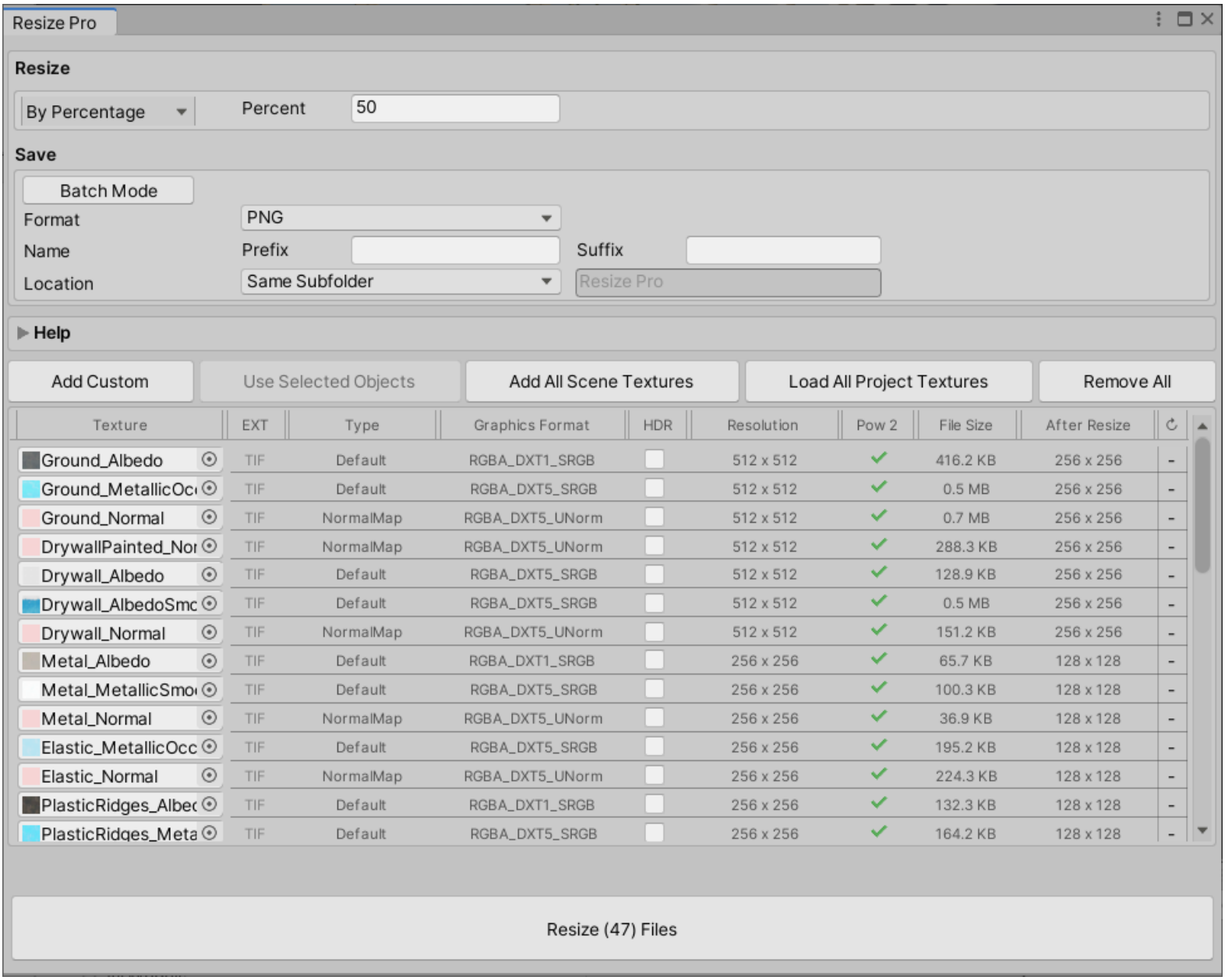
Resize Pro

amazingassets.world

1. EDITOR

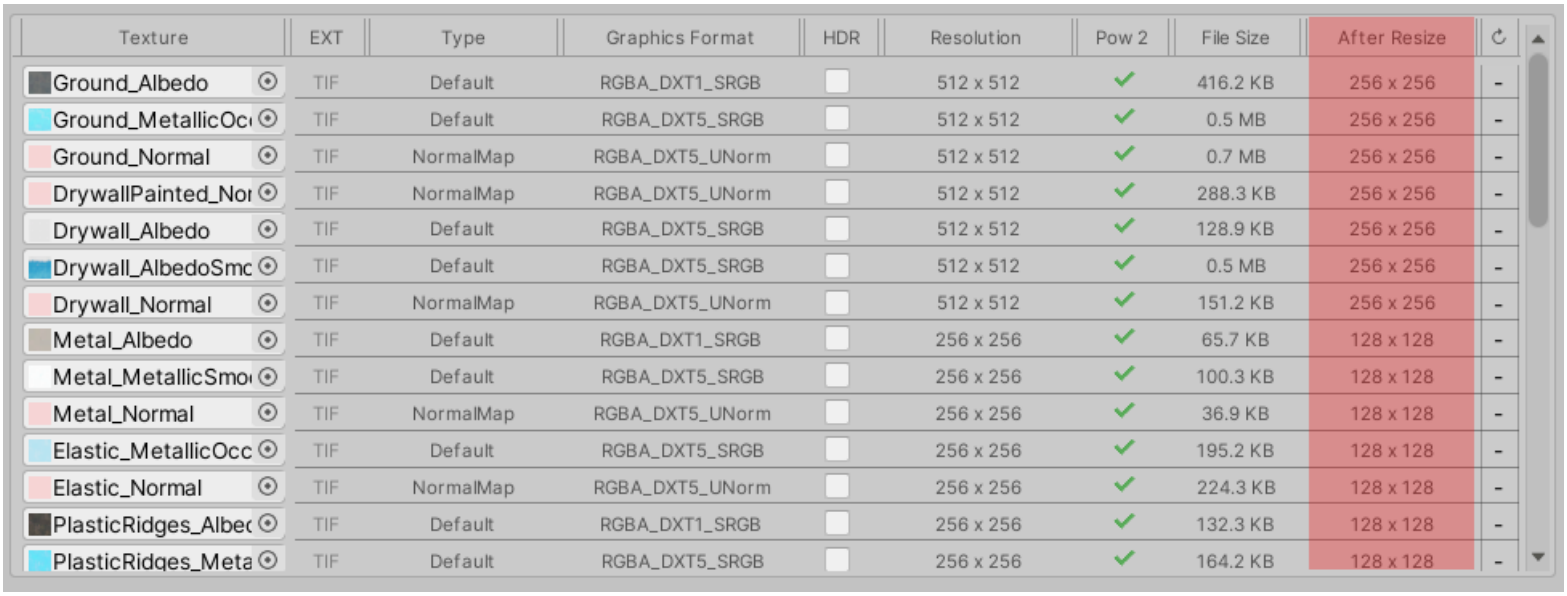
Resize Pro editor window can be opened from **Unity Main Manu/Window/Amazing Assets/Resize Pro**.



Textures for resizing can be added to the window using tool buttons or by drag & drop texture assets from Project window, same as materials and entire folders.

Note, all textures displayed inside window are loaded into the memory. If it is required to resize big count of textures, better use Batch mode.

Resize By - Allows resizing textures by providing exact width and height in pixels, by percentage or by choosing max resolution.
Editor window displays various texture information, including its size after resize.



Texture	EXT	Type	Graphics Format	HDR	Resolution	Pow 2	File Size	After Resize
Ground_Albedo	TIF	Default	RGBA_DXT1_SRGB	☐	512 x 512	✓	416.2 KB	256 x 256
Ground_MetallicOc	TIF	Default	RGBA_DXT5_SRGB	☐	512 x 512	✓	0.5 MB	256 x 256
Ground_Normal	TIF	NormalMap	RGBA_DXT5_UNorm	☐	512 x 512	✓	0.7 MB	256 x 256
DrywallPainted_Nor	TIF	NormalMap	RGBA_DXT5_UNorm	☐	512 x 512	✓	288.3 KB	256 x 256
Drywall_Albedo	TIF	Default	RGBA_DXT5_SRGB	☐	512 x 512	✓	128.9 KB	256 x 256
Drywall_AlbedoSmc	TIF	Default	RGBA_DXT5_SRGB	☐	512 x 512	✓	0.5 MB	256 x 256
Drywall_Normal	TIF	NormalMap	RGBA_DXT5_UNorm	☐	512 x 512	✓	151.2 KB	256 x 256
Metal_Albedo	TIF	Default	RGBA_DXT1_SRGB	☐	256 x 256	✓	65.7 KB	128 x 128
Metal_MetallicSmo	TIF	Default	RGBA_DXT5_SRGB	☐	256 x 256	✓	100.3 KB	128 x 128
Metal_Normal	TIF	NormalMap	RGBA_DXT5_UNorm	☐	256 x 256	✓	36.9 KB	128 x 128
Elastic_MetallicOcc	TIF	Default	RGBA_DXT5_SRGB	☐	256 x 256	✓	195.2 KB	128 x 128
Elastic_Normal	TIF	NormalMap	RGBA_DXT5_UNorm	☐	256 x 256	✓	224.3 KB	128 x 128
PlasticRidges_Albe	TIF	Default	RGBA_DXT1_SRGB	☐	256 x 256	✓	132.3 KB	128 x 128
PlasticRidges_Meta	TIF	Default	RGBA_DXT5_SRGB	☐	256 x 256	✓	164.2 KB	128 x 128

Note, editor window displays resolution of Texture object imported into engine. ResizePro preforms resize based on the file original resolution and not Texture object.

Format – Resize Pro reads all Unity Texture2D formats, but editor tool can save new textures only in **JPG** (8bits per-pixel), **PNG**(8bits per-pixel), **TGA**(8bits per-pixel) and **EXR**(16 and 32bits per-pixel) formats.

Name Prefix / Suffix – Adds prefix/suffix to the generated file name.

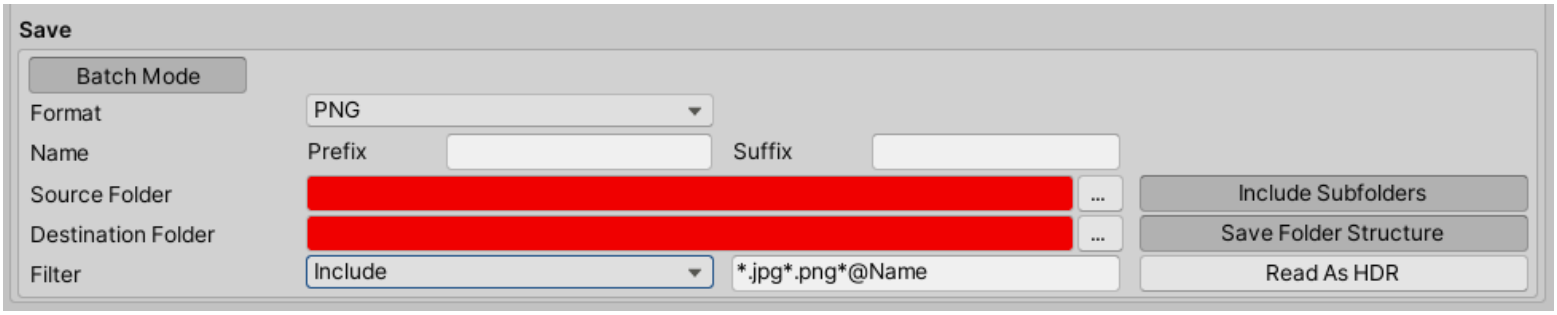
Save Location – Allows saving resized textures in:

- **Same Folder** – Saving resized textures in the same folder will overwrite original textures that have same extensions. All materials, scripts and other resources do not lose references and automatically use new textures.

  Note, inside editor window textures that will be overwritten with the new textures, will have yellow highlights.

- **Same Subfolder** – Resized textures are saved in the same folder as the original files but inside user defined subfolder. Original textures are not overwritten.
- **Custom folder** – New textures can be saved anywhere on the hard drive. Save location needs to have Read/Write permission.

Batch Mode – Enables textures resize using batch mode, without loading them into a memory.



Source Folder – Must be project relative path.

Include Subfolders – If enabled, Resize Pro will read all files from subfolders inside **Source Folder**.

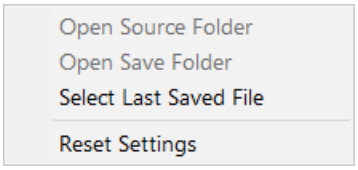
Destination Folder – Resized textures save location. Can be any folder on the hard drive with Read/Write permission.

Save Folder Structure – If enabled, resized texture will be saved inside **Destination Folder** with the same folder structure as they have in a project.
If disabled all textures are saved in the **Destination Folder**. Files with same name and extension may overwrite each other.

Filter – Filter for choosing which textures will be read and resized from **Source Folder**, by providing *.extensions* and *@names* separated by ***** symbol.

HDR – Enable only for resizing HDR textures.

Reseting editor window settings and/or selecting new resized textures can be done from the context menu.



2. RUN-TIME API

Resize Pro run-time API can be brought into scope with this using directive:

```
C#
using AmazingAssets.ResizePro;
```

Now Unity Texture2D class will have new **ResizePro** extension methods.

```
bool ResizePro(int width, int height, bool linear, bool hasMipMap, bool compress)
```

```
bool ResizePro(int width, int height, bool linear, bool hasMipMap, TextureFormat textureFormat)
```

Resizes the texture by ***width*** and ***height***.
linear – Used for resizing non-color textures (BumpMap, DisplaceMaps, etc).
hasMipMap – Generates mipmaps.
compress – After resizing texture always is in uncompressed format, use this option to compress it using Unity compress method.
textureFormat – Allows changing texture format after resizing. Only uncompressed formats are allowed.

Methods return *false* if texture resize fails. Texture in this case is not modified.

Note, above resize methods work only with textures that are not locked in a memory and have Read/Write flag enabled inside Texture Import Settings.

```
bool ResizePro(out Texture2D dstTexture, int width, int height ...)
```

Exactly same resize methods as above, except source texture is not modified and **`dstTexture`** contains new resized texture data.
Those methods read data from any texture, even if they have no Read/Write flag enabled and are locked in a memory.

```
bool IsTextureResizable ()
```

Returns *true* if a texture is readable and can be resized.