COLLADA Exporter v1.6.2 in the Unity Asset Store
(4.5+ or 5.3+, 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, SceneKit, Autodesk Max, Maya, Blender, ...) , or the web (collada2gltf), or other game engines (threejs, Turbulenz, bullet physics ...), even back into Unity and of course your own applications[1].

Unity 5.3+: https://www.assetstore.unity3d.com/en/#!/content/40946
Unity 4.5+: https://www.assetstore.unity3d.com/en/#!/content/4031

Changes in v1.6.2:
- Rewrote XML load/save to avoid random fileIO system errors
- Replaced lightmapNear and Far with function that replace with new Dir and Light with Unity5.5+
- Fixed Morph export issues
- Fixed shader parsing issues, added try/catch for better stability

Changes in v1.6.1:
- Fixed crash when parsing new (5.3) shaders - added property.params from shader.

Changes in v1.6.0:
- Code refactor to **allow for batch scripting**. Example script included (api_cs.txt)
- Code refactor to allow dlls to be generated for **Unity4 or Unity5**. Unfortunately its not possible to generate universal dlls given Unity incompatibilities, and its is not possible to have a single package in the asset store that covers different versions of Unity, so we have to have two separate assets from this point.
- fixed crash when exporting build-in particle texture
- check if colision shapes are attached to the rigidbody, and reate new <rigid_body> otherwise
- added technique_common to all instance_rigid_body, even when empty to be 100% compatible with COLLADA 1.4.1 schema
- Added spread and pan (5.0+) for 3D audio extra
- Changed uuid.cs namespace to collada_exporter

---

[1] If you encounter a problem loading COLLADA files into Unity or other packages, please report those bugs directly to the software vendor providing the importer. This plug-in creates valid COLLADA documents, that Unity or other software should be able to import. Feel free to reach out to ritaturk@gmail.com for help.

Changes in v1.5.1:
- Fixed bug crashing exporter when prefab rendering does not have material attached
- Updated with Unity 1.5.5 built-in shaders

Changes in v1.5.0:
- Added support for blendshape (<morph>). Because of this addition, the COLLADA exporter is now only compatible with Unity 4.3+. Please contact us if you need an exporter for a previous version of Unity (without the blendshape support of course)
- Reverted broken UI logic for enabling lightmap export
- Fixed UI options for skin export. (using the NO option will export the geometry without the skin)
- Added TEXTANGENT and TEXBINORMAL export option
- Fixed yfov before xfov in camera, which created non-conformant files rejected by Apple preview
- Improved vertex color compatibility by moving vertex color out of the <vertex> element
- Added common profile material support for shininess and specular values
- Added option to not export <animation_clip> information, which for some reason Apple preview does not like. Without <animation_clip> Preview plays the animations.
- Fixed calculation for animation approximation optimization, slow but constant movement used to remove too many keys.

Changes in v1.4.14:
- Fixed logic to find correct skeleton for bone hierarchy.
- option to eliminate redundant matrices in animation export
- Fixed audio file url encoding
- UUID script is embedded into the COLLADA export dll
- Fixed terrain exporter, was missing last row/column

Changes in v1.4.13:
- Fixed issues in prefab exporter – transform was duplicated in scene and prefab file
- Fixed missing texture exports when exporting all scene with prefabs
- Added option to export locally modified prefabs to separate files
- Added options to set UUID to GameObjects
- fixed logic to mark node as JOINTS when <instance_controller> is using a controller already present.

Changes in v1.4.12:
- Added support for splitting prefabs out to separate DAE files

- Fixed URIs issue when no 'textures' folder is specified
- Added support for export animations as TRS values
- Removed third UV option export (Unity only has two)
- UI re-design
- Added conversion to 3 Euler angle rotate in TRS
- Fixed ToAxisAngle transformation
- Adjusted test for when not to export transform (smaller translations)
- Replaced all approximate comparison with Mathf.Approximately
- Unity's internal textures can now be converted and saved on export
- Ambient light color is now exported
- Presets
  - Switched settings over to XML, so that per-project settings can be saved
- (extra) particles
- (extra) audio

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 ritaturk@gmail.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.

Unity4 users -> ready to use after installation. You can delete the unity5.zip file if you want
Unity5 users -> unzip the content of unit5.zip and replace the two dlls. Do not run the automatic unity script conversion, it won't work. If you did, it is not a problem, just replace the dlls with the content of unity5.zip regardless.

Clicking on the menu entry will give you the COLLADA exporter dialog box:
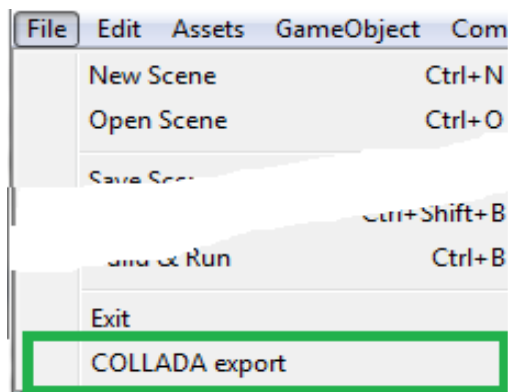
From the top, here is the COLLADA exporter explained:

COLLADA Exporter 1.5.0

Export choice	Entire Scene

Header information

Author	
Comments	
Copyright	
Length unit name	meter
Length unit size in meters	1

Textures options

Copy textures	☑ Folder: textures
Conversion choice	Convert Images To PNG
Export cubemaps	☐
Export normalmaps	☐
Export lightmaps	☐
UV set	Export First UVSet

Animation options

Export animations	☐

Skin and Morph options

Export skins	☑
node type="JOINT"	Mark Skeleton Hierarchy As Joints
Export blendshapes	☐

Transform options

Bake transforms matrices	☑
Prune identity transforms	☐

Terrain options

Export terrain	☐

Physics options

Export colliders	☐
Export rigidbodies	☐

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:
Copy textures - If you select this option, then 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 *Copy textures* is unchecked, then 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
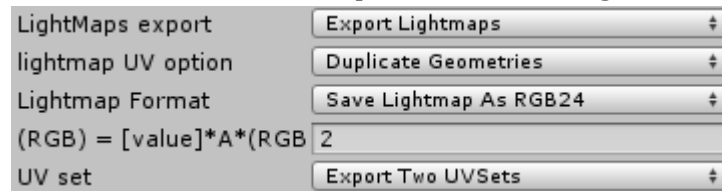
useful if you modify the images outside of Unity and you do not want the exporter to delete your changes.

The default *Conversion choice* for images is to "Convert 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 Images To 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.
Note that Unity has a lot of limitations on what texture formats can be converted back into png. For instance RGB565 will not export to png. Also, normal textures crested from psd will not be exporter correctly. The recommended way is to only use .png images (or any other desired format) when importing into Unity, and to use the 'do not convert' option in the exporter.

The next choices are to *Export cubemaps or normalmaps*. Cube maps will be exported as 6 independent .png (rgb) images, but they are not referenced by the material. Bumpmaps are referenced in the material using the <extra> profile="FCOLLADA" <bump>.

When checked, *Export lightmaps* 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.

LightMaps export	Export Lightmaps
lightmap UV option	Duplicate Geometries
Lightmap Format	Save Lightmap As RGB24
(RGB) = [value]*A*(RGB	2
UV set	Export Two UVSets

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.
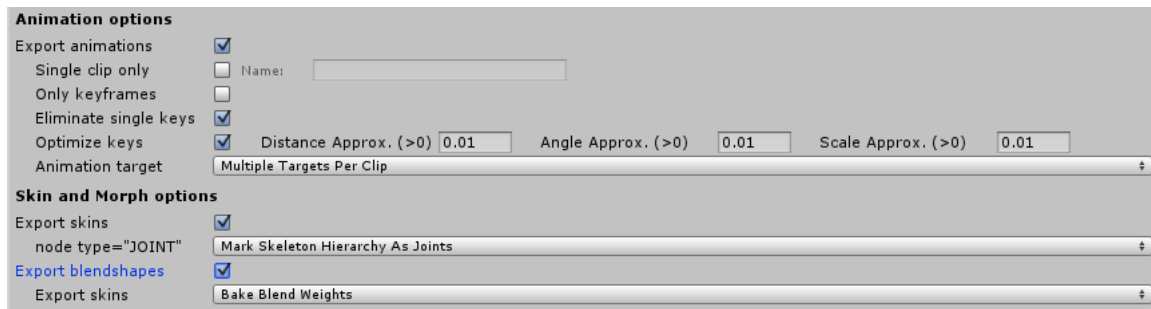
The last option, *UV set*, defaults to "Export First UV Set," which is the texture mapping used for diffuse textures. When exporting lightmaps, two UV sets will be exported at a minimum.

Animations options:
This option is to export 'Legacy Animations' from Unity. The new animation system is NOT supported. Only transforms animations are supported.

Animations are exported according to the *Transform Option* selection, which could be either MATRIX or TRS. In former versions of this exporter only matrices where supported. Note that TRS export for *Keyframes only* is not very well supported by importers/viewers. So it is recommended NOT to select *Keyframes Only* with TRS transform mode.

The first choice is to export or not export any animations. When selected additional options are presented:

If *Single clip only* is checked, then only animations with the name specified will be exported. For example, a character that has '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 *Single clip only* is not checked, then 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 exported. If the importer does not support the clip information, the different animations will simply appear to be playing continuously one after another.

The *Only keyframes* option will export animation for the existing keyframes only. This create smaller files, but the quality of the interpolation will not be perfect. It is a good idea to use keyframes only option, but to make sure that enough keyframes are defined in Unity. Otherwise, the animation will be sampled at constant time intervals and exported. This option should only be used if the software in which you want to import your COLLADA document is able to interpolate the values between the keyframes.

The next option enable exporting all the samples as matrices, but reduce the size of the exported file by eliminating duplicate transforms. When *Bake Transforms* is selected, an additional choice is provided in the animation export:
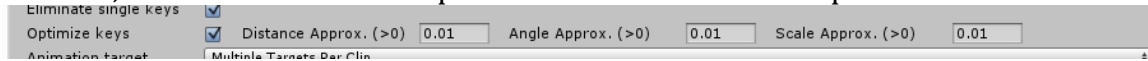
The first checkbox enable this option. The three values are then entered to select at which distance, scale, and angle difference two transforms should be considered identical, and thus the key eliminated from the output. For position and scale, the value is compared to the squared magnitude of the difference of both vectors. For angle, the difference in angle between the two quaternion is compared to the value.

The next option, *Eliminate single keys,* 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 next option is *Export skins.* This option has changed in version 1.5.0, it is now a simple check-box. If the animation export is selected, the animation will be applied to the bones through the controllers. If this option is not selected, the exporter 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 *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.

Transform options:
NEW! Morph (blendshape) has been added to the COLLADA exporter. When this option is selected the blendshapes will be exported in one of two ways. Selecting *Bake Blend Weights* will create one single geometry which position corresponds to the current blend weights. This is useful for creating collision volumes, but also to export geometries for importers that do not have support for COLLADA <morph>. The second choice is *Export Morph Target* will export all the blend shapers. If this option is not selected, the geometry exported will be the first blend shape target.

Transform options:
The *Bake transforms (matrices)* option allows individual translation, rotation, and

scale to be combined into a single matrix, for node transforms and animations. Without this option, both transforms and animation curves are exported in decomposed form (i.e. separate translation, rotation, and scale).
Note that it is recommended to use the Bake transform option, as we have noticed that not many importers can handle correctly the separate transforms.

Prune identity transforms allows the user to optimize the COLLADA document by removing extraneous <translation>, <rotate>, and <scale> elements to be removed where they are equivalent to the identity transform (i.e. zero translation, no rotation, unit scale). This option is not allowed if transforms are baked, since there is only a single <matrix> element. Also, this option is not allowed if animations are exported, since the nodes must exist in the .dae document in order to be animated.

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:
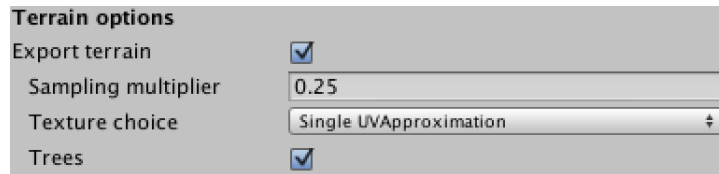
In Unity a terrain is a height map, associated with a Splat Map that contains up to4 values (stored in a RGBA image) that are interpreted by the algorithm to blend between different materials (splat prototypes). The COLLADA exporter creates a triangle mesh model of the terrain geometry with a density 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.

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.
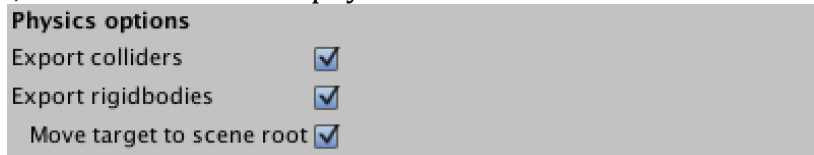
In a physics simulation rigid bodies are moved directly by the physics engine. In COLLADA rigid bodies are attached to position in the scene, and the importer should calculate the absolute position from the scene hierarchy and create the rigid bodies at that position. But in our testing, we found out 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 importers – such as Bullet viewer - do not know now to decompose a transform matrix, in that case it is recommended to export without *Bake transforms (matrices)*.

Miscellaneous options:
The first option, *Split prefabs into files*, enables the exporter to split out prefabs to separate files and store them in the directory specified (defaults to "prefabs"). Note that most viewers and importers are not compatible with this option. This option is mainly provided for those creating their own application loading COLLADA documents, or able to script importers to deal with multiple files and external references.
However this option is very useful for applications that would load the main scene first, and then page the various assets as needed. This is also very useful when dealing with a large scene, in an iterative process. Once the entire scene had been

exported with the option *Export Everything*, changes made to specific prefabs can be exported much faster by selecting the prefab, choosing the top choice as *Selection Only*, and selecting *Export Prefabs Only* option with the Split option selected. Another useful case for this option is when the size of the scene is close to the maximum size allowed in 32bits applications, and the exporter will crash the editor as it won't be able to allocate enough memory to export the entire scene at once.

Note that it is possible to just export the main scene, if prefabs have not been modified, but simply moved, removed or cloned in the scene. Select the *Entire Scene* to be exported, and then choose the option Export Main Scene Only with the Split option selected.

Unity allows many changes to be made to prefabs used in the scene. Those changes are applied by Unity when loading the prefabs in the scene, creating new transforms/game objects from the prefabs. When the scene or a transform in the scene is selected for export, the GameObjects that are in the scene contains the modified prefabs. But when the option to split prefabs in separate files is selected, the original prefab will be exported in the separate files – none of the changes done in the scene will be exported. When selecting *Do Not Split Prefabs If Locally Modified,* the exporter will only split the prefab in a separate file if there are local changes. If local changes are detected, the Transform hierarchy will be exported in the main scene file, as if the option to split was not enabled. When selecting *Split Prefabs Ignoring Local Modifications*, all the local modifications will be ignored and one file per prefab will be created. Check the output in the Console to see what local changes have been ignored. Note that some local changes are allowed, such as the root Transform, since this transform will be stored in the main scene file, and not in the prefab file.
 In order to export all the local modifications and still create separate file, the option *Bake Local Modifications And Duplicate If Needed* and the option *Bake Local Modifications And Always Duplicate* will create a separate file for each instanced prefabs, containing all the modifications. The difference between the two options if an instance does not make modifications to the prefab, only one prefab export will be created and referenced multiple times in the main scene.

If *Export cameras* is selected, then the cameras in Unity are exported. This is especially useful for tools such as Preview to enable the end user to have preset views.

If *Export lights* is selected, then light sources in unity will export. Most of the time, it is recommended <u>not to</u> 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 *Export lightmaps* option described above.
NEW! - <extra> information is added to both the <light> and the <instance_node> to export the shadow settings.

Export Audio enables the export of both the Audio scene elements as <extra> and the audio files into the specified folder ('audio' is the default name for this folder. Since this is purely an <extra>, audio information will be ignored by most applications, but this <extra> information is very useful for those creating their own COLLADA application.

Export Particles enables export of the Legacy particle system as <extra>. It does not support the new Shuriken system. Since this is purely an <extra>, particles information will be ignored by most applications, but this <extra> information is very useful for those creating their own COLLADA application.

Fix names to valid XML properly XML encodes names, so they validate against the COLLADA 1.4.1 schema. This restriction was lifted in 1.5, but some applications will not be able to load a COLLADA document. This option is on by default.

[NEW] Assign UUID to GameObjects is useful to make sure that COLLADA id are always the same between several exports of the same scene. Without this option the exporter uses the `GetInstanceID()` provided by Unity, but unfortunately this ID is not consistent and change every time the same scene/asset is loaded into the Unity Editor. So if you need to do some post processing or need to rely on the ID of COLLADA nodes, it is necessary to assign a unique ID to the Transforms. This option will test for the presence of a script that contains a UUID, and use it if it exists. Otherwise it will create one and add to the GameObject. However Unity does assign a UUID to assets that are represented by files, such as animations and textures. In that case the exported will use the existing UUID.

Important: This option does modify the GameObjects, so the scene will need to be saved after exporting, in order for the UUIDs to be saved in the scene. There will be an alert to tell when the scene is modified and will need to be saved.

Please note that if you manually copy objects in the scene in Unity, the copied object will have a copy of the UUID, and that is bad – as now you have duplicate IDs. If you copy/paste nodes inside Unity scene, remember to remove all UUIDs from the copied nodes immediately.
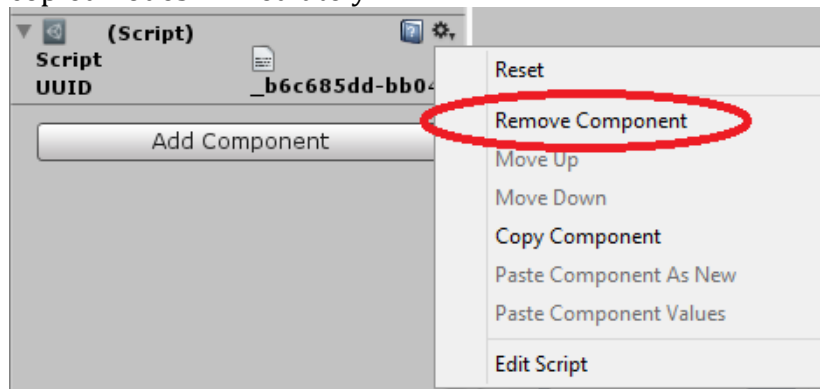
Also note that if the exported is asked to split Prefabs in separate files (but not to bake the modifications), the prefabs will not be able to receive a UUID, since they are fundamentally read-only. So best is to select the Bake Local Modifications options when splitting the export in multiple files, and needing consistent UUIDs.

Invert material shininess will export any shininess parameters in materials as 1 / the original shininess. Unfortunately some tools/viewers do not correctly interpret shininess, this option provides a workaround for this.

Preset
Presets allow for settings to be saved for various types of exports (e.g. environments vs. characters). Settings are automatically saved when the COLLADA export UI is closed and re-opened, but it may be useful to have several sets of settings available.
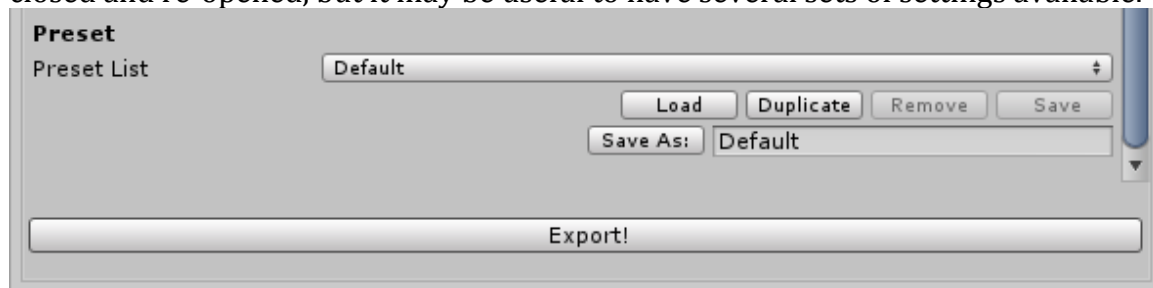
Simply select a preset from the list and press 'Load'. In order to create a new preset, simply type a name and press the 'Save As' button. A preset can be deleted using the 'Remove' button, but the Default present cannot be removed or modified. To update an existing preset, simply press the 'Save' button.

Note: presets are saved in the file _ColladaExportSettings.xml in the Assets of the current Unity project. If this file does not exist is will be created with the default settings preset.

Export
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. 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.

Batch script
Example batch script is provided if you want to use the exporter in your own script. The file api_cs.txt (that can be renamed api.cs) provides the source code to set-up the parameters and call the exporter. The result will be a XML document that you'll have to save yourself.