

# Texture array inspector

---

Advanced inspector for the texture array (Texture2DArray) object. This tool allows to:

- View and to change the size, format, compression, color space and other texture array properties;
- Add, remove and change texture array layers order;
- Assign a common texture (called 'source texture') as a texture array layer;
- Extract and save a single texture array layer as common texture (PNG file).

Each time the assigned source texture is changed it will automatically update the texture array, replacing the layer it's assigned to.

Source textures are editor-only dependencies. Texture Array Inspector will not make them to be included in build (unless they are used somewhere else in a project).

This tool allows to enable or disable texture array read/write mode as well. Note that both texture array and source textures does not have to be readable to be used with this tool.

Shipped with a TextureArrayTool class, that works similarly to [ArrayUtility](#) except it works with texture arrays instead of the common arrays.

## Quick Start

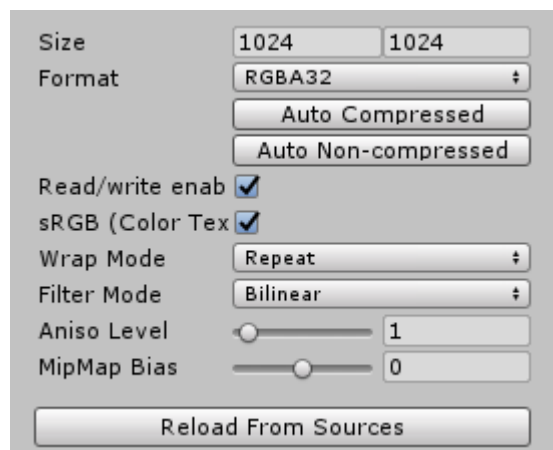
---

Create a Texture2DArray by right-clicking in the Project window, Create -> Texture Array.

Select a created (or any existing) Texture2DArray asset, and feel free to modify texture array and texture layers properties.

## Settings

---



- **Size:** width and height of the texture array (in pixels).
- **Format:** number of channels, channels order and compression type. Useful link: [compression formats](#),
- **Auto Compressed / Auto Non-Compressed:** will automatically select a proper texture format using current build platform and texture type. Texture type is selected from the popup window after pressing the button:
  - RGBA: standard color texture with transparency channel;
  - RGB (No alpha): color texture with no transparency channel;

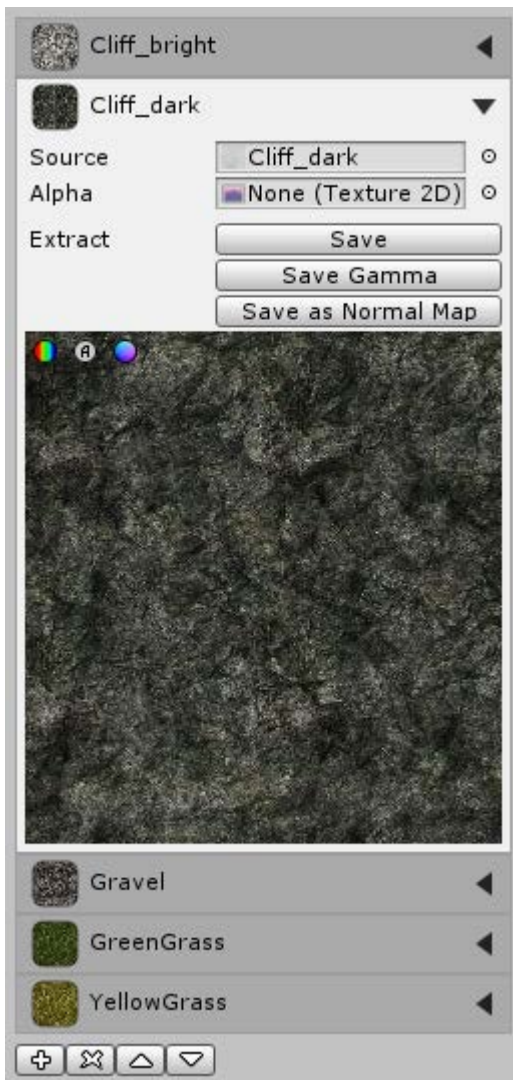
Normal Map: 3-channel tangent-space normal map

- Monochrome: single channel image
- Monochrome 32 bit float: high precision single channel image. In most cases it's a heightmap.
- **Read/Write Enabled:** Check this box to enable access to the Texture data from script functions (Texture2DArray.SetPixels and Texture2DArray.GetPixels). Note that a copy of the Texture data is made, doubling the amount of memory required for Texture Assets, so only use this property if absolutely necessary. This is only valid for uncompressed and DXT compressed Textures; other types of compressed textures cannot be read from.
- **sRGB (Color Texture):** Check this box to specify that the Texture is stored in gamma space. This should always be checked for non-HDR color Textures (such as Albedo and Specular Color). If the Texture stores information that has a specific meaning, and you need the exact values in the Shader (for example, the smoothness or the metalness), uncheck this box. This box is checked by default.
- **Wrap Mode:** how the Texture behaves when tiled
  - Repeat: the Texture repeats itself in tiles.
  - Clamp: the Texture's edges are stretched
  - Mirror: every 2nd texture tile is mirrored
  - Mirror once: similar to clamp, except it produces 2\*2 mirrored tiles grid
- **Filter Mode:** Select how the Texture is filtered when it gets stretched by 3D transformations:
  - Point: the Texture appears blocky up close,
  - Bilinear: the Texture appears blurry up close,
  - Trilinear: like Bilinear, but the Texture also blurs between the different MIP levels
- **Aniso Level:** Increases Texture quality when viewing the Texture at a steep angle. Good for floor and ground Textures. The value range of this variable goes from 1 to 9, where 1 equals no filtering applied and 9 equals full filtering applied. As the value gets bigger, the texture is clearer at shallow angles. Lower values mean the texture will be more blurry at shallow angles. See Unity documentation on [Importing Textures](#) for more information on Anisotropic filtering.
- **MipMap Bias:** Mip map bias of the texture. A positive bias makes a texture appear extra blurry, while a negative bias sharpens the texture. Note that using large negative bias can reduce performance, so it's not recommended to use more than -0.5 negative bias. In most cases better sharpening of the texture can be achieved by using anisotropic filtering.

## Layers

---

All of the texture layers are listed in a layers tab:



Each layer could be selected by clicking on it. Below the layers, next to the "Layers:" label you can see layer control buttons:

- Add: will add new layer under the selected one.
- Remove: removes currently selected layer.
- Up: will change the layers order by moving the selected layer up.
- Down: will change the layers order by moving the selected layer down.

The layer properties could be expanded by clicking on chevron button ( / ) to the right of layer name.

- **Source:** texture that fills an appropriate texture array layer. Note that it's an editor-only slot, source texture is not included in build unless it's used anywhere else in a project. After any change in source texture this layer will be re-filled with a changed texture.
- **Alpha:** same as source texture, but sets texture array alpha-channel. If no alpha defined, source texture sets all 4 channels (RGB and A) of texture array. If alpha defined source sets only the RGB channels, and Alpha is filled using desaturated 'alpha' RGB.
- **Extract:** saves currently selected layer as a PNG file
  - Save: saves texture pixels exactly as in the texture array
  - Save Gamma: applies gamma correction before save. For linear arrays.
  - Save as Normal: for 2-channel normal maps. Restores the 3rd (blue) channel before save.
- **Preview:**
  - : view color (RGB) channels of the texture
  - : view texture alpha channel

- : view 2-channel normal map as a standard 3-channel tangent space normal map

## TextureArrayTools API

---

TextureArrayTools extensions class adds an API that makes Texture2DArray work similarly to list or [ArrayUtility](#) except it works with texture array layers. This way you can treat texture arrays as a common arrays, and add, remove, switch individual layers as Texture2D objects.

TextureArrayTools namespace is `PluginTools`.

The following are the extension methods, so it's possible to use them both ways: `myTextureArray.GetTexture(i)` and `TextureArrayTools.GetTexture(myTextureArray, i)`:

- `Texture2D GetTexture(int num)` Returns a texture array num layer as a Texture2D
- `void SetTexture(Texture2D tex, int num)` Sets a texture as texture array layer. The texture does not have to be the same size and format as a texture array - it will be resized and formatted automatically. It does not have to be readable too.
- `void SetTextureAlpha(Texture2D tex, Texture2D alpha, int num)` Sets RGB to src RGB, and A to desaturated alpha RGB.
- `Texture2D[] GetTextures()` Returns all of the layers as a common array of textures.
- `void FillTexture (Texture2D dst, int num)` Fills dst texture with a num layer. Similar to GetTexture, but the dst texture could be created ahead.

The following methods work similarly to ArrayUtility and could be used with a ref keyword: `TextureArrayTools.Add(ref myTexArr, tex)`. Non-ref call is also possible, in this case it will return the new array: `Texture2DArray dstArr = TextureArrayTools.Add(srcArr, tex)`:

- `void Add (ref Texture2DArray texArr, Texture2D tex)` Appends texture to the end of texture array.
- `void Insert (ref Texture2DArray texArr, int pos, Texture2D tex)` Inserts tex texture at num position index.
- `void RemoveAt (ref Texture2DArray texArr, int num)` Remove element at num position index.
- `void ChangeCount (ref Texture2DArray texArr, int newSize)` Changes the length of texture array.

The methods create a texture array clone:

- `Texture2DArray ResizedClone (this Texture2DArray texArr, int newWidth, int newHeight)` Changes the cloned texture array width and height.
- `Texture2DArray FormattedClone (this Texture2DArray texArr, TextureFormat format)` Changes the cloned texture array texture format.
- `Texture2DArray LinearClone (this Texture2DArray texArr, bool linear)` Changes the color space (gamma (linear=false) or linear (linear=true)).
- `Texture2DArray WritableClone (this Texture2DArray texArr)` Enables texture read/write in clone.