolean	Used in the game or not
Id	string	The file guid of assets or file guid with local id of nested assets
Hash	String	The hash of the asset file

Shaders

Headers	data type	Comments
Name	string	Shader name
FileName	string	Name of shader file
FileSize	number	Size of shader file
RenderQueueText	string	Render Queue in text
RenderQueue	number	Render Queue in number
LOD	number	Level of Detail
VariantsIncluded	number	Shader variants only in scene
VariantsTotal	number	All shader variants
SurfaceShader	boolean	has surface shader or not
CastShadows	boolean	Cast shadows or not
IgnoreProjector	boolean	Ignore projector or not
DisableBatching	string	Batch is disabled or not
Path	string	The file path of shader file
Used	boolean	Used in the game or not
Id	string	The file guid of assets or file guid with local id of nested assets
Hash	String	The hash of the asset file

StreamingAssets

Headers	data type	Comments
Name	string	File name
FileSize	number	Size of file
Path	string	The file path of file
Used	boolean	Used in the game or not
Id	string	The file guid of assets or file guid with local id of nested assets
Hash	String	The hash of the asset file

Code files

Headers	data type	Comments
Name	string	File name
FileSize	number	Size of file
FileType	string	File type of asset file
Path	string	The file path of file
Used	boolean	Used in the game or not
Id	string	The file guid of assets or file guid with local id of nested assets
Hash	String	The hash of the asset file

Others

Headers	data type	Comments
Name	string	File name
FileSize	number	Size of file
FileType	string	File type of asset file
Path	string	The file path of file
Used	boolean	Used in the game or not
Id	string	The file guid of assets or file guid with local id of nested assets
Hash	String	The hash of the asset file

Extend Assets Explorer

Add new asset category

Assets Explorer will detect classes in Editor assembly (c# files under Editor folder in Projector), if anyone is inherited from `AssetCategory` class, a new asset category will be added into Asset Explorer UI.

Below is the sample code for add new category base on scriptable object assets.

```
using System.Collections.Generic;
using UnityEngine;
using APlus2;
using System;
```

```
using UnityEditor;

/// <summary>
/// Change your scriptable object inherit from APScriptableObject class
/// </summary>
[CreateAssetMenu]
public class DemoScript : APScriptableObject
{
    public int Strong;
}

/// <summary>
/// Create a asset class inherit from APAsset form caching
/// </summary>
public class DemoScriptAsset : APAsset
{
    public int Strong;
}

/// <summary>
/// Change your scriptable object editor script to inherit from APAssetImporter class
/// </summary>
[CanEditMultipleObjects]
[CustomEditor(typeof(DemoScript))]
public class DemoScriptEditor : APAssetImporter<DemoScript, DemoScriptAsset>
{
    /// <summary>
    /// If needs update asset in cache or not?
    /// </summary>
    protected override bool ShouldUpdate(DemoScriptAsset cachItem, DemoScript target)
    {
        return cachItem.Strong != target.Strong;
    }

    /// <summary>
    /// Copy data to asset in cache
    /// </summary>
    protected override DemoScriptAsset TargetToAsset(DemoScript target, DemoScriptAsset
currentAsset)
    {
        currentAsset.Strong = target.Strong;
        return currentAsset;
    }
}

/// <summary>
/// If class inherit from AssetCategory class, a new asset category will be added into
Asset Explorer UI
/// </summary>
```

```
public class DemoScriptAssetCategory : AssetCategory
{
    public override void AddAsset(string assetPath)
    {
        AddAsset<DemoScriptAsset>(assetPath, GetDemoScriptAssetFromGuid);
    }

    public override string CreateAssetType()
    {
        return "demoscript";
    }

    public override Dictionary<string, ColumnAction> CreateColumnActions()
    {
        // Key have to be lower case
        return new Dictionary<string, ColumnAction>()
        {
            { "demoscriptasset_strong", new LabelColumnAction<DemoScriptAsset>(asset =>
asset.Strong.ToString(), asset => asset.Strong) },
        };
    }

    public override List<Column> CreateColumns()
    {
        // Key format is [ClassName]_[FieldName]
        return new List<Column>()
        {
            Column.CreateInstance("APAsset_Name", "Name", 240),
            Column.CreateInstance("APAsset_FileSize", "FileSize", false),
            Column.CreateInstance("DemoScriptAsset_Strong", "strong", 140),
            Column.CreateInstance("APAsset_Path", "Path", 360, false),
            Column.CreateInstance("APAsset_Used", "Used"),
            Column.CreateInstance("APAsset_Id", "Id", 280, false),
            Column.CreateInstance("APAsset_Hash", "Hash", 280, false),
            Column.CreateInstance("APAsset_InAssetBundle", "InAssetBundle", 140, false)
        };
    }

    public override NaviMenuItem CreateMenu()
    {
        return new NaviMenuItem(Icons.Others, this.CreateAssetType(), "Demo Script");
    }

    public override APOverviewItem CreateOverviewItem(AppState state)
    {
        var assets = state.GetAssetCacheItem(this.CreateAssetType()).assets;
        return CreateItem(
            assets.Count,
            GetTotalFileSize(assets),
```

```
            GetUsedStorageSize<DemoScriptAsset>(assets, tex => tex.FileSize),
            GetStorageSize<DemoScriptAsset>(assets, tex => tex.FileSize)
        );
    }

    public override List<APAsset> GetAssets()
    {
        return AssetsHelper.GetAssetsListByType<APAsset>("DemoScript",
GetDemoScriptAssetFromGuid);
    }

    private static DemoScriptAsset GetDemoScriptAssetFromGuid(string id)
    {
        var path = AssetDatabase.GUIDToAssetPath(id);
        DemoScript script = AssetDatabase.LoadAssetAtPath<DemoScript>(path);

        DemoScriptAsset asset = new DemoScriptAsset();
        asset.Id = id;
        asset.Path = path;
        asset.Hash = Utilities.GetFileMd5(path);
        asset.Name = Utilities.GetFileName(path);
        asset.FileSize = Utilities.GetFileSize(path);
        asset.Strong = script.Strong;
        return asset;
    }

    public override bool IsMatch(UnityEngine.Object obj)
    {
        return obj is DemoScript;
    }

    public override Type RegisterAPAssetClass()
    {
        return typeof(DemoScriptAsset);
    }

    public override void UpdateAsset(string assetPath)
    {
        UpdateAsset<DemoScriptAsset>(assetPath, GetDemoScriptAssetFromGuid);
    }
}
```

More Info

Visit https://www.amlovey.com