**COLLADA Exporter v1.3.0 in the Unity Asset Store**

**(3.x and 4.x, free and pro, windows and mac)**

With this Unity extension to the editor's 'file' menu, you can take your work from Unity to any COLLADA (.dae format) importing application, such as Apple's Preview or iBook, Google Earth, SketchUp,…), as well as most digital content creation tools (Photoshop, Autodesk Max, Maya, Blender, ...) , or other game engines (threejs, Turbulenz, Fl4re3d, bullet physics …), even back into Unity and of course your own applications.

This extension to the Unity file menu allows you to export COLLADA (.dae) files from Unity. It is very easy to set up and use.

**Changes in v1.3.0:**

Added as vertical scroller to the COLLADA export UI that became too big to fit on some screens. Also save and restore UI parameters as editor preferences, so options are restored when loading a project in Unity.

Added Vertex color to the export.

Added an option to inverse shininess to adjust between phone and Torrance-Sparrow model.

Added lightmap UV export options to enable lightmap can be seen without having to write a shader. Works great with Apple preview for example.

Added collider and rigid body export into COLLADA physics. This was tested with Bullet and with Turbulenz.

Misc: Fixed bugs in the image converter, detect 565 conversion problem and print warning message. Added option to export only 2UVsets. Fixed timing issues with animation duplication option. Fixed bug in instance material when material did not have a texture, Fixed cubemap init\_from name, added ‘COLLADA\_EXPORT’ to Log message coming from the COLLADA exporter. Fixed terrain exporter bug[index out of range]. Fixed skinning export bug: duplicate controllers if instanced several times

**Changes in v1.2.0:**

Added ‘bump and cube map’ export. Both the images and material information are exported.

Improved common profile material export

Added support for shader parameters. All the parameters that are visible in Unity editor are exported with a <newparam> element in the <effect>.

Fixed some issues with png conversion, in particular 565 format conversion.

Added export of ‘JOINT’ nodes for proper skin/bone import in tools. Fixed <skeleton> element and <instance\_controller> placement in the scene.

Detect and suppresses negative scaling in animations.

Checked the plug-in works in both Unity 3.x and 4.x. Note that the plug-in does not export any features specifics to 4.0. In particular, the Mecanim characters animations will not be exported, only the ‘legacy’ animations will be exported from 3.x or 4.x

 + minor fixes/typos

**Changes in v1.1.1:**

Added 'animation clips' option

Added 'fix names to valid'

 + minor fixes/typos

**Changes in v1.1:**

Added Terrain export option, including trees.

Cleaned up texture PNG export options.

Added ‘convert names to XML’ option.

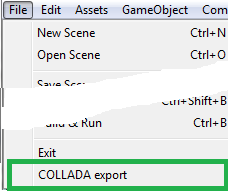
+ minor fixes/typos

**Questions?** Send email to rita@fl4re.com for more info

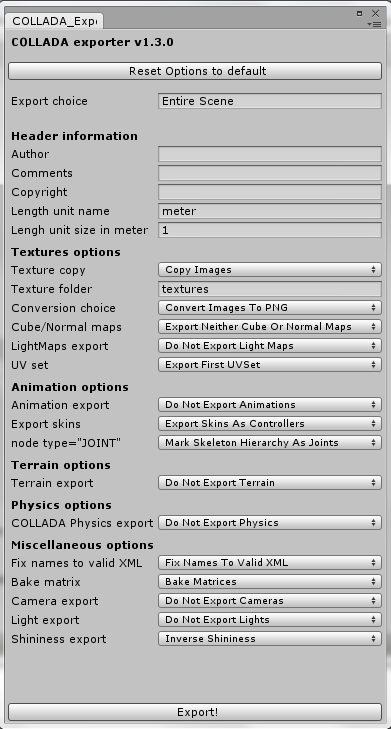
**Set up:**

The COLLADA export should appear in the ‘File’ menu. If not, simply drag the downloaded .dll file and drop it into the ‘Editor’ section under the Unity *Project* window.

Clicking on the menu entry will give you the COLLADA exporter dialog box:



**From the top, here is the COLLADA exporter explained:**



**Reset Options to default**. (new in v1.3.0) COLLADA exporter options are saved as editor preferences and restored with the project. This button will reset all the options to their default values, as depicted in the image above.

**Export choice** may be either “*Entire Scene*” (the default when nothing selected by the user), or you may select one or several objects to export by choosing “*Selection Only*”. Selection can be done in the scene or the project.

**Header information:**

The first set of options is the Header information, where you can fill in the names of the *Authors*, *Comments* and *Copyright information* – all these are optional. This comprises the header of the COLLADA header <asset> information.

Then you have the *Length unity name* and *Length unit size* in meters which defaults to “meter” and “1” meter is the default. If you use inches for example, you can enter “inch” in the unit name box and “0.0254” in the unit size box. No actual conversion is performed by the exporter, the application importing the COLLADA document uses that information to conform all imported files to the same dimensions.

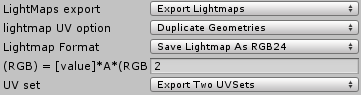
**Textures options:**

*Texture Copy* - If you select “Copy Images”, all the images used by the Asset exported will be stored in sub-directories provided in the *Texture folder*. The default name for the texture folder is “textures” but you can change that to whatever you want.

The COLLADA document (.dae) will reference the image file that was copied or converted. If *texture copy* is set to “Do Not Copy Images”, the images won’t be copied but the *Conversion choice* parameter will still be used to create the corresponding file name and extension in the COLLADA document. This option is very useful when frequently exporting your scene is necessary, but you have not changed your images. It’s also useful if you modify the images outside of Unity and you do not want the exporter to delete your changes.

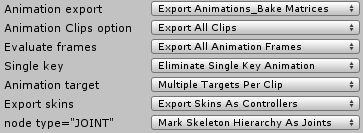
The default *Conversion choice* for images is to “Convert All Images To PNG”. The exporter will create either RGB (24) or RGBA(32) textures according to the internal format in Unity. Another option is “Convert ImagesTo PNG 24*”*. This will convert all images to RBG (24) .png, thus removing the Alpha/transparency information, this is useful when the alpha channel is not used for transparency and would carry-out undesired transparency information. The last option is “Do Not Convert Images”. In that case, the original images will be copied into the *Texture folder* directly. The issue with this option is that most COLLADA importers may not recognize formats such as .psd.

Next choice is *include bump mapping and/or Cube Mapping* in the export. Cube maps will be exported as 6 independent .png(rbg) images, but they are not referenced by the material. Bumpmaps are referenced in the material using the <extra> profile=”FCOLLADA” <bump>.

(new in 1.3.0) when selected, *LightMaps export* will copy the images used for lightmapping into the texture folder. In addition it provides two options to reference the lightmap in the exported COLLADA document, and two options for the image format.

In order to enable the use of lightmaps without a shader, the exporter includes a reference to the lightmap in the material <emission>, and pre-calculate the lightmap UV on all vertices, stored in the second set of UV. The problem is when using the same object (prefab) at multiple places, different set of UVs need to be calculated to map into the correct lightmap. The first option is to duplicate the entire geometry of such objects. This works well with all COLLADA importers, but create a large file. The other option is to create one extra set of UV in the same object, and select the correct one when instancing in the COLLADA <scene>. This create files that are a lot smaller, but unfortunately a lot of importers do not correctly import this. The *Lightmap format* option can be either to export the lightmap data as-is (RGBA), which requires per-pixel blending to be seen correctly, or to pre-calculate a RGB only ‘glow’ map, with the multiplication coefficient provided as an exporter option. For example, option of 2 works fine with Apple Preview, option of 4 looks good with Turbulenz viewer.

The last option is UV set that defaults to “Export First UV Set” - which is the texture mapping used for diffuse textures. When exporting lightmaps, two UV sets will be exported at minimum. Total of three UV sets are possible to export.

**Animations options:** The first choice is to export or not export any animations. If you export Animations, the exporter will automatically “Bake Matrices” in the scene and export animations as 4x4 matrices. When selected additional options are presented:

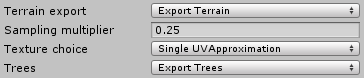
The *Animation Clips Options* can be set to 'export all clips' or Export single clip. If 'export single clip' is selected an additional parameter appears 'name of clip to export'. The user has to type the name of the animation clip exactly as it appears in the unity animation clop selector to have only that animation selected. For example, a character that has a 'idle' , 'run', 'walk' animation clips can be exported with just the 'run' animation. This options is there because several tools and COLLADA importer do not support the animation\_clip information. If 'export all clips' is selected, all the animation curves will be stored at successive times, and the correct animation\_clip information with begin and end time for each clip will be exporter. If the importer does not support the clip information, the different animations will simply appear to be playing continuously one after another.

The *Evaluate frames* default is “Export All Animation Frames” which will export one matrix per frame of animation. If you select “Export Animation Keys” only the matrices at keyframes will be exported. This option should be used only if software in which you want to import your COLLADA document is able to interpolate the values between the keyframes. The next option is called *Single key* that defaults to “Eliminate Single Key Animation.” This is included because Unity allows single key animation in clips which effectively freezes the animation to the value in that key.

The next option is *Animation target*. COLLADA allows for “Multiple Targets Per Clip” in order to keep the document small by sharing animation across multiple instanced objects (aka: prefabs). Unfortunately we found out that a some COLLADA importers do not support multiple targets per clip therefore one can select the option “Duplicate Animation Clips” which is a work-around but which bloats the size of the exported document. Note that “Multiple Targets Per Clip” works fine with Apple Preview in Mountain Lion (10.8).

The last Animation option is *Export skins* and the default value is “Export Skins As Controllers” which is how skins and bones are expressed in COLLADA. If the animation export is selected, the animation will be applied to the bones through the controllers. Another option here is to “Export Skins as Geometry” which will convert the skins and bones to basic geometry (without the bone animation). This is useful when you want to export the model into an application that does not recognize skinning. The last choice is “Not Export Skins” which will completely ignore skinned objects in the export.

The *node type=”JOINT”* option lets you decide if the hierarchy nodes used for bones should be marked as “JOINT” in the COLLADA scene. It is recommended to leave this option in its default ‘Mark Skeleton Hierarchy As Joints’ as this is what most COLLADA importers require.

**Terrain options:** The COLLADA exporter enables export of terrain and trees, but it will only export a fixed geometry, and not the dynamic terrain used by Unity. If enabled, additional options are available:

In Unity a terrain is a height map, associated with a Splat Map that contains up to4 values (stored in a RGBA image) that are interpreted by the algorithm to blend between different materials (splat prototypes). The COLLADA exporter creates a triangle mesh model of the terrain geometry with a density defined by the *Sampling Multiplier*. Two triangles are created per quads, which corners correspond to each value in the HeightMap. So a 100x100 HeightMap will define 99\*99 quads, resulting in 19,602 triangles. Using a multiplier of 0.25, the number of triangles will be INT(99\*0.25) \* INT(99\*0.25) \* 2 = 1728 triangles.

The options for the diffuse material on export are defined by the *Texture choice* parameter. The first *Texture choice* option is to not export any texture.

“Single UV Approximation” is the default choice that assigns each triangle with the most important (greater coefficient in the Splat Map) material at its barycenter. This provides a coarse visual approximation of the terrain as rendered in Unity. A limitation is that the exporter only create one UV set that is scaled by the ‘size’ of the first splat prototype. If different sizes are used in the terrain the textures other than the first one will have incorrect scale. Since it is only an approximation of the visual terrain it may be sufficient anyway. The more triangles are generated (the density factor can be modified in the exporter) the better the approximation. But this will generate a lot of triangles, and there still would be only one material per triangle - no blending.

Another choice is to export the SplatMap that covers the terrain. This is probably the most valuable option as this enables the user to use or modify the blending coefficients. The prototype textures will also be exported, but won’t be attached to the terrain. In order to reproduce the visual aspect, instead of seeing the splatmap values, it is necessary to provide a special shader that look-up the value of the splatmap at each pixel, and use the R,G,B,A values as 4 coeficients (between 0 and 1) that are then used to blend all 4 materials assigned to the terrain. It may be possible to create such complex material in modeling tools, please share with us if you know how. However there is another possibility - requiring some manual work: In unity display the terrain only (no trees, no objects only one ambient light) flat full screen and take a snapshot. Export the scene with the splatmap option. In the modeler, replace the splatmap texture with the image that was manually copied from Unity.

(new in 1.3.0) **Physics options:** This is the first instalment of the COLLADA exporter that includes COLLADA Physics scene. Only two physics objects are exporter in this version: colliders and rigid bodies. The Colliders will export the corresponding Physics <shape>, including compound colliders as a hierarchy of shapes. The rigid bodies are exported in one physics\_model, and instanced in the physics scene. When physics export is selected an additional option is available:

In a physics simulation rigid bodies are moved directly by the physics engine. In COLLADA rigid bodies are attached to position in the scene, and the importer should calculate the absolute position from the scene hierarchy and create the rigid bodies at that position. But in our testing, we found out that the Bullet viewer as well as the Turbulenz importer are not doing this operation and get confused when rigid bodies are not attached directly at the root of the scene. The option “Move Rigidbody Target To Scene Root” create a <node> at the root of the scene with the position calculated from where it is attached in the scene.

Note also that some physics default importer – such as Bullet viewer - do not know now to decompose a transform matrix, in that case it is recommended to export without the animations and select “Separate Rotation Translation and Scale” in the option below:

**Miscellaneous options:**

The first option enable the exporter to save the names as-is in the COLLADA document. By default the names are converted to be “Valid XML” name strings so the exported document will validate against the COLLADA schema. This a restriction of COLLADA 1.4.1 schema which has been lifted in 1.5, but some applications will not be able to load a COLLADA document which names are not correctly XML encoded.

The next option is only available if “Do Not Export Animation” was selected. It allows the option to “Separate Rotation Translation And Scale” in the COLLADA scene.

The second option is to export or not the cameras. This is especially useful for tools such as Preview to enable the end user to have preset views.

The last option is *Light export*, which allows for exporting (or not) the lights sources. Most of the time, it is recommended not to export the lights as many applications can only handle a very limited number of lights; otherwise the scene may display completely black with too many lights! Given the nature of Unity models, it is generally preferable to use the LightMaps export option described above.

The last button is *Export!* Once you click on this, it allows you to select the filename and directory you want to export the COLLADA (.dae) document and texture folder. Be aware that by default Unity will select a directory in the Unity project folder – it is not recommended to save the COLLADA document and textures inside the Unity project folder.