DOC VERSION 2.4.1 - JULY 2023

Contents

Introduction

Thank you for purchasing the **Rainbow Folders** extension! We hope you enjoy using the product and that it makes your game development project a success.

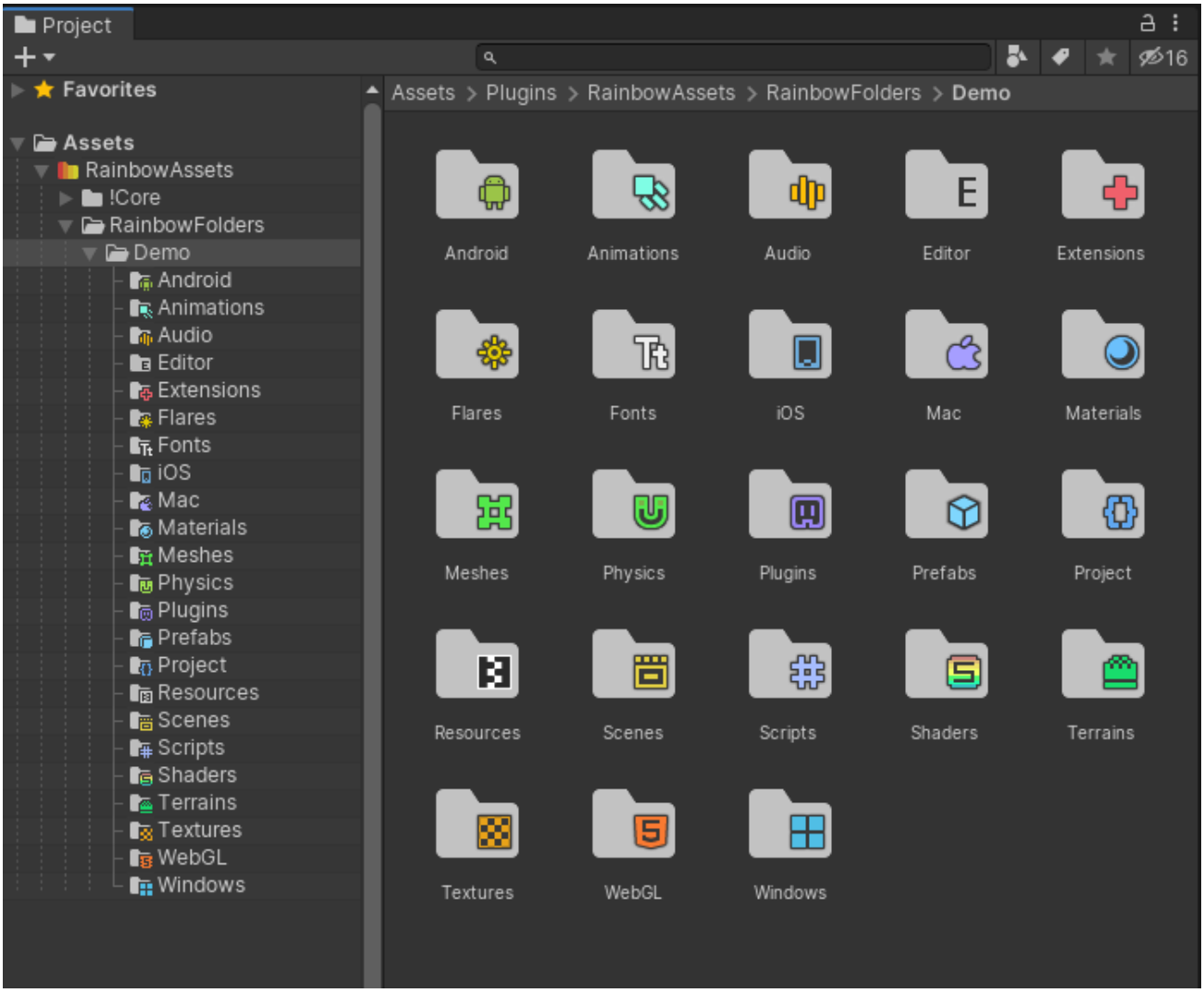
If you have questions, please do not hesitate to contact us at *support@borodar.com*, we will be glad to help you out.

Also, when you have a spare moment, please leave us a review on the Asset Store.

Overview

Have you ever thought about highlighting often used project folders? This simple but colorful asset allows you to do that!

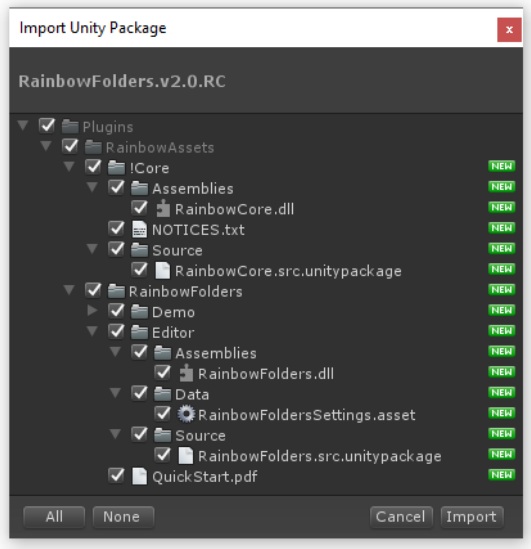
With **Rainbow Folders**, you can set a custom icon and background for any folder in the Unity project browser:



Installing

Rainbow Folders is a standard Unity extension and should be installed like any other Unity package. Just drag the `RainbowFolders.unitypackage` into your current project, or in the Editor go to the drop-down menu **Assets** → **Import Package** → **Custom Package** and then browse to the `RainbowFolders.unitypackage` file.

When downloading from the Asset Store, then the Download Manager will automate this process.



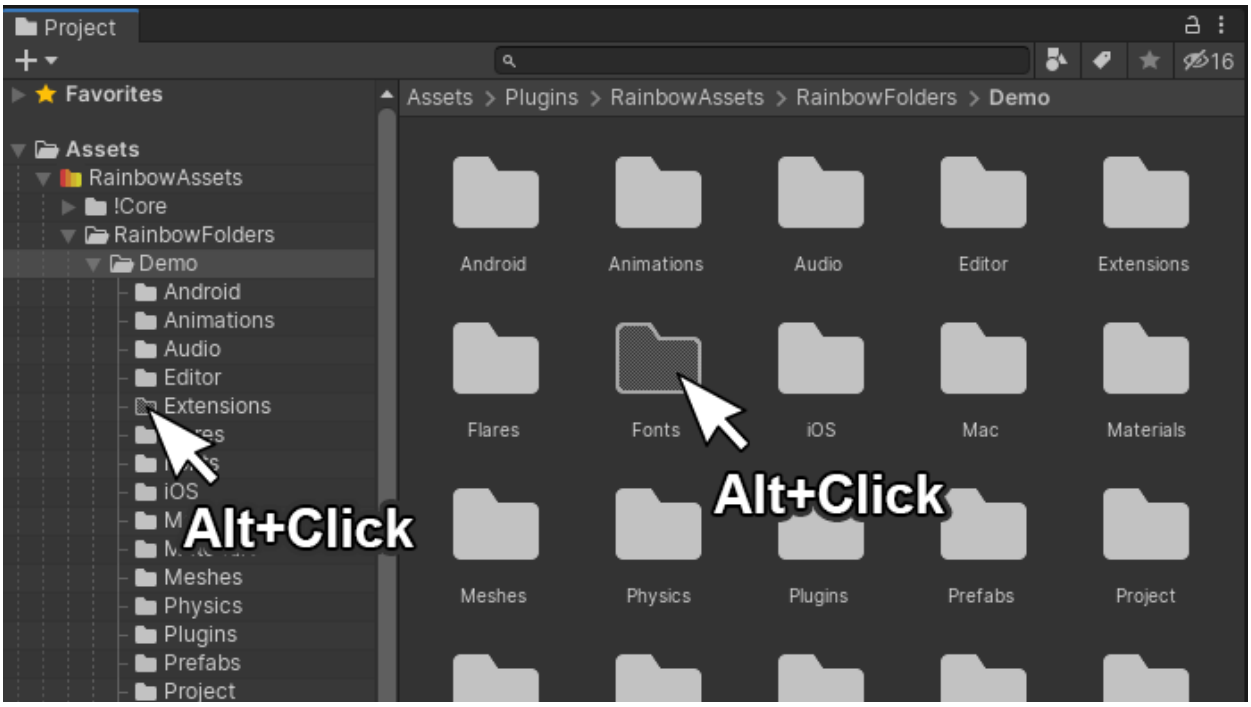
Once the Importing dialog appears, just click the Import button.

Asset Folder Location

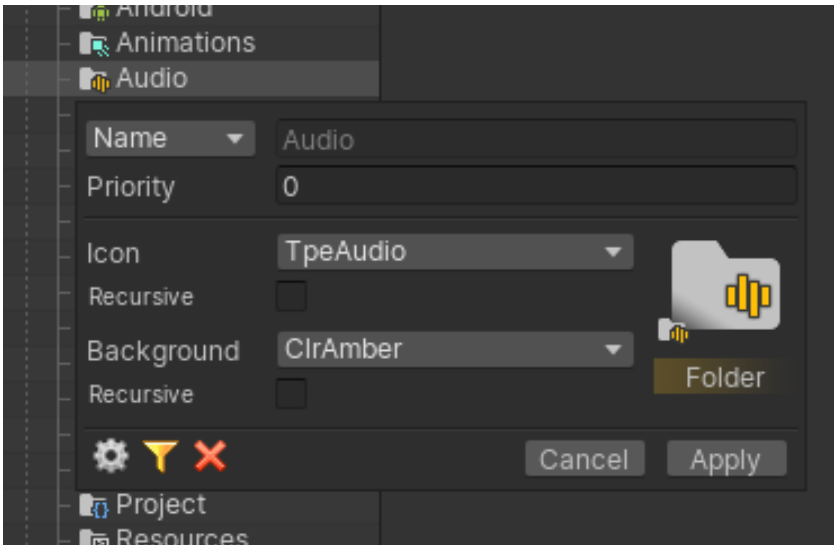
The package will be imported into the `Assets/Plugins/Borodar/RainbowFolders` folder by default. Most users prefer to keep it here, but you can freely move it wherever you want. **Rainbow Folders** will detect new location automatically, no additional actions are needed.

Configuring folder icons

To apply a custom icon and/or background for some folder in your project view, just hold the **Modifier key** and click on any folder icon in the Unity project browser. By default, it's the **Alt key,** but you can change it in **Project Settings** → **Borodar** → **Rainbow Folders**.



Configuration dialogue will appear, and you'll be able to assign icon and background to the corresponding folder.



What you need to configure for each custom folder rule, are these fields:

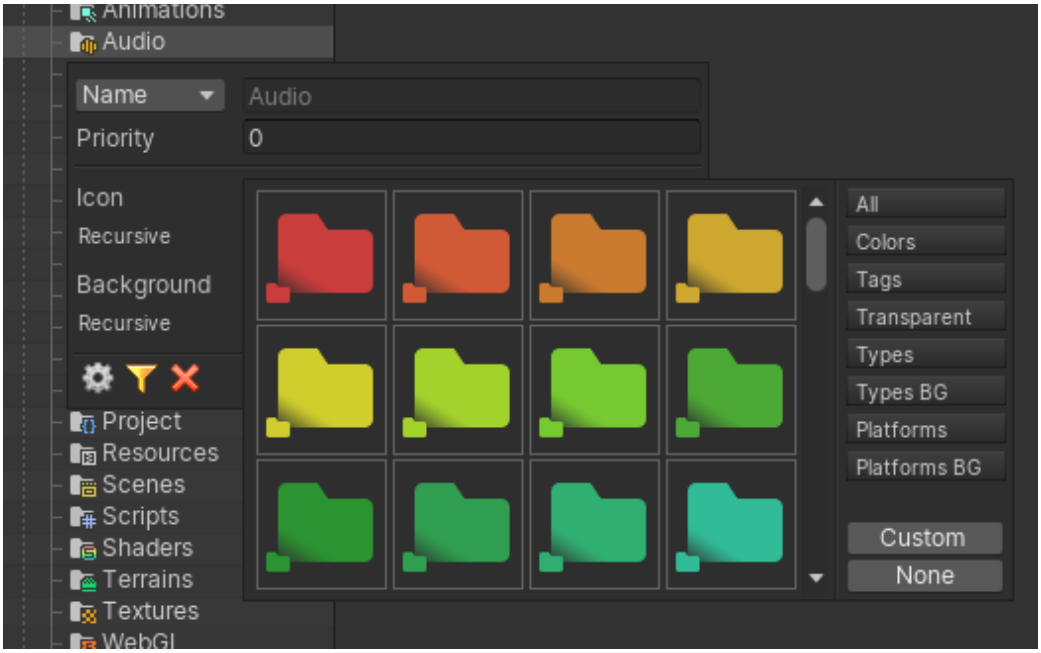
- **Folder Name** - this rule will be applied to all folders with that name.
  or **Folder Path -** this rule will be applied to a single folder with the specified path.
- **Priority** - if there is more than one rule that should be applied for the same folder, then the rule with the highest priority will be applied.

- **Icon** - a custom icon that should be applied to the corresponding folder. You can choose it from presets or apply your own texture.
- **Background** - a custom background that should be applied to the corresponding folder. You can choose it from presets or apply your own texture.
- **Recursive** checkboxes - the same icon or background will be applied automatically to all subfolders.

Your changes will be applied next time when the project browser will retrieve focus.

Presets

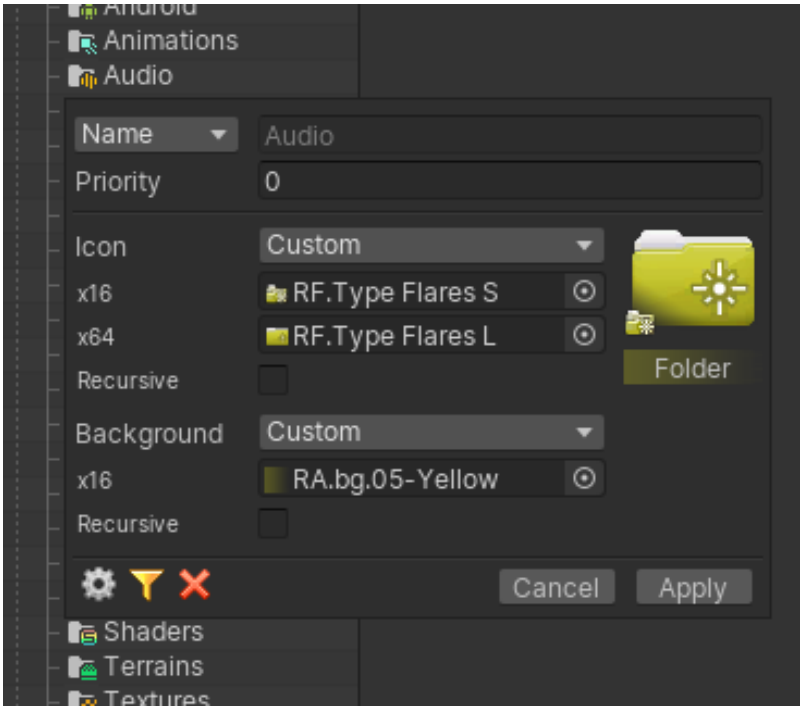
You can choose icons and backgrounds from a few dozen of presets. Simply click on the corresponding field in the configuration dialog, select one of them from the popup, and apply changes.



Your own icons

If you want to apply your own texture as a folder icon or background then click on the corresponding field in the configuration dialog. The icon selection popup will appear and it will have the “Custom” button you should click on.

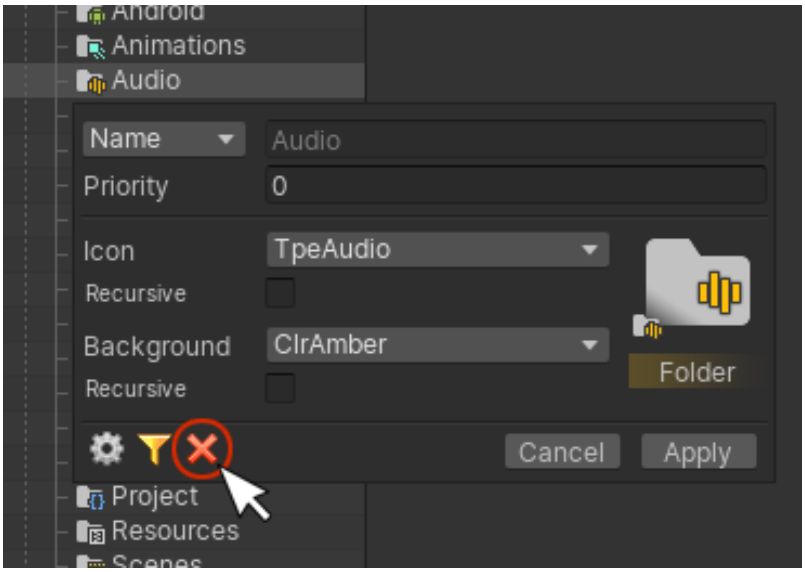
In this case selection popup will be closed and additional texture input fields will be shown in the configuration dialog to allow you to drag your own textures here:



Recommended sizes are **16x16 px** and **64x64 px** for small and large folder icons correspondingly. Regarding folder background, its height should be **16 px** and the width could be arbitrary.

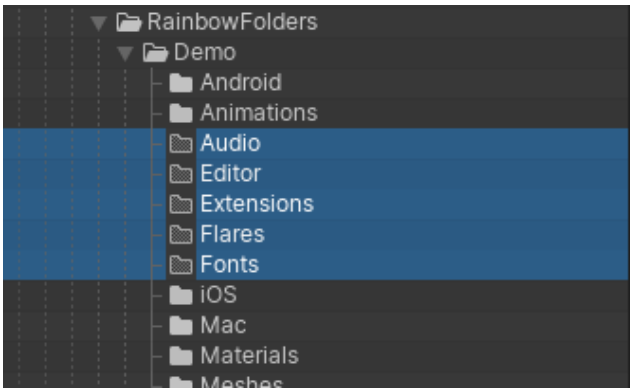
Revert to default

To reset the folder icon and background to the default ones, just **Alt-click** on it, then press the red cross button in the configuration dialogue and apply changes.



Multi-editing

You can also edit multiple folders at once, just select them all and **Alt-click** at one of their icons.

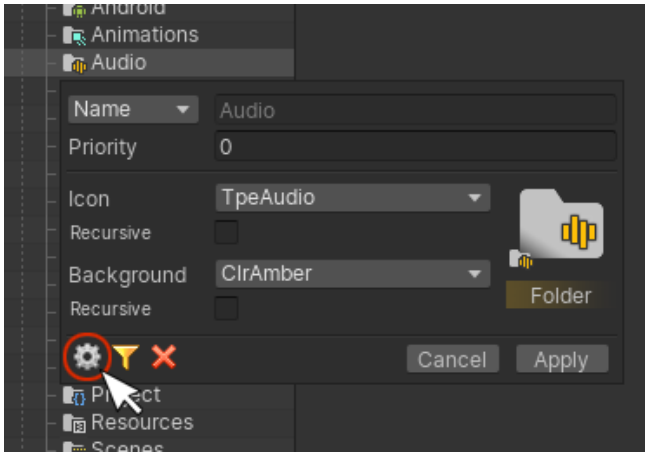


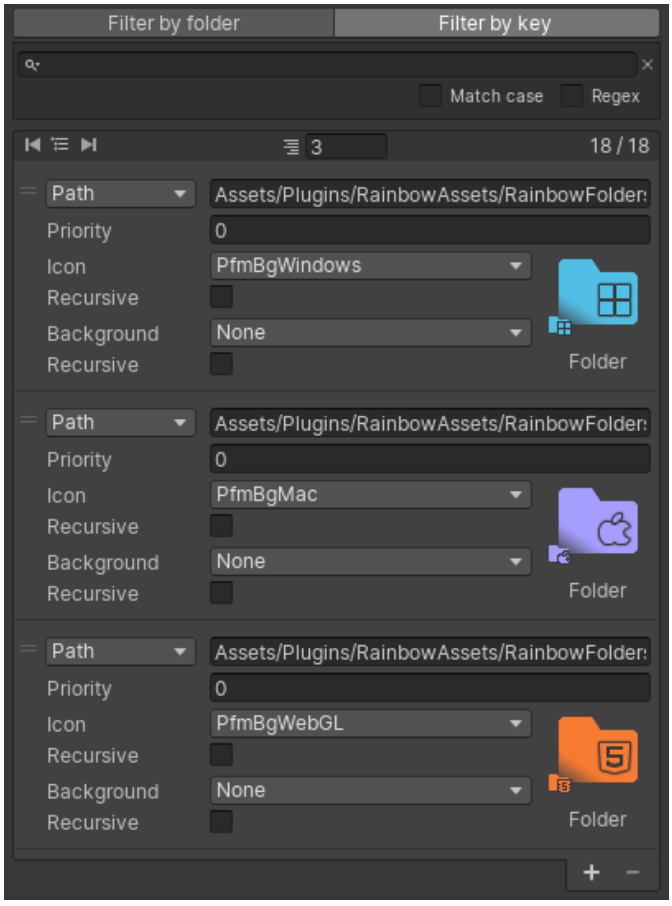
Manage rules directly

To view all existing rules, click on the **“gear”** button in the configuration dialog, then take a look at the Inspector.

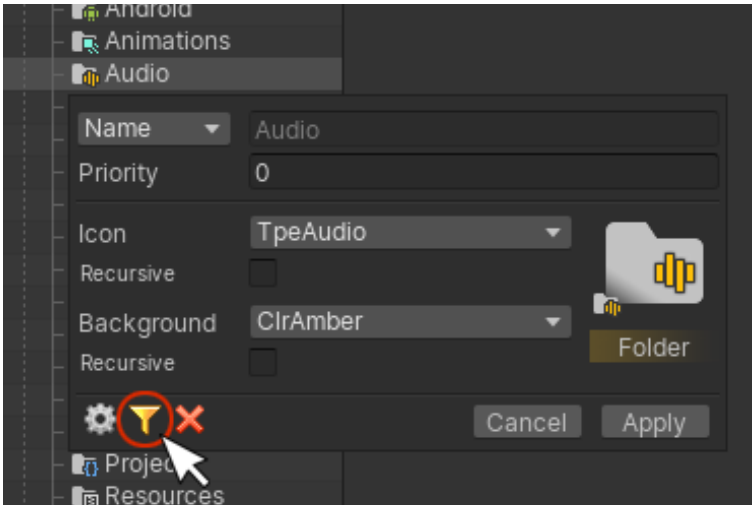
There is a reorderable list with all defined "folder" rules. You can modify existing rules, remove them using the **"-"** button, or add new ones by clicking the **"+"** button below.

If there is more than one rule for the same folder (including recursive assignments), then a rule with the **highest priority** will be applied. If the rules have the same priority then the **latest (lowest) item** in the list will be applied.

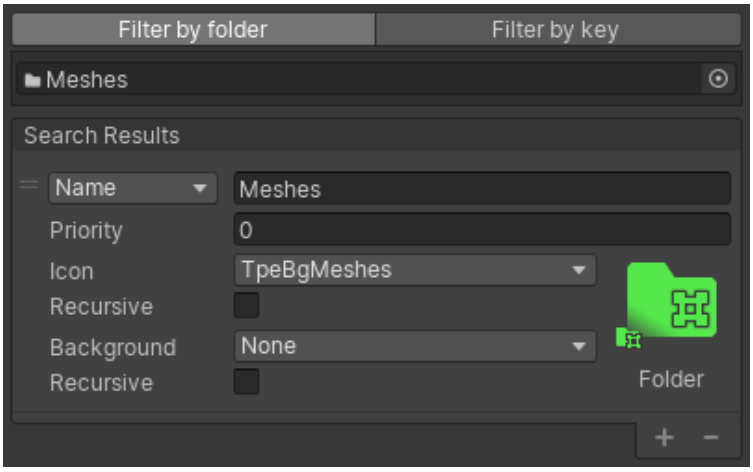




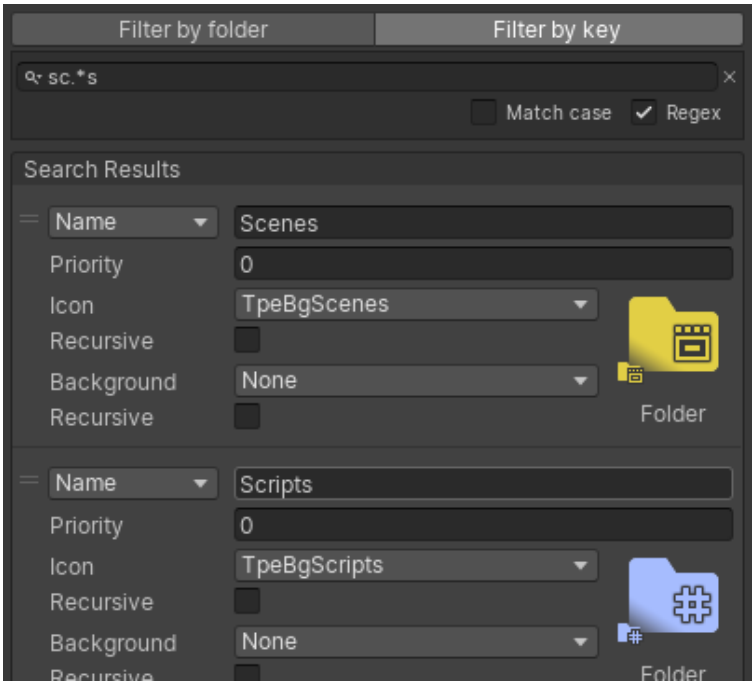
To view rules for a specific folder only, click on the **“filter”** button in the configuration dialog, then take a look at the Inspector. The whole ruleset should be filtered out and only rules related to this folder should be shown.



You can also achieve this by using the **“Filter by folder”** tab directly in Inspector.

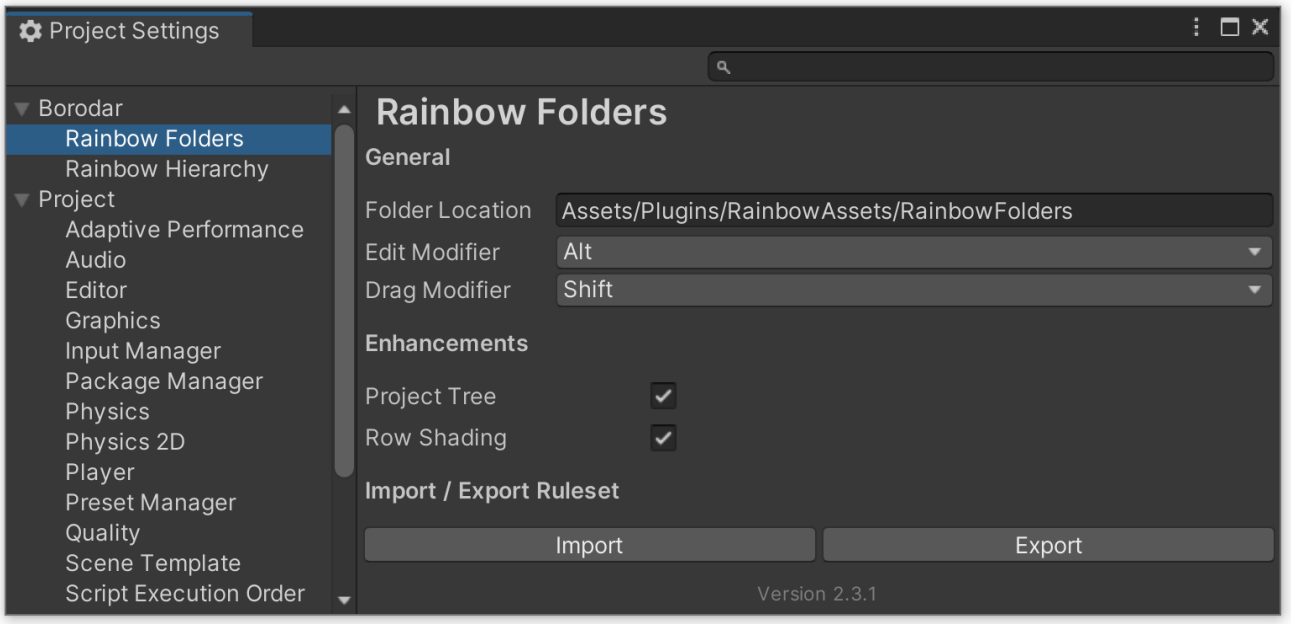


Also, you can filter rules out by their **“Key”** values using the **“Filter by key”** tab.



Import/Export Ruleset

The **Rainbow Folders** asset enables you to preserve and share your icons ruleset as a `.rfset` file. You can archive and store your preferred icon settings and use them for other Unity installations or make them available to your colleagues.



Exporting ruleset to .rfset archive

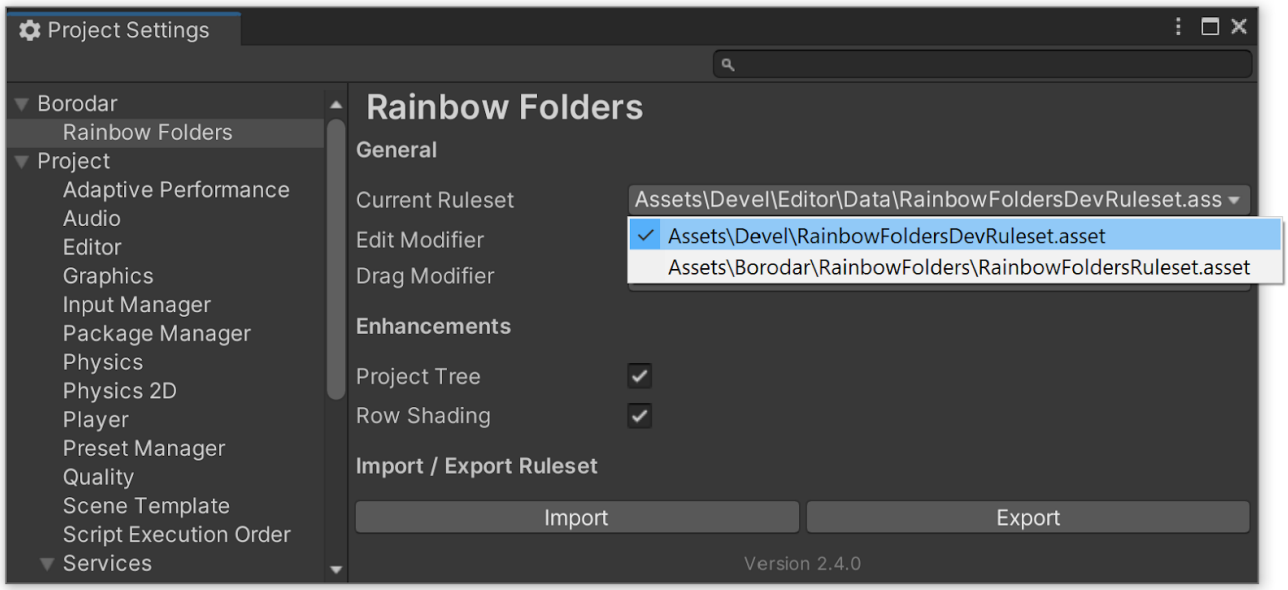
- Choose **Edit → Project Settings → Borodar → Rainbow Folders** from the main menu.
- Press the "**Export**" button below.
- In the **"Export Ruleset"** dialog specify a location where the ruleset archive will be saved.
- When exporting is done, you could find the `<you-project-name>.rfset` file in the specified folder.
- If your ruleset contains custom icons, they will be exported to the same folder as `<you-project-name>.unitypackage` file.

Import ruleset from .rfset archive

- Choose **Edit → Project Settings → Borodar → Rainbow Folders** from the main menu.
- Press the **"Import"** button below.
- In the **"Import Ruleset"** dialog choose a `<project-name>.rfset` file you want to import.
- If your ruleset contains custom icons and the corresponding `<project-name>.unitypackage` is located in the same folder, then it will be imported automatically into your project as well.

Multiple rulesets

If you have more than one ruleset, you can switch between them easily from project settings. Choose **Edit → Project Settings → Borodar → Rainbow Folders** from the main menu and select the required ruleset from the corresponding dropdown.



If you need more rulesets, you could select the default one in Project View and just duplicate it by pressing **Ctrl/Cmd+D**. Rainbow Folders will detect a new ruleset automatically.

Upgrading

Please always do a clean import of the **Rainbow Folders** package (delete the old version before importing the new one). Otherwise, you may receive a number of difficult-to-diagnose issues.

- Back up your ruleset using **Edit → Project Settings → Borodar → Rainbow Folders → Export** tool.
- Delete the `Assets/.../Borodar/RainbowFolders` folder.
- Import the new version from the package or from the Asset Store.
- Restore your ruleset by using **Edit → Project Settings → Borodar → Rainbow Folders → Import** tool.

Changelog

v 2.4.1

- Better compatibility with Unity 2023.1
- Several optimizations and small fixes

v 2.4.0

- Asset location is auto-detectable now, put it wherever you want in your project
- Added support for multiple rulesets
- Better support for Retina/HiDPI displays
- Reduced memory footprint by using lazy initialization for built-in textures

v 2.3.1

- New import/export tool that enables you to preserve and share your icons rulesets between projects or with colleagues
- A number of minor performance optimizations

v 2.3.0

- New icon/background selection dialogs instead of inconvenient context menus
- An additional set of “colored” folder icons per customers request
- Several optimizations and small fixes

v 2.2.0

- Improved compatibility with Unity 2019.3
- All icons have been completely reworked to match the new UI in Unity 2019.3
- A few minor optimizations and small fixes

v 2.1.0

- Added explicit “Priority” field for folder rules to avoid manual reordering when you have multiple rules for the same folder
- Added pagination when browsing through all the folder rules
- Added possibility to filter rules by specific folder
- Added possibility to filter rules by their “Key” value
- Several optimizations and small fixes

v 2.0.3

- Fixed bug regarding Editor performance degradation for projects that contain a large number of scriptable objects or large prefabs
- A number of minor performance optimizations

v 2.0.2

- Optimized performance when using a lot of custom folder icons applied by name

v 2.0.1

- Fixed bug with invalid Base-64 string when decoding the "edit" texture in Unity 2019.1+

v 2.0

- Folder icons are now actually replaced instead of drawing them on top of default ones
- All included icons now are baked into the code and will no longer bother you when picking/searching your own textures
- Added optional project tree outlines
- Added optional row shading
- Added “transparent” folder icons
- Improved compatibility with Unity 2018.3 and Unity 2019.1
- Fixed bug with wrong icons offset when fully zoomed out in “Two Column” mode

v 0.9.3

- Better compatibility with Unity 2018.2
- A number of small optimizations and fixes

v 0.9.2

- Fixed bug with missing EditorWindow when building player

v 0.9.1

- Fixed bug with possible conflicts with existing AssemblyInfo attributes from other assets
- Fixed typo for crimson color name in the various context menus

v 0.9

- Added option to apply custom backgrounds for folder names
- Added 24 background presets
- Better compatibility with Unity 2017.2

v 0.8.1

- Compatibility fix For Unity 2017.1
- Fixed bug with Unity Collaborate overlay when icons scale changed
- Fixed bug with Unity Version Control overlay when icons scale changed

v 0.8

- All icons have been completely reworked
- Added 12 new folder colors
- Added 12 new folder tags
- Added 3 new folder types (Animations, Physics, and Flares)
- Fixed bug when the Rainbow Folders menu item appeared on top of the context menu
- Fixed bug with applying icons recursively by name, when root icon didn't appear correctly

v 0.7.1

- Added support for Unity Version Control Overlay (beta)

v 0.7

- Added option to change modifier key for the configuration dialog

- Added support for Unity Collaborate (beta)
- Install “Rainbow Folders” to the Plugins folder by default
- Compatibility fix For Unity 5.6.0 beta

v 0.6

- Added option to apply custom icon for all subfolders automatically
- Compatibility fix For Unity 5.4.4

v 0.5.1

- Added back the context menu, according to numerous requests
- New platform icons (Android, iOS, Mac, WebGL, Windows)
- Minor bugfixes

v 0.5

- Improved workflow: change icons with alt-click on a folder, right in the project view.
- Added possibility to change icons for multiple folders at once
- Added option to move "Rainbow Folders" wherever you want in your project
- Got rid of the "Editor Default Resources" folder.
- Fixed the "Scripts" icon appearance for the dark theme
- Minor bugfixes

v 0.3

- Apply color "tags" for folders from context menu
- Support for colorizing/tagging folders from left column view in two-column layout
- Support for batch colorizing/tagging folders (select multiple folders and colorize/tag from context menu)
- Now keeping all settings assets in Editor Default Resources folder so they are not included in build.
- Changed namespace for Rotorz reorderable list, to avoid conflicts with existing installs of Rotorz plugins
- Fixed error messages in Unity 5.0.4x when loading settings
- New icons (Fonts, Shaders, Terrains, Meshes)

v 0.2

- A few more icons added
- Folder structure simplified
- Minor bugfixes
- Docs updated