Texture Overview for Unity

Table of Contents

1 Introduction

Texture Overview is an editor extension for Unity personal and professional, that is giving you a great summary view of all your textures with just the right information to aid in optimization.

Texture Overview is not a replacement for Unity's Project Browser.

2 Features

- Lists textures in your project, current scene, a folder, an asset bundle manifest or of any selected asset
- Sorts textures by Storage Size, Runtime Size, Format, Sprite Packing Tag, etc. Very easy to see which textures consume the most memory.
- Filters textures by type (Sprite, Normalmap, Cubemap, GUI, etc), name and path
- Detects a few texture compression issues
- Finds materials and prefabs that use a particular texture (can be slow)
- Exports content as CSV to generate fancy graphs about memory usage in spreadsheet applications.

- Used in projects that contain several thousands of textures .

3 What users say about Texture Overview

The following quotes are copied from the asset stores' Texture Overview review section ([here](#) and [here](#)).

> *The plugin is fast and easy to install, unobtrusive, and provides very useful information about textures. Its great to get a quick look at only the textures in your scene arranged by memory usage, to get a really good idea of where your memory is going. Its good for optimizing because you can see textures that might not be worth their size, or certain props that are dragging in more texture than their worth visually.*

> *The project works well and is intuitive. It helped me identify memory hog textures and put them in their place. The support from Peter is solid; he is committed to his assets.*

> *Great at giving you a summary view of all your textures with just the right information to aid in optimization. Well worth the money.*

4 Contact

If you have a question, want to provide feedback or suggest a new feature, please use the Texture Overview forum thread:
http://forum.unity3d.com/threads/197707-Released-Texture-Overview-Plugin

In case you want to keep it private, send me an email at unitytools@console-dev.de

5 Installation

Before importing the TextureOverview package, make sure your project does not have any compile errors.

Files in the „TextureOverview" package import to "Assets/Editor/TextureOverview" by default. Once you imported the package, you can find "Texture Overview" in the main menu under the "Window" popup.

In case you want it to be located under a different popup, you can change the location in

"Assets/Editor/TextureOverview/ TextureOverviewMenuItem.cs". Simply change the line `[MenuItem("Window/Texture Overview")]` to whatever fit your needs.

5.1 Source Code

If you purchased "Texture Overview Pro", the package also contains the full C# source code to build Texture Overview. The source code is stored in a separate zip archive found in the installation directory. If you use the default installation directory, it is: "Assets/Editor/TextureOverview/TextureOverviewSource.zip".

The source code can be built using Microsoft Visual Studio. I tested it with VS 2012 Express. You can find instructions how to build the project in "readme.txt", which is part of "TextureOverviewSource.zip".

Do **not** unzip the source code to any directory under the "Assets" folder in your project, as this will conflict with the prebuilt DLL version.

6 The first start

Starting Texture Overview after you imported it from the Asset Store, will first display a "Installation window", which is responsible to install the correct plugin version for your Unity installation. The installation usually takes a few seconds only and does not need any user interaction.
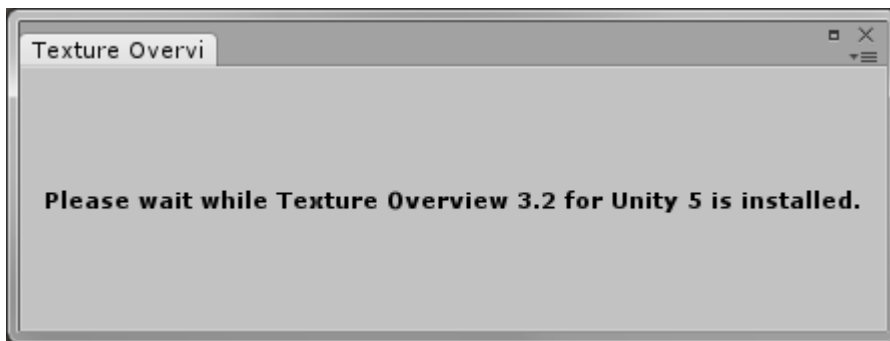
After the installation has been completed, Texture Overview automatically opens, starts to read texture settings from your project and stores them in a persistent Cache File. Reading texture settings might take a while, depending on the number of textures in your project (from a few seconds to several minutes).

However, the next time you open the plugin, it will be much faster (instant)!

Cache File (the technical part)

Texture Overview automatically creates a file called "TextureOverview.cache" in the

projects "Library" directory. Think of this cache file as a database with texture settings.

"TextureOverview.cache" must **not** be added to revision control (Subversion, Perforce, Git, Mercurial, etc). This file is very specific to the state of the Unity project on your computer only.

The .cache file contains texture settings that have been read by Texture Overview. Caching this information allows Texture Overview to start much faster when you open it another time, since the plugin does not need to read all the texture settings over and over again, which is an expensive operation.

Instead, it only needs to check if textures have been modified since the last start of the plugin. If textures have not changed, Texture Overview can use settings from the cache file, which is fast. If a texture did change, Texture Overview will read the new settings and update its cache file, which is slower.

Texture settings usually need to be updated when when the texture has been modified, its import settings are changed or when the target platform is switched in Unity.

7 Source Selection

Texture Overview displays textures either from the Project, Scene or Selection. Use the drop-down menu as shown in the picture below to select the source from where Texture Overview will display textures.
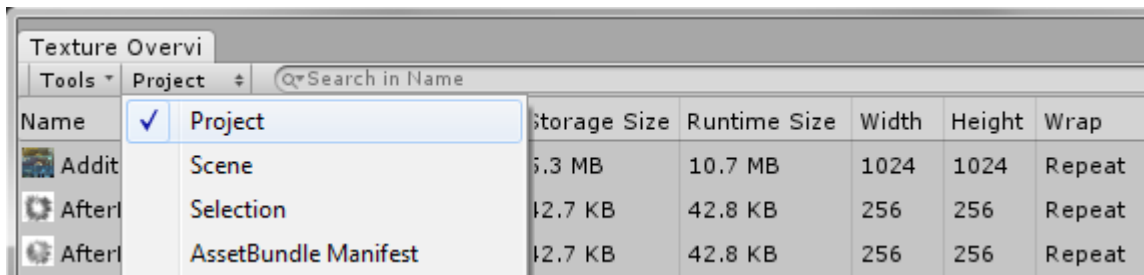
Project

Displays all textures located in the project.

Scene

Displays textures that are referenced by game objects in the currently open scene(s).

Selection

Displays textures that are used by the selected asset in the Unity project window or from the selected GameObject in the Hierachy window.

AssetBundle Manifest

Displays textures that are referenced by an asset bundle manifest file. Unitys BuildPipeline outputs a manifest file for each asset bundle that is generated. Such manifest file contains which assets have been explicitly added to an asset bundle. Asset dependencies are not resolved.

Texture Overview does not load the actual asset bundle content. It uses the manifest file to detect which assets are stored in a bundle. It displays texture settings from textures located in your project, not the ones stored in that asset bundle.

Choosing the "AssetBundle Manifest" mode displays a new panel at the left side of the Texture Overview window. The "Open..." button can be used to select the directory where your *.manifest files are stored.
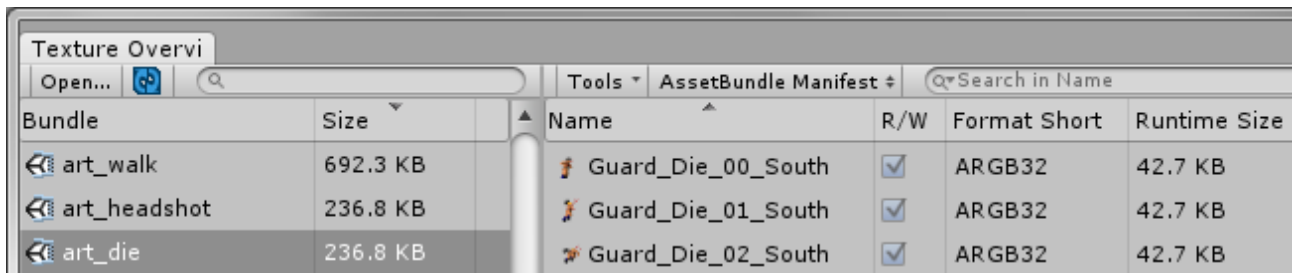
Texture Overview scans the selected directory and displays each *.manifest file it could find. If a file with either no file extension or .unity3d file extension is located next to the .manifest file (which is usually the actual asset bundle), its size is display in Texture Overview as well.

8 Platform Selection

The top-right area in the toolbar displays buttons to choose the platform for which Texture Overview will display texture settings.
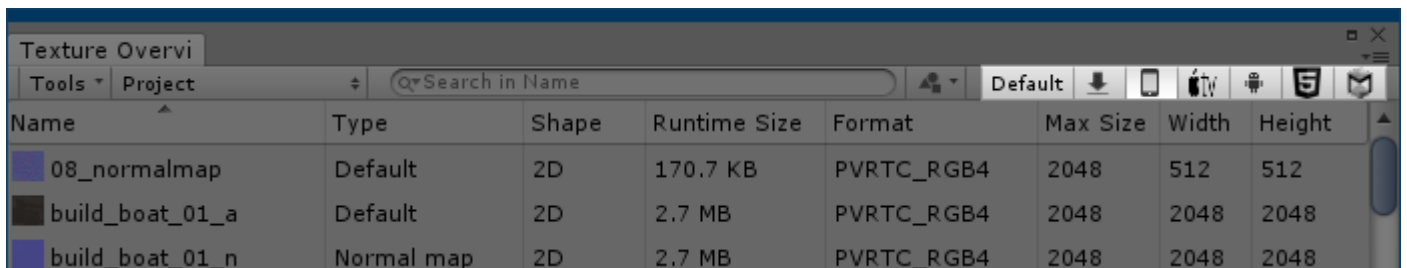
The platform buttons that appear in Texture Overview are the installed Platform Modules in Unity (the platforms you select when installing Unity).

The "Default" option will leave a few columns blank, such as Runtime Size, Format, Width

and Height. You need to select an actual platform to display these values, such as "PC, Mac & Linux Standalone" or "iOS".
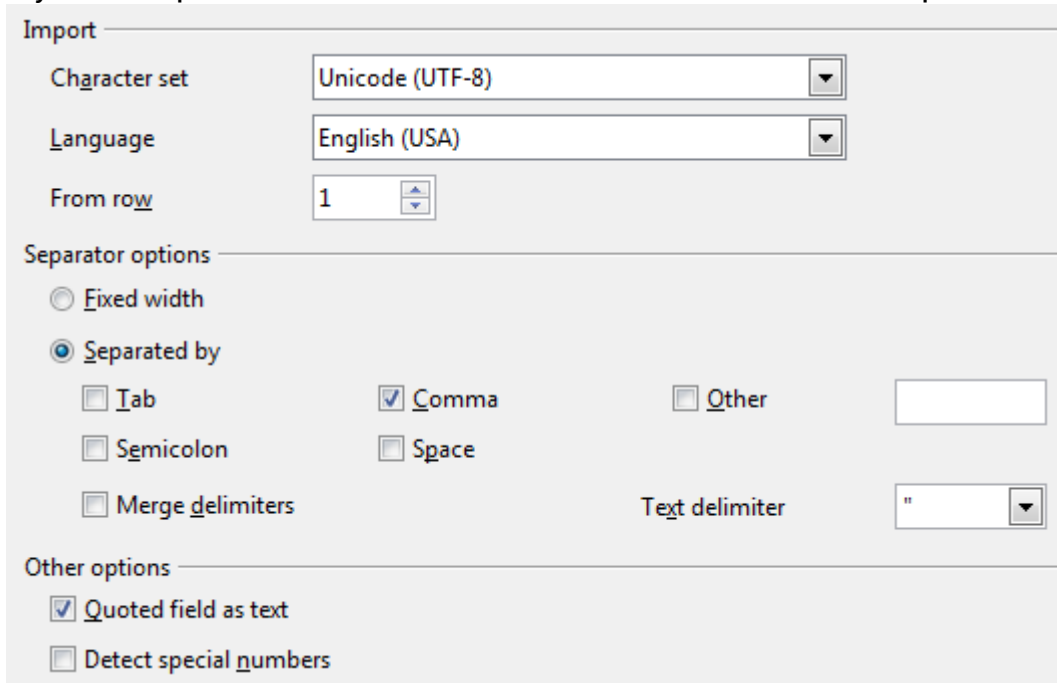
9 Tools menu

The „Tools" button in the toolbar opens a popup menu with functionality as decribed below.

9.1 Export CSV

Export the list content as CSV (comma separated, text put in quotes). The exported CSV file can be used in spreadsheet applications to generate fancy graphs about memory usage for example.

If you use Open Office to view the CSV file, these are the CSV import settings:

9.2 Memory Usage Overlay

Displays an overlay window that shows the runtime, storage as well as graphics and main memory usage of textures currently displayed in the list.
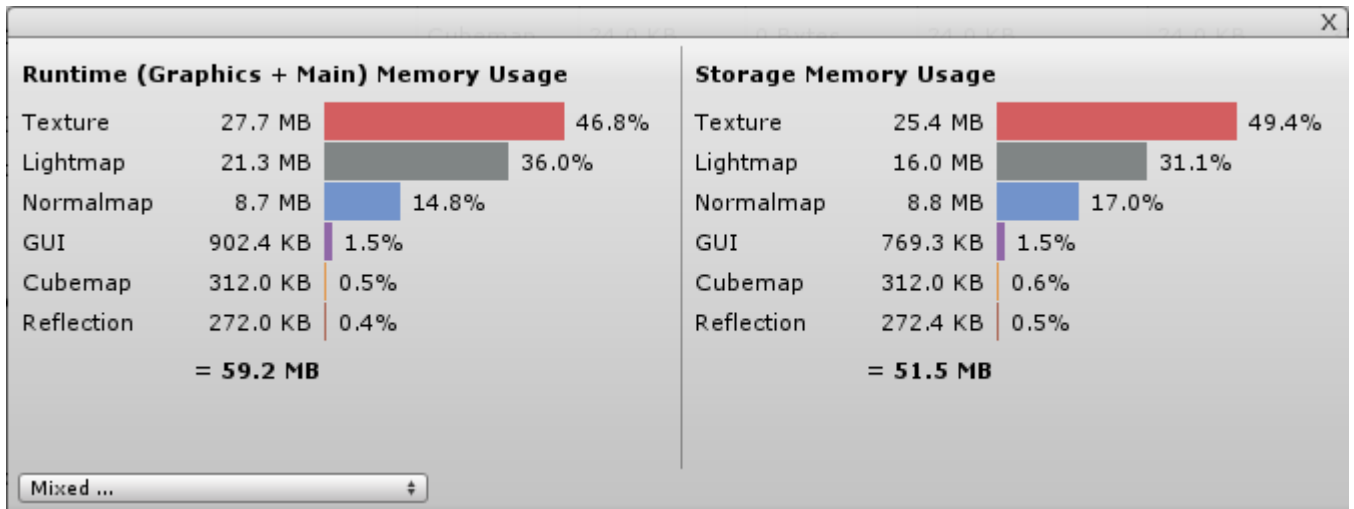
Storage memory usage for "Crunched" textures is a worst-case estimate only, not the actual filesize.
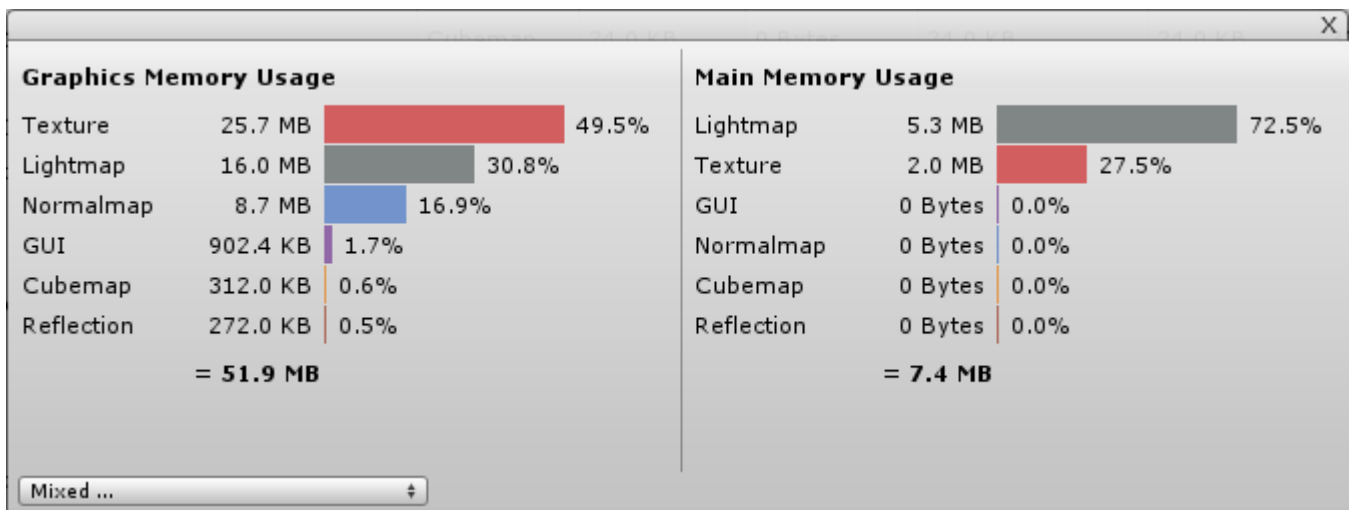
As you filter textures in the list, you see the statistics to change accordingly. The memory usage overlay gets hidden if "Default" is selected as platform.

9.3 Advanced > Handle Rgb24 as Rgba32 on GPU

Most, if not all graphics card expand RGB24 textures to XRGB32 (where X is ignored). This means, a RGB24 texture costs the same amount of memory as a RGBA32 texture on the GPU.

This option makes sure that Texture Overview calculates GPU memory consumption for RGB24 texture formats as they were XRGB32, to better match how much memory the texture actually costs on the graphics card.

The default value for this option is ON.

9.4 Advanced > Count Renderer (EXPERIMENTAL)

The "Count Renderer" option is available in "Scene Mode" only.

It is counting how many Mesh and SkinnedMeshRenderer in the scene reference a texture via their Renderer.sharedMaterials property.

The count for each texture is displayed using the "Renderer Count" column, which can be activated by right-clicking a column header.

This feature is currently EXPERIMENTAL only.

9.5 Advanced > Show Editor Assets

If this option is enabled, Texture Overview also displays textures that are located in directories that Unity considers as editor assets, such as: "Editor", "Gizmos" and "Editor Default Resources".

The default setting is off, which will not show editor assets.

9.6 Searching

The textfield in the toolbar can be used to search/filter the list. The search checks if any of the provided words occur, for example, in the texture name (depends on search mode). The search performs case insensitive.

Provide multiple words, separated by a space character, to narrow the search even further. If you need to search for a name that contains a space character, you can put the word in quotes, „like this".

Search Modes

The magnifying glass next to the search field can be used to choose on which property of a texture the search term is tested against.
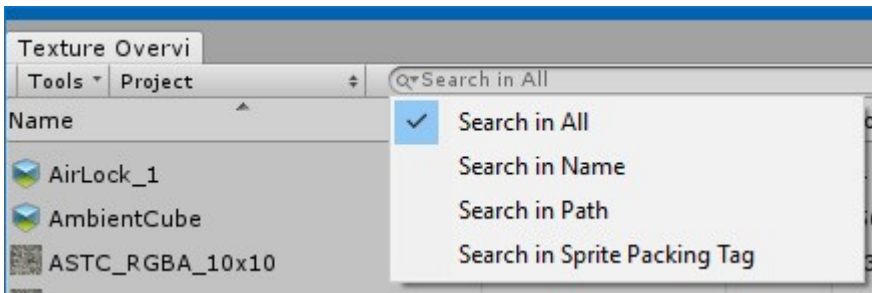
„Search in All" searches in all available options in the list: Name, Path and Sprite Packing Tag.

Search in Folders

If you want to search for items in a certain folder, you can switch the search mode to „Search in Path" and then simply drag&drop one or multiple folders from the Unity project window onto the search field in Texture Overview

The following search shows all items that are located either in a folder under „Assets/Textures/Effects".
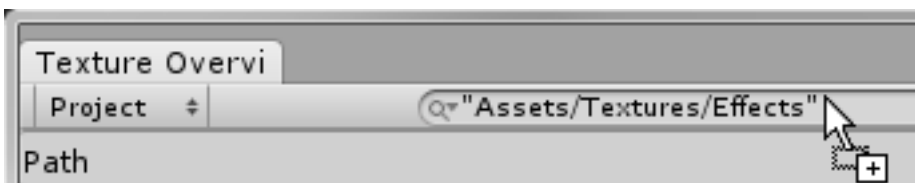
Search Operators

If you're not finding what you're searching, try a search operator. Add these symbols to your search terms in the search field to gain more control over the results.

These search operators are heavily influenced by the C# and Java language, because Unity users are most likely familiar with one of these languages.

Search for either word a || b	If you want to search for items that may have just one of several words, include || between words. Without the || symbol, your results would only items that match both terms.
Exclude a word !a	Add an exclamation mark (!) before a word to exclude all results that include that word.
Search for both words a && b	The „and" operator is the default when no operator is specified. If you want to search for items that contain both words, you can either separate those with a space, or include the && between words.

Search Examples
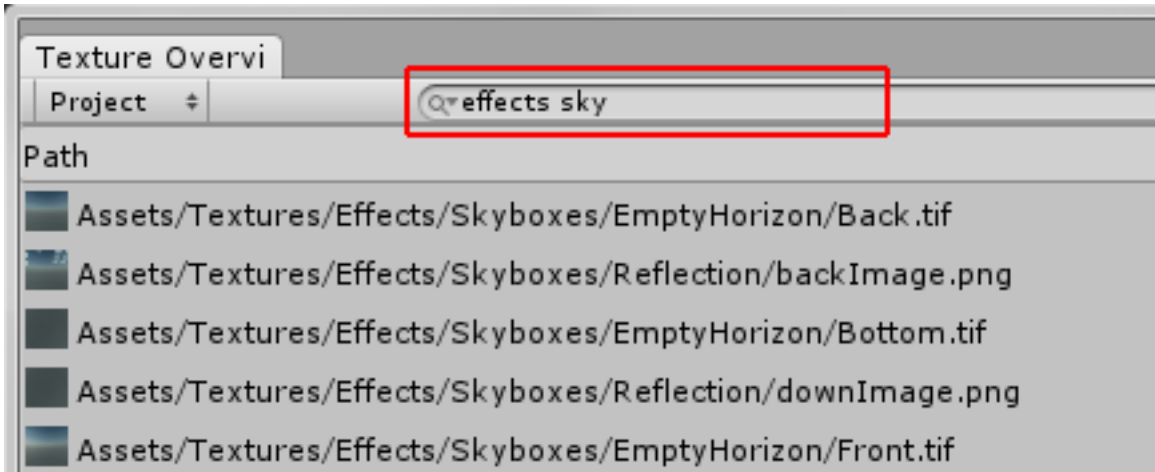
Find all items that contain „effects" and „sky" in their path:

Find all items that contain „tif" or „png" and „reflection" in their path:
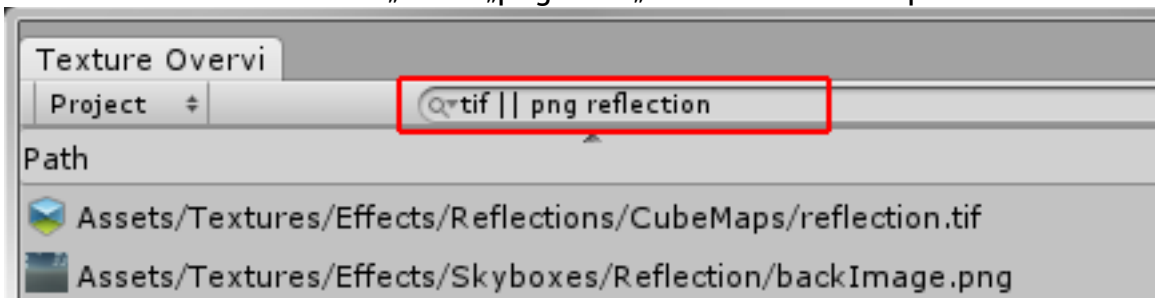
Find all items that contain „tif" or „png" and „reflection" and not „image" in their path:
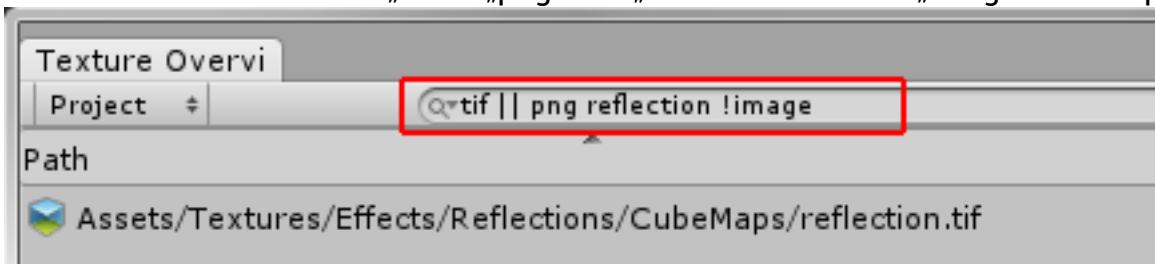
The following terms are equivalent, because the „and" operator (&&) is the default one that gets substituted when no operator is specified:

effects sky	effects && sky

tif || png reflection	tif || png && reflect
tif || png reflection !image	tif || png && reflect && !image

9.7 Filter by texture type

You can filter textures of a certain type. Use the [icon] toolbar button to open a popup-menu with various texture types to choose. Select texture types you want to display in the list. If at least one type is selected, the toolbar button turns yellow to indicate a texture type filter is active.
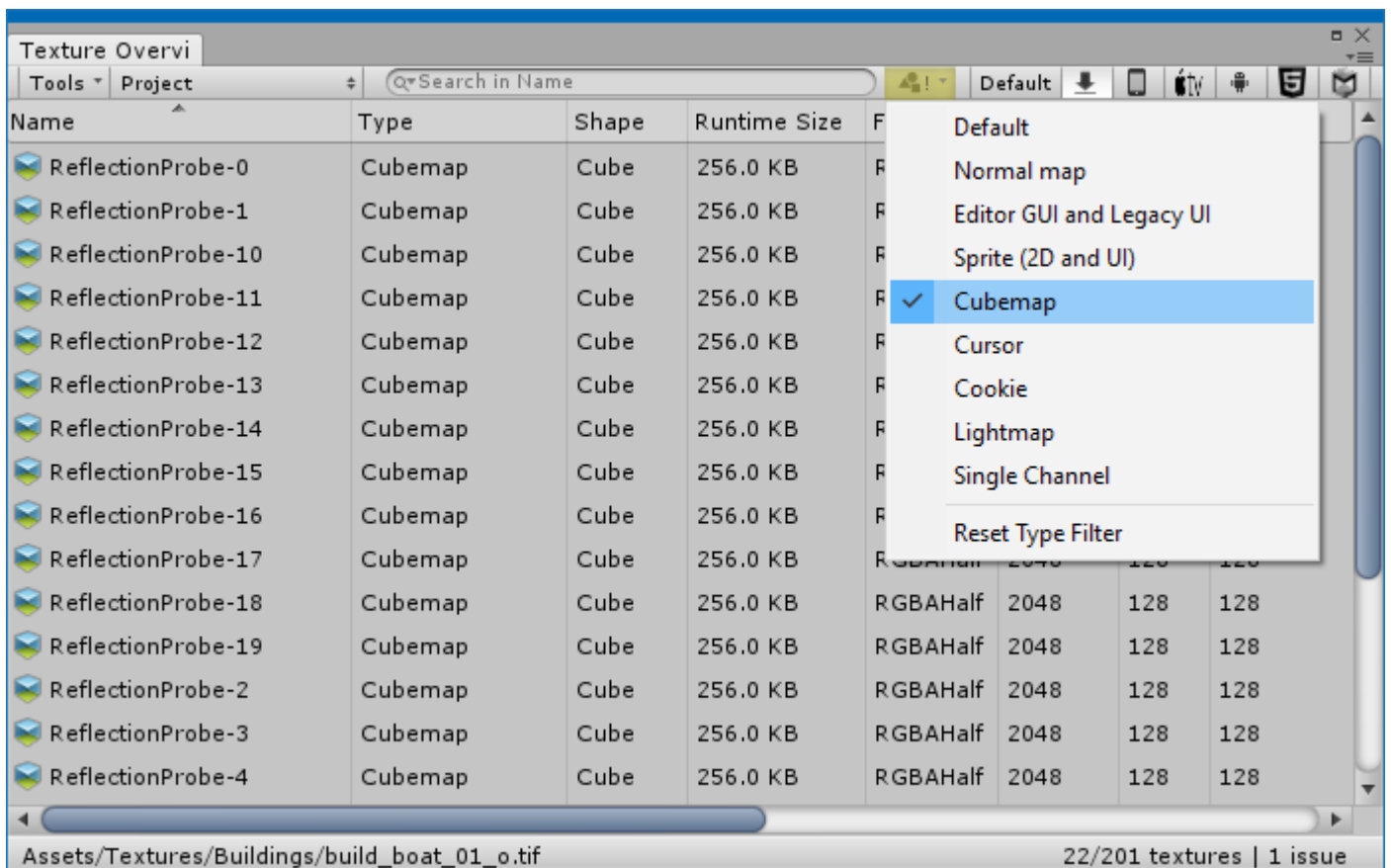
10   Infobar

The bottom bar displays information about the current selection, the active build platform as well as how many textures are in the list.

build_building_01_o	Default	2D	682.7 KB	PVRTC_RGB4	8192	1024	1024
build_building_01_sg	Default	2D	2.7 MB	PVRTC_RGBA4	2048	2048	2048
build_building_02_a	Default	2D	10.7 MB	PVRTC_RGB4	8192	4096	4096
build_building_02_n	Normal map	2D	10.7 MB	PVRTC_RGB4	8192	4096	4096
build_building_02_o	Default	2D	682.7 KB	PVRTC_RGB4	8192	1024	1024
build_building_02_sg	Default	2D	2.7 MB	PVRTC_RGBA4	2048	2048	2048

4 selected | Storage: 24.7 MB | Runtime: 24.7 MB   201 textures | 4 issues

From left to right:

4 selected
Displays how many textures are currently selected in Texture Overview

Storage: 24.7 MB
Displays how much storage memory the selected textures consume.

Runtime: 24.7 MB
Displays how much runtime memory the selected textures consume.
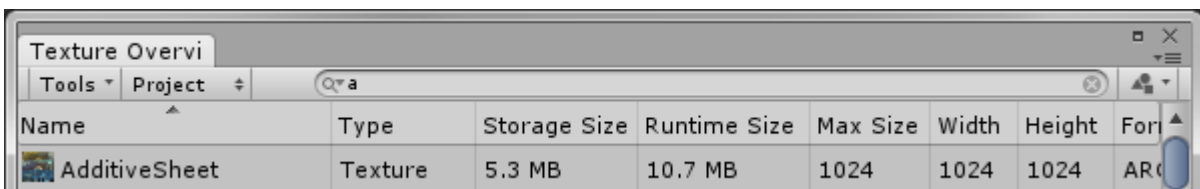
201 textures
The total number of textures currently in the list. This number changes as you filter the list for example.

4 issues
The total number of issues for textures currently in the list.

11   Columns

Texture Overview displays settings using columns. Several columns are hidden by default, which can be activated by clicking with the right-mouse-button on the column header.

Name	Type	Storage Size	Runtime Size	Max Size	Width	Height	For
AdditiveSheet	Texture	5.3 MB	10.7 MB	1024	1024	1024	AR(

Texture Overview supports to display the following settings:

Name

Asset filename without file-extention.

Path

The full asset path relative to the project directory, eg
"Assets/GFX/Textures/Explosion.psd".

Type

Indicates for what the texture will be used for. For example "Normalmap", "GUI",
"Reflection", "Sprite", etc. This setting is displayed in the Unity Texture Inspector as
"Texture Type".

Storage Size

How much memory the asset consumes on the storage device (disk).
Please note, for "Crunched" textures, this is only a worst-case estimate and not the actual
file size.

Runtime Size

How much memory the asset consumes during runtime in memory, which is the sum of
graphics and main memory consumption.

Graphics Memory

How much memory the asset consumes during runtime in graphics memory. Please also
see advanced option Handle Rgb24 as Rgba32 on GPU.

Main Memory

How much memory the asset consumes during runtime in main memory. This applies to
uncompressed and DXT compressed textures that have the "Read/Write Enabled" setting
turned ON. In this case, Unity keeps a copy of the texture in memory.

Source Width

Width in pixels of the original texture, prior import.

Source Height

Height in pixels of the original texture, prior import.

Max Size

Maximum allowed texture size for the selected platform in Texture Overview (platform

buttons in toolbar).

Width

Width in pixels of the imported texture for the select platform in Texture Overview.

Height

Height in pixels of the imported texture for the select platform in Texture Overview.

Resize Ratio

The ratio of the number of pixels in the source and imported texture (root mip-map). If the source texture is 1024x1024 and the imported texture is 512x512 pixels only, the resize ratio is 25%, because the imported texture is only a quarter in pixel-size. A really low resize ratio often causes texture detail being lost.

Format (full)

The full or long name of the internal representation used for the texture for the selected platform in Texture Overview.

Format

Same as "Format", but in a short form. Tech-artists and programmers most likely prefer this over the friendly readable "Format (full)" display.

Compression Quality

iOS/Android: Quality of texture compression used during import in the editor. This can be either Fastest, Normal or Best. On some build targets an integer number is displayed instead, where lower numbers correspond to "Faster" and higher numbers to "Better".

Resize Algorithm (in Unity 2017.2 and newer only)

Indicates what algorithm Unity uses to downsample the texture during import.

Wrap

Texture wrap mode (Clamp, Repeat).

Filter

Texture filtering mode (Point, Bilinear, Trilinear).

Aniso

Anisotropic filtering level. Displayed as "Aniso Level" in the Unity inspector.

Read/Write Enabled

Indicates whether "Read/Write Enabled" is checked in the Unity inspector (Advanced view).

Power of 2

Indicates whether the imported texture size of the selected platform in Texture Overview is power to two.

Mips

Number of map-maps in the texture. 1 indicates the texture has no mip-maps (only the root "mip").

Mips Enabled

Indicates whether the „Generate Mip Maps" setting is ticked in Unity's Texture Inspector.

sRGB (Color Texture)

Indicates whether the „sRGB (Color Texture)" setting is ticked in Unity's Texture Inspector.

Extension

File-extension of the source asset, eg "psd".

Sprite Mode

Indicates how the sprite graphic will be extracted from the image.

Sprite Packing Tag

Displays the name of an optional sprite atlas into which this texture should be packed.

Sprite Pixel per Unit
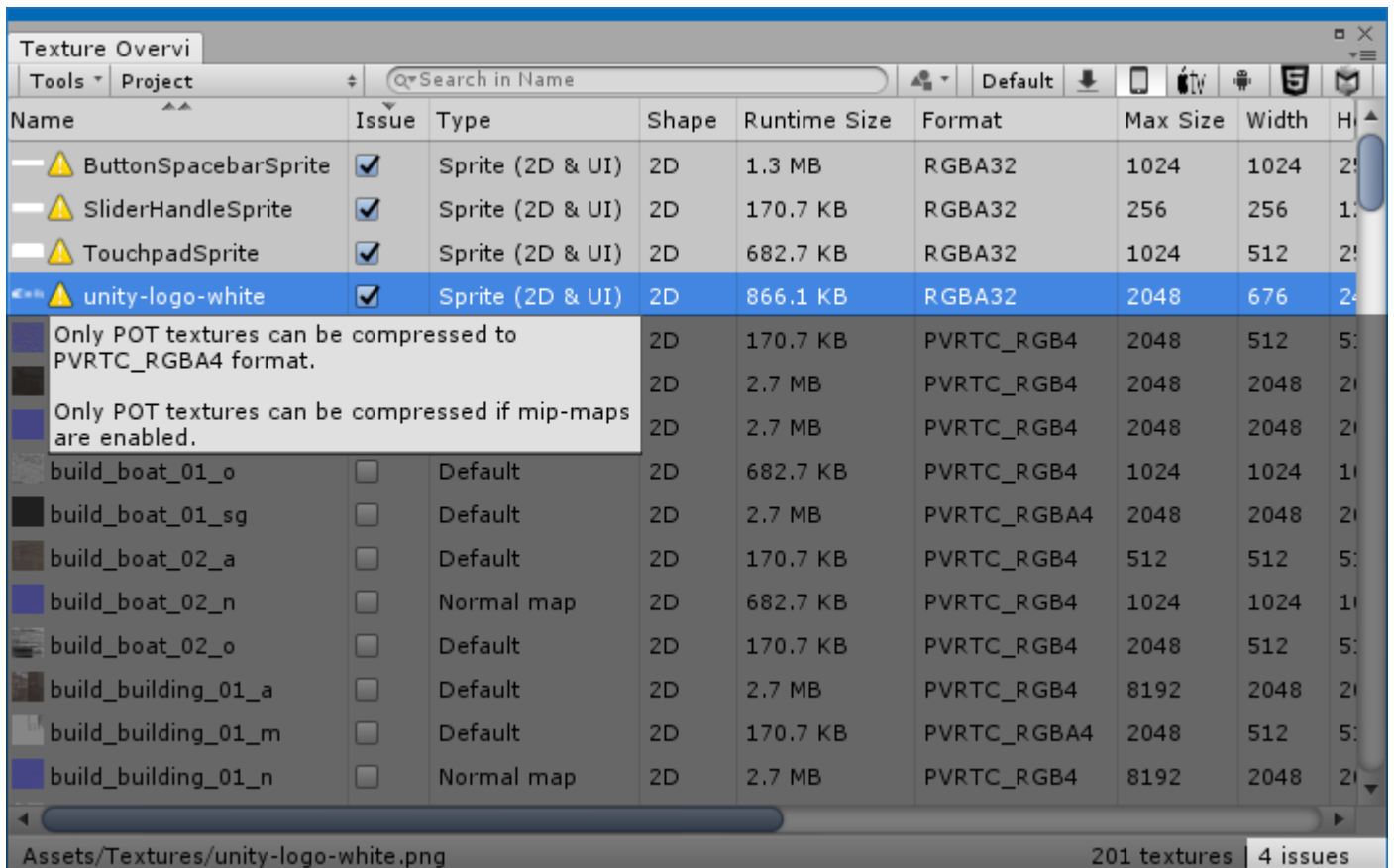
The number of pixels of width/height in the sprite image that will correspond to one distance unit in world space.

12   Issue detection

Texture compression is often used to save storage- and bandwith-memory, to allow for more efficient rendering. However, not every texture can be compressed. There are requirements that must be fulfilled first and if they are not, Unity picks an uncompressed texture format instead, causing memory go up quickly.

Unity displays a warning in the Texture Inspector window if such requirement isn't fulfilled and that's useful. However, Unity's Texture Inspector displays this for one texture and only for the active build platform. It's really easy that inefficient textures sneak into the project and keep there undetected until someone with a technical understanding selects that texture.

Texture Overview tries to make these issues more prominent, by displaying the number of detected issues in the lower right corner. A warning icon is displayed as well and hovering the mouse over that icon reveals the warning text as tooltip.

But, how to find these issues quickly? The idea is that you can enable the „Issue" column in Texture Overview and simply sort for issues.

In the image above, I right-clicked any column header and turned on the „Issue" column. I then clicked on the „Issue" column that Texture Overview sorts the list by issues. Holding Ctrl and clicking the „Name" column makes Texture Overview show all issues sorted by texture names.

Then simply go over the list, double click an entry to assign the texture to Unity's Texture Inspector and fix its settings or adjust the source texture.

Texture Overview currently detects these issues, which are often depending on the target platform and compression format.

- Compressed texture uses mip-maps, but its size is not a power-of-two.
- Compressed texture width/height is not a multiple of <blocksize>.
- Compressed texture size is not a power-of-two, thus fallbacks to an uncompressed format.
- Compressed texture is not square, thus fallbacks to uncompressed.
- The source image is provided as a lossy image, such as .jpeg (can be turned off in „Warnings" menu).
- Legacy Cubemaps are used (can be turned off in „Warnings" menu).

13   Context Menu

Click with the right mouse button on a selection to display the context menu. The context menu provides various useful tools, such as copying the path to the clipboard, showing the texture file in Explorer/Finder or selecting all materials in your project that use the selected texture(s).
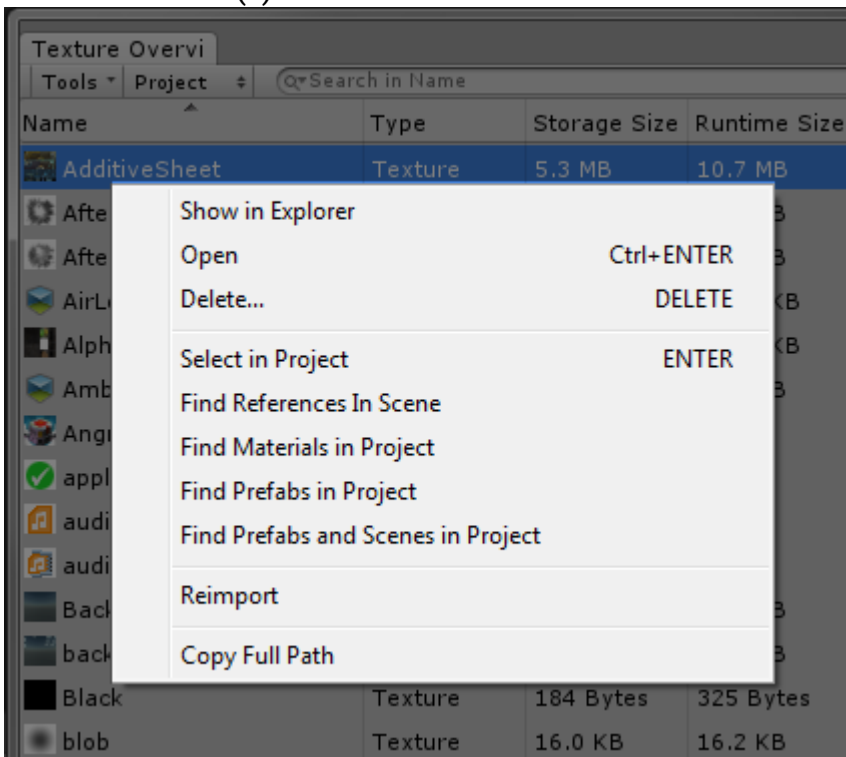
Show in Explorer / Reveal in Finder

Shows the selected textures in the operating-systems file-system browser.

Open

Opens the selected textures in the associated application.

Delete

Deletes the selected textures from the Unity project.

Select in Project

Selects the selected textures in the Unity project- and assigns them to the Unity inspector-window. This function requires to load the texture assets. If the selection contains many textures (thousands), this might take a while to complete and could even cause Unity to display an out-of-memory error. However, a progressbar is shown during this time that can be used to cancel the operation at any point.

Find References in Scene

Inserts the name of the selected texture into the Hierarchy-window search field.

Find Materials in Project

Checks every material file in the project if it's referencing any of the selected texture(s). The result is displayed in the Find Usage Results window. This function can take a while to complete, depending on the number of materials in the project, but it can be canceled at any time.

Find Prefabs in Project

Checks every prefab file in the project if it's referencing any of the selected texture(s). The result is displayed in the Find Usage Results window. This function can take a while to complete, depending on the number of materials in the project, but it can be canceled at any time.

Find Prefabs and Scenes in Project

Checks every prefab and every scene file in the project if it's referencing any of the selected texture(s). The result is displayed in the Find Usage Results window. This function can take a while to complete, depending on the number of materials in the project, but it can be canceled at any time.

Reimport

Forces an asset reimport of the selected textures.

Copy Full Path

Copies the asset path, including file-extension, of the selected textures into the clipboard.

13.1 Find Usage Results

The "Find Usage Results" window displays the result of a "Find … in Project" operation, such as "Find Materials in Project" and "Find Prefabs in Project".
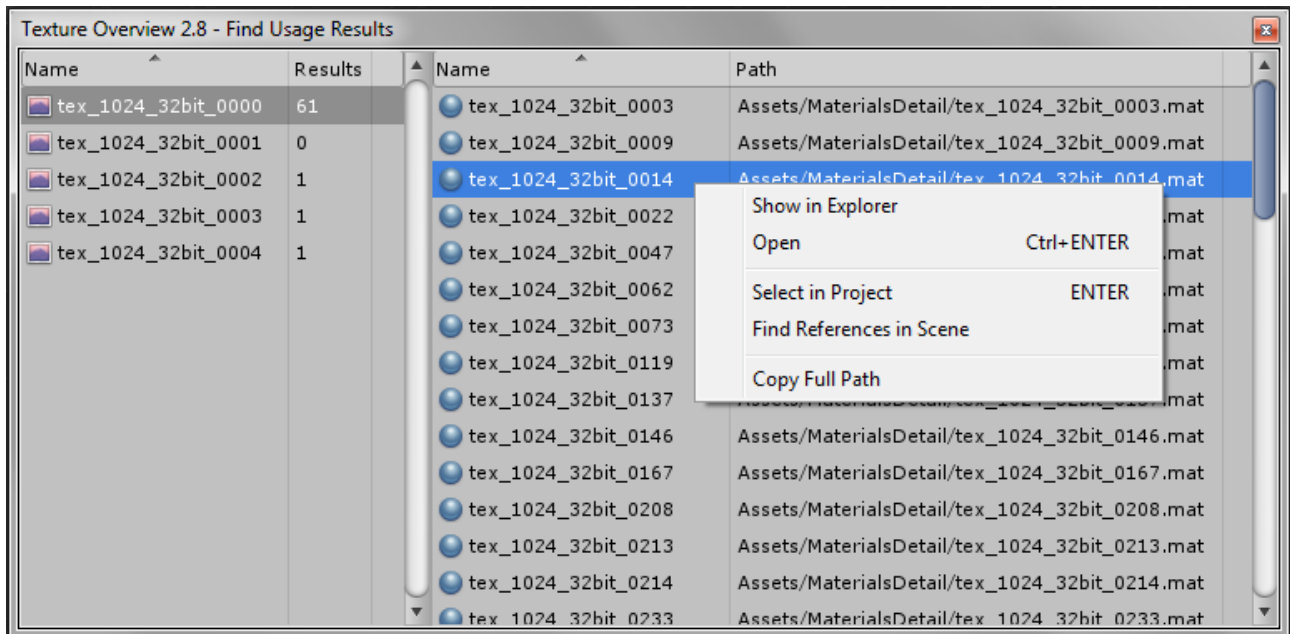
The list at the left side contains the textures you searched for. The list at the right side, displays which assets (materials, prefabs, scenes) use the selected texture.

The "Results" column in the left list displays how many assets have been found, that use the particular texture.

A right-click with the mouse opens a context-menu with helper functions, such as showing the asset in Explorer/Finder.

A double-click with the left mouse button assigns the selected asset to the Unity Inspector-window.

14 Sorting

Each column can be sorted in ascending and descending order by simply clicking with the left mouse button on the particular column header. An up/down arrow indicates the sorting mode.

You can add an additional sort criteria by holding down the Control-key and clicking another column header. This comes in handy if you sort by „Format" for example, but still want to see textures sorted alphabetically by „Name". In this case, you sort by Format first and add the Name column as second sort criteria.
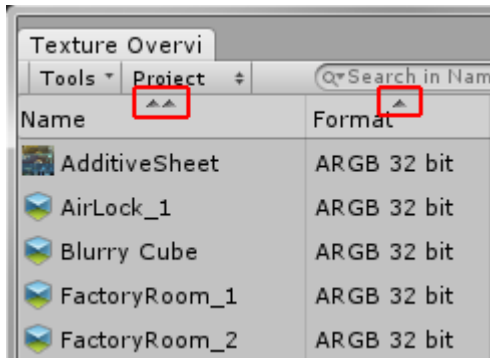
14.1   Organize Columns

If you feel like you need columns in a different order, click with the right mouse button in the column header and choose 'Organize Columns' in the popup.

This will open a new window where you can re-arrange columns in a drag&drop fashion, as well as changing the visibility of every column.

15   FAQ

Is the tool showing Unity's automatically generated atlas textures?

No. Texture Overview does not display information about those Unity generated atlas textures, such as the „Sprite Atlas" asset and the atlases generated using the legacy „Sprite Packing Tag". They might not even exist in the editor, depending on your „Enable atlas in build only" option.

Can I use Texture Overview to change texture seetings?

No. Texture Overview does not support to change texture settings. However, changing texture settings is just a double-click away. Just double-click or press Enter to assign a Texture to Unity's Inspector, where you can change its settings.

Why is „Storage Size" and a few other columns empty?

Texture Overview shows various empty columns if the selected platform in Texture Overview is set to „Default". In order to display all the information, you need to choose an actual platform using the platform buttons in the upper-right corner of the Texture Overview window.

Why do I get a „UnityEditor could not be found" when building the game?

Texture Overview must be installed under a directory named „Editor". The default installation path is „Assets/Editor/TextureOverview/". If you move your third-party packages around, you need to follow this Unity requirement and place editor extensions under a directory called „Editor".

Why do textures show up as RGB24 or RGB32 even when I set them to DXT or ASTC compression?

This occurs when you don't use „Compress assets on import". You can find this setting in Unity „Edit > Preferences..." under the „General" tab.

Texture Overview display a message that it cannot run after I upgraded Unity!

Re-install Texture Overview from the asset store, this should solve the issue. Texture Overview installs for a specific Unity version, precisely the Unity version that you used when you started Texture Overview for the first time. If you upgraded Unity in the meantime, you also need to install Texture Overview again.

16   Known Issues

Crunched textures don't show the correct „Storage Size"

This observation is spot on. The reason is I can't calculate how much memory the texture would have after Crunch compression. I would have to read the actual file-size, however the file is only available in Unity for the active build platform.

Therefore Texture Overview displays a worst-case estimate and not the actual file size for „Crunched" textures.

„Please wait while Texture Overview is installed" message stays after update

This issue occurs sometimes / rarely after updating Texture Overview to a new version. You can workaround this issue by just focusing another application and then switching back to Unity. If the focus-trick didn't work, restart Unity.

„Find Prefabs, Materials and Scenes" takes so long

In order to find if a texture is used by a particular asset, Texture Overview loads all these assets (prefabs/materials/scenes) and checks if that texture is a dependency of such asset. Depending on the number of prefabs/materials/scenes in your project, this can take some time (seconds to minutes). Subsequent searches during the same Unity session are often much faster though.

Tizen platform is missing in Unity 5.5

There is a bug that causes Unity to crash when querying Tizen texture settings. In order to prevent that Texture Overview might cause such crash, I excluded Tizen for now when using Unity 5.5. I will include it once Unity fixed the crash.
https://issuetracker.unity3d.com/issues/crash-getautomaticformat-tizen-causes-textureimporter-formatandusagefromparameters-on-certain-texture-formats

17   Changelog

Version 4.9

- Fixed ETC2 issue detection. Texture Overview was showing an issue for non-power-of-two ETC2 textures without mip-maps always. However, this is actually a valid configuration, as long as it's a multiple of 4.

Version 4.8

- Added Unity 2019.1 alpha compatibility
- Checked Unity 2018.3 compatibility
- Fixed „jpeg" textures not showing up in the list, but „jpg" textures did.
- Added option to toggle issue detection if texture compression fails. This was hard-coded to be checked always prios version 4.8.
- Added issue detection if a source image uses a lossy compression format, such as adding .jpeg to the project. Can be turned off in „Tools > Issue Detection" menu.
- Added issue detection if Legacy Cubemap assets are found. Can be turned off in „Tools > Issue Detection" menu.

Version 4.7

- Unity 2018.2 alpha compatibility
- Fixed option „Advanced/Show assets from Editor directories". It now also detects assets located in a directory called „Editor Resources" as editor assets.
- Added „Advanced/Show assets from Packages". This lets you include/exclude textures that are pulled in through Unity's Package Manager. (Unity 2017.3 newer only)

Version 4.6

- Fixed IndexOutOfRangeException error in Unity 2017.3 and newer, if android project contains a texture in ETC_RGB4Crunched or ETC2_RGBA8Crunched format.

Version 4.5

- Unity 2018.1 beta 6 compatibility
- Added „R8" format (available in 2018.1 and newer only)

Version 4.4

- Unity 2017.3 beta 8 compatibility
- Checked Unity 2017.2.0f3 compatibility

Version 4.3

- Ignore textures in „StreamingAssets" as these can't be read.
- Added option to filter textures in editor folders (Editor, Gizmos, Editor Default Resources). The default setting is off. You can change this in the „Tools/Advanced/Show assets from Editor directories" menu in the toolbar inside Texture Overview.

Version 4.2

- Unity 2017.2 beta 4 compatibility
- Added „Resize Algorithm" column (available in 2017.2 and newer only)

Version 4.1

- Checked Unity 2017 beta 10 compatibility
- Fixed: Issue detection incorrectly reported that DXT compressed non-power-of-two images, without mipmaps, can't be compressed.
- Fixed: Issue detection incorrectly reported that PVRTC images must be multiple of 4 (it's caught by the power-of-two requirement anyway).
- Fixed: Issue detection didn't warn „Only POT textures can be compressed if mip.maps are enabled" if the image has a sprite packing tag.
- Fixed: Issue detection didn't warn „Only POT textures can be compressed if mip.maps are enabled" when using ASTC compression, with mip-maps and the image is non-power-of-two.

Version 4.0

- Unity 2017 beta 1 compatibility
- Unity 5.6: Enabled Tizen platform button in toolbar (earlier disabled, because of this issue: https://issuetracker.unity3d.com/issues/crash-getautomaticformat-tizen-causes-textureimporter-formatandusagefromparameters-on-certain-texture-formats)

Version 3.9

- Unity 5.6.0b9 compatibility.
- Fixed negative size for compressed textures on 3DS platform using Unity 5.2
- Added „Search in Sprite Packing Tag" option (click magnifying glass icon in search box)

Version 3.8

- Unity 5.5 compatibility.
- Unity 5.6.0b3 compatibility.
- Added platform buttons to the toolbar, to easily switch between different platform

texture settings.
- Added many more columns to Texture Overview. Right-click a column header to select which ones to display.
- Added „Issues" detection. If, for example, a texture cannot be compressed because its size isn't a multiple of 4, Texture Overview is able to show you that. This is a very powerful feature to easily find issues between different platforms.

Version 3.7

- Fixed following warning that occurred using Unity 5.4 beta: Could not find method 'TextureUtil.GetGLWidth(Texture)'
- Fixed following warning that occurred using Unity 5.4 beta: Could not find method 'TextureUtil.GetGLHeight(Texture)'
- Fixed following warning that occurred using Unity 5.4 beta: Could not find method 'TextureUtil.CountMipmaps'
- Fixed error when canceling a progressbar (seems the error occurred under Windows 10 only?)

Version 3.6

- Scene mode now also finds textures used by inactive GameObjects.
- Added [AssetBundle Manifest](#) mode. It allows you to display referenced textures in an asset bundle manifest file inside Texture Overview.
- Better error message when Texture Overview for Unity 4 is used in a project that was upgraded to Unity 5.

Version 3.5

- Fixed the following warning that occured when Texture Overview started „Could not find method 'UnityEditor.BuildPipeline.GetBuildTargetGroup(BuildTarget)'" (Unity 5.1 beta).
- Fixed memory calculation for DXT1_Crunched and DXT5_Crunched compressed texture formats (Unity 5.1 beta).

Version 3.4

- Fixed memory calculation for ASTC compressed textures.

Version 3.3

- Added detection/handling of .exr and .hdr assets.
- Added „Tools > Advanced" option, see [Handle Rgb24 as Rgba32 for graphics memory calculus](#) documentation.
- Added [Graphics Memory column](#). Displays how many space the texture costs during runtime in graphics memory.
- Added [Main Memory column](#). Displays how many space the texture costs during runtime in main memory.

- Added Graphics and Main Memory to ["Memory Usage" overlay](#).
- Fixed Runtime and Main Memory calculation for ARGB4444, RGB565 and RGBA4444 texture formats.
- Fixed installation window somtimes displaying an error for a second, but then complete installation successfully.
- Fixed [Memory Usage](#) chart colors being all white when player settings are set to Colorspace=Linear.

Version 3.2

- Texture Overview Pro: Fixed error that occured when compiling the provided source code found in TextureOverviewSource.zip using Unity 5 DLL's.
- Texture Overview Pro: Updated readme.txt in TextureOverviewSource.zip that explains how to build Texture Overview.
- Added an „installation window" that appears when open Texture Overview after having imported a new version from the Asset Store. The installation window is repsonsible to install the proper plugin for your Unity version.
- Renamed column „Sprite Pixels to Units" to „Sprite Pixels per Unit" as it's found in Unity's documentation.

Version 3.1

- Fix: Fixed the following warning, which started to appear using Unity 5 beta 17: "Invalid AssetDatabase path: <FullPath>/UnityEditor.UI.dll. Use path relative to the project folder."

Version 3.0

- Texture Overview 3.0 requires Unity 4.3 or newer, because it now also displays „Sprite" related settings.
- New: Added „Sprite Mode" column.
- New: Added „Sprite Packing Tag" column.
- New: Added „Sprite Pixel per Unit" column.
- New: Added an error dialog when you try to open Texture Overview 3.0 using an older version than Unity 4.3.
- Change: Moved .cache file from plugin installation directory to project „Library" directory. Texture Overview 3.0 will automatically move this file from the old to the new location.
- Fix: Fixed rare issue where the .cache file wasn't updated after a texture import.

Version 2.9

- Fix: The plugin displayed the following error using Unity 4.6 open beta, which caused the plugin to start much slower than usual: "Library/assetDatabase3 not supported, seems not to be an assetDatabase3 file."

Version 2.8

- New: Added „Find Prefabs and Scenes in Project" function to context menu.
- New: Added „Find Usage Results" window, which displays the result of „Find Materials in Project" and „Find Prefabs in Project" search operations.
- Fix: „Find Prefabs in Project" caused an out-of-memory editor crash in some/big projects.

Version 2.7

- New: Added Memory Usage overlay window. You can open it from the „Tools" popup menu. It displays how much memory certain texture types are using.

- Fix: OSX Editor crash when build target is set to iOS and the project contains thousands of textures.

- Fix: „RuntimeSize" was doubling the amount of memory when the texture is set to readable always, but it should do this only for Uncompressed, DTX and Alpha8 formats. (correct value is applied the next to you import that texture).

- Fix: „NullReferenceException at TextureOverview.MainWindow.OnSelectionChange" occasionally occured when pressing Play in the editor.

- Fix: OSX displays a resize grabber in the lower right corner of the Textur Overview window, which was overlapping the „Info" button. Added some space to prevent that overlap.

- Fix: „Name" and „Path" columns were using CurrentCultureIgnoreCase for comparison, but it's recommended to compare path names using OrdinalIgnoreCase. This fix also makes sorting by „Name" or „Path" significantly faster when the project contains 10000 and more textures.

Version 2.6

- New: Texture Overview Pro now contains the full C# source code to build Texture Overview.

- Fix: „Export to CSV" generated broken CSV when a comma was in the asset path.

- Change: I replaced 3rd party icons with selfmade ones to not cause any license issues, now that the source code is available and therefore the icons as well.

Version 2.5

- Fix: „Runtime Size" reported negative numbers for textures in ETC2 formats.
- Fix: „Format Short" for ETC2 formats showed a number rather than the actual texture format as short string.

- New: Last column sorting information now gets restored when the plugin starts.

Version 2.4

- Fix: Plugin starts much quicker now, see „Cache File (the technical part)" section in this document. Only the first start might take a while to complete.
- New: Added "Runtime Size" column, which displays how many space textures consume during runtime in memory. The "Read/Write Enabled" setting is respected in the size calculation.
- New: Added "Compression Quality" column, which displays the Quality of texture compression used during import in the editor (iOS/Android).
- New: Added "Bypass sRGB Sampling" column, which displays whether the texture will be converted from gamma to linear space when sampled.
- New: Added "Scene" to popup menu in top toolbar, which displays textures used in the opened scene.
- Fix: The „Playmode tint" setting in Unity Preferences is now correctly handled in Texture Overview.
- Fix: „Filter by Type" wasn't able to correctly filter textures with „Texture Type = Advanced" and „Generate Cubemap" to something other than „None".
- Fix: The „Color Space" player-setting no longer impacts the icons in Texture Overview.
- Fix: Sometimes a „Getting control 0's position in a group with only 0 controls when doing a repaint" occured when the plugin automatically opens with Unity and the Unity „New Version" dialog appeared while Texture Overview was busy loading texture settings.
- New: Pressing a key, while the list has the focus, jumps to the item of the texture that starts with the key.
- New: Added key shortcuts for a few context-menu functions (Open, Select in Project, Delete)
- New: Added „Export CSV" to „Tools" popup-menu. The exported CSV (comma separated) file can be used to generate fancy graphs in spreadsheet applications to visualize memory usage for example.
- New: Pressing „Enter" now assigns the selected textures to the Unity texture inspector.
- New: Added links to Texture Overview asset store and forum pages in About Dialog.

Version 2.3

- New: The search field accepts drag&drop from the Unity project window. This makes it much easier to filter textures by path, no typing needed anymore.
- New: The search supports basic search operators now: „add", „or" and „not"
- Fix: The search mode (Search by Name, Path) has been lost every time Texture Overview has been closed.

- Change: Increased the width of the search field.
- Change: Renamed „Filter by Name" to „Search in Name", etc

Version 2.2

- Fix: Lots of memory optimizations done. Texture Overview needs to load textures into memory, then read their properties to display them. This caused a „Out of Memory" fatal error when several thousand textures are in the project and memory consumption went near 4gb. Now, textures get properly unloaded once their properties have been read and thus are no longer required.
- Fix: If a texture couldn't be loaded, Texture Overview occasionally displayed a „null-reference" error-dialog.
- Fix: Texture Overview got slower over time, the longer it was used the slower it got. This was noticable when having 4000 textures in Texture Overview, working with it a couple of minutes, then using „Ctrl+A" to select all items. This took a couple seconds to complete. Now it's super fast again.

Version 2.1

- Fix: Some textures display "???" as texture type.
- Fix: Added missing types to the texture type filter.
- Fix: Context menu item "Delete" didn't work when the Unity project pane was using "Two Column Layout" and a folder was selected. In this case "Texture Overview" asked to delete the folder, rather than the textures selected in "Texture Overview".
- Fix: Context menu item "Show in Explorer" is now shown as "Reveal in Finder" when running Unity editor on MacOS.
- Fix: Text inside the list was sometimes cropped vertically.
- Fix: Inactive selection color when using pro skin (dark theme) didn't really match the color used in Unity itself for an inactive selection.

Version 2.0

- Initial public release