
Table of Contents

Introduction	1.1
v3.1	1.2
v3.2	1.3
Installation	1.3.1
Export of OBJ files	1.3.2
Materials and Textures	1.3.3
Animations	1.3.4
Conditions	1.3.5
FAQ	1.3.6
v3.3	1.4
3.3 Installation - Detailed Notes	1.4.1
Materials and Textures	1.4.2
Export of OBJ files	1.4.3
Animations	1.4.4
Conditions	1.4.5
Advanced Techniques	1.4.6
Manipulations	1.4.7
FAQ	1.4.8
v3.4	1.5
Installation	1.5.1
XPlane2Blender Versions	1.5.2
Quick Example	1.5.3
Scene Settings	1.5.4
Object Settings	1.5.5
Layer Settings	1.5.6
Material Settings	1.5.7
Texture Settings	1.5.8
Lamp Settings	1.5.9
In Depth Articles	1.5.10
Aiming Special Non-WYSIWYG Lights	1.5.10.1

v3.5	1.6
Empty Settings	1.6.1
Layer Settings	1.6.2
Object Settings	1.6.3
In Depth Articles	1.6.4
Animation Requirments For Smart VR Manipulators	1.6.4.1
Reporting Bugs	1.7
Donations	1.8

XPlane2Blender Documentation

Welcome to XPlane2Blender's manual for versions v3.1 and up. Choose an exporter version from the table of contents.

This manual was written with the help of

- Alex Unruh
- Andrzej Borysewicz
- Ben Supnik
- Ondrej Brinkel
- Ted Greene

and everyone else who has contributed to it!

Scenery and Cockpit import/export scripts for Blender

Preface

This is the final version of Xplane2Blender v 3.10. It's been tested and works with Blender 2.49 on Mac, Windows, and Linux. Development of this script is no longer done by Marginal, but collectively and in an Open-Source manner. So please file requests and issues at the project website.

Overview

Blender is an open source 3D object editor for Windows, Mac and UNIX.

These Blender scripts export models created in Blender to X-Plane v7, v8, v9 or CSL .obj format.

The scripts also import existing X-Plane v6, v7, v8, v9 and CSL .obj files and X-Plane v7, v8 and v9 .acf airplanes into Blender.

Requirements

Runs on Windows 2000 or later, Mac OS 10.3.9 or later, and Linux.

Requires Blender 2.45 or later:

Windows:

[Download](#) and run the "Installer" build of Blender.

Mac OS 10.5:

[Download](#) the "Python 2.5" build of Blender for Intel or PPC. To install, copy the Blender application out of the disk image eg to your /Applications folder.

Mac OS 10.3 & 10.4:

[Download](#) the default "Python 2.3" build of Blender for Intel or PPC, unless you have installed Python 2.5. To install, copy the Blender application out of the disk image eg to your /Applications folder.

Linux:

Most distros come with Blender, so you can install it via your package manager (eg Synaptic, YaST, YUM). Or [download](#) the build that corresponds to the version of Python that is installed on your system.

Changes for 3.10 Final

The two major changes since Beta 3 are:

- support for ATTR_light_level (this allows you to assign an object in Blender with the ability to independently "turn on" the LIT texture, based on a given dataref.)
- Fixed the multiple texture bug that was still present in Beta 4. This applies ONLY to the cockpit .obj, since it's the only one to support 2 textures. Now the script correctly interprets which of the textures is assigned to which animated mesh in Blender, and exports it accordingly.

ATTR_light_level

The way you use ATTR_light_level is outlined in detail on the obj8 spec website. That site describes what the resulting .obj file would look like in a text editor. This script allows you to store that information in Blender, so that it can be saved as part of your aircraft's source files.

What this feature does is it allows you to swap the daytime and nighttime (LIT) textures on a per-mesh basis, triggered by a dataref of your choice. This allows you to, for example, build 3D buttons that "light up" when their particular dataref is triggered.

Here's how you would do it in Blender. Let's say we're dealing with a 3D "fuel pump" button that's supposed to light up when the fuel pump is on. First, you'd have to have a daytime and LIT version of the texture. Without ANY CHANGES, this button's behavior would be such, that when X-Plane goes from day-mode to night-mode, this button's texture would swap from daytime to LIT texture. However, you can override this, and cause the texture swap to take place when the dataref for the fuel pump is called up. With the button selected (pink) in Blender, call up the "Game Logic" button (F4 keyboard shortcut). Click on "Add Property" and type "ATTR_light_level" (no quotes) into the space labeled "Name:" Then select "String"

from the drop-down menu directly to the left of this blank. Now enter the dataref for the fuel pump into the blank space to the right. (in this case, it would be "sim/cockpit/engine/fuel_pump_on").

You can now save this .blend file, and the lighting information you've entered for this button should be grabbed by the new exporter.

Tutorials

This document is a reference and assumes familiarity with Blender. There is a detailed series of video tutorials on YouTube that aim to provide developers with the background and techniques needed to make basic planes, and to continue making detailed planes using this script and Blender. The [first 14 tutorials](#) are aimed at familiarizing you with PlaneMaker.

The [following tutorials](#) showcase how to create and animate detailed .obj files that you can append to your .acf file, and even allow you to replace PlaneMaker components, which are low-res, low-polygon parts, with highly detailed, intricately textured .obj files.

- [Using Blender to create X-Plane scenery](#)
- [Blender3D: Noob to Pro](#)
- [The Blender Manual](#)

Installation

Run Blender to check that it is installed correctly. If the Blender window appears "blurry" turn off anti-aliasing in your video card's OpenGL settings and re-start Blender.

Now close Blender.

Windows Vista:

Extract the contents of the zip file to a new (temporary) folder. Right-click on the file install.cmd and choose Run as administrator. (This installs the scripts in the folder %HOME%\blender\scripts if it exists, otherwise in the copy of Blender that is associated with .blend files).

Windows 2000/XP:

Extract the contents of the zip file to a new (temporary) folder. Double-click on the file install.cmd. (This installs the scripts in the folder %HOME%\blender\scripts if it exists, otherwise in the copy of Blender that is associated with .blend files).

Mac OS:

Double-click on the file `install.command`. (This installs the scripts in the folder `~/blender/scripts` if it exists, otherwise in all copies of Blender that the Finder knows about).

Linux:

Start a shell window, and type: `mkdir -p ~/blender/scripts`

Extract the contents of the zip file to this folder.

(Re)start Blender. This file can now be viewed by choosing `Help → X-Plane` in Blender.

You can now delete the temporary folder.

If the above mentioned installation of the scripts does not work, you can still open these Python-scripts in Blender's text-editor window, and run each script by pressing "Alt+P." This method can also save time. If, for instance, you keep the Python-script in one of Blender's text-editor windows, it is easier to call it up via the "Alt+P" key command, than by browsing to the appropriate plug-in via the pop-up menus in the Python window.

The two most commonly used Python scripts once development is in full swing, are the "XplaneAnimObject.py" and the "XplaneExport8.py." I've found that keeping these scripts in a small window in your Blender project makes animating and exporting X-Plane .obj's faster and easier.

Importing objects

First, move the 3D Cursor to where you want the imported object to be placed. Usually you'll want the object to be placed at the origin so just press `Shift C` to centre the 3D Cursor at the origin.

Choose `File → Import → X-Plane Object`, select a .obj file and press `Import OBJ`.

The scenery is imported at the 3D Cursor position.

You can import multiple .obj files and re-export them as a single file. But note that the X-Plane .obj file format only supports the use of one texture file, so you'll have to create a single larger file containing all required textures - see below. Or you can have Blender create this single file automatically by selecting all the objects and, in a UV/Image Editor window, choosing `Image → Consolidate into one image`.

Importing planes

First, move the 3D Cursor to where you want the imported plane or weapon to be placed. Usually you'll want it to be placed at the origin so just press Shift C to centre the 3D Cursor at the origin.

Choose File → Import → X-Plane Plane or Weapon and choose whether to import the plane or weapon so that the "reference point" is located at the 3D Cursor (for making cockpit & misc objects) or so that the "centre of gravity" is located at the 3D Cursor (for making CSLs and static scenery).

Then select a .acf or .wpn file and press Import ACF or WPN.

The script creates up to three versions of the plane or weapon in Blender, one in each of layers 1, 2 and 3. The versions in layers 2 and 3 use approximately 1/10th and 1/100th of the number of faces compared to the version in layer 1. See "Level of Detail" below for an explanation of why it does this.

Imported planes need some tweaking before you can export them as scenery or as a CSL object; see "Tweaking planes" below.

Exporting objects

First, choose File → Save As... and save the .blend file in the aircraft or scenery folder where you want the .obj file to end up. Then choose:

```
File → Export → X-Plane CSL Object or  
File → Export → X-Plane v7 Object or  
File → Export → X-Plane v8/v9 Object
```

The object is exported in the same folder and with the same name as the current Blender file, but with a .obj extension. Blender may display some informational messages - click on one of these messages to see which object(s) the message refers to.

If there is an error then the scripts will attempt to identify and highlight the offending Blender object(s).

cs: These objects are intended for use with multi-user plugins such as [XSquawkBox](#) or [X-lvAp](#). v8: These objects are supported by X-Plane versions 8.20 and later. v9: These objects are supported by X-Plane versions 9.00 and later.

Creating X-Plane scenery

- Find and open the Custom Scenery folder inside of your X-Plane installation.

- Create a subfolder with the name of the scenery package that you're making.
- Save your Blender file in this subfolder folder with a descriptive name, eg:

```
X-Plane/Custom Scenery/EGLL/house.blend
```

```
The X-Plane .obj file will be exported to the same place, ie:
```

```
X-Plane/Custom Scenery/EGLL/house.obj
```

Tweaking planes

Imported planes need to be positioned correctly on the ground for use as static scenery (don't do this if you're making a CSL or cockpit object):

- Select layers 1-3, select all objects and position the plane so that its undercarriage is sitting directly on the ground (represented by the x/y axes).
- You may also need to rotate the plane slightly so that all wheels are level; press r and move the mouse to rotate the plane so that its undercarriage is sitting directly on the ground. Click to set the plane's position.

The primary file for textures is named `airplane_paint`. Most planes also use textures from a secondary file named `airplane_paint2`. Objects that use textures from the secondary file are imported with "" after their name to make them easier to identify in Blender's Outliner window.

The X-Plane .obj scenery file formats only support the use of a single file for textures, up to 1024x1024 pixels in size. If your plane only has a few simple objects that use textures from `airplane_paint2` then you should re-texture these objects to use `airplane_paint`, following the same procedure described below for weapons and misc objects. If that is not feasible you can use this procedure to make the plane use textures from a single file:

- Save your Blender model.
- In an image editor application, resize `airplane_paint` and `airplane_paint2` to 512x512. (You don't need to save these resized versions).
- Create a new bitmap file 1024 pixels wide and 512 pixels high.
- Paste the resized `airplane_paint` into the left half of this bitmap.
- Paste the resized `airplane_paint2` into the right half of this bitmap.
- Save the bitmap file in the same folder and with the same name as your .blend file. If you're making a CSL object then you must save in PNG format.
- If your plane uses night-time textures then repeat this procedure for the `_LIT` bitmap files.
- In a Blender UV/Image Editor window choose Image → Merge `_paint` and `_paint2`.

If your imported plane uses weapons or misc objects then each of these will use an additional bitmap file. Weapons are imported with their names starting with "Wnn" and objects with their names starting with "Onn". Also note that reduced-LOD versions of weapons and misc objects may be present in layers 2 and 3.

Open an Outliner window and choose View → Show Outliner. For each mesh that has a name starting with "Wnn" or "Onn" or ending with "*", either:

- Delete the mesh, or
- Copy the required textures to an unused area in the primary bitmap file and use the UV/Image Editor window to map the new copy of the textures to the mesh's faces.

Consider performance issues when the plane is rendered in X-Plane. Ask yourself the following questions:

- Most important: Do you really need fully detailed 1024x1024 textures for your plane? Video memory is used up by terrain and object geometry and textures. Once you run out of video memory the GPU has to fall back to main memory, and this really slows things down. One 1024x1024 texture uses 4MB of video memory. Some people are running X-Plane on computers with only 32MB of video memory, so one 1024x1024 texture at "extreme res" in X-Plane uses 1/8th of their video memory. Consider resizing the texture file in an image editor program to half or even quarter size. Use Image → Replace to use the new texture file.
- Does the model have hidden faces? Some versions of X-Plane don't handle hidden faces in v7 scenery objects very well and they also cause a small performance hit. Look for things like wings or misc bodies that are partially or wholly buried in the fuselage and delete any wholly hidden faces before exporting.
- Do you really need all that detail? Consider deleting details like flap tracks, antennae etc before exporting. This especially goes for the lower Level of Detail versions of the plane in layers 2 and 3 which are only viewed in X-Plane from >1000m and >4000m respectively.

Creating 3D cockpits

X-Plane 3D cockpits are just normal v7, v8 or v9 scenery objects except that cockpits can't contain multiple Levels of Detail, so only objects in Blender layer 1 are exported. If you want to keep objects in your Blender scene for reference but which you don't want to export - eg the plane fuselage - then put them in layer 4 or greater before exporting.

Choose File → Save As and save the blender file in the same folder as your plane's .acf file with the name airplane_cockpit.blend, airplane_cockpit_INN.blend or airplane_cockpit_OUT.blend (where airplane is the name of your plane's .acf file):

- X-Plane displays airplane_cockpit.obj in both internal and external views.
- X-Plane displays airplane_cockpit_INN.obj only in internal views.
- X-Plane displays airplane_cockpit_OUT.obj only in external views. If you create an airplane_cockpit_INN.obj and/or airplane_cockpit_OUT.obj then you should not create airplane_cockpit.obj.

You will usually want to import your plane into Blender to act as a reference and/or starting point for your cockpit. Delete any plane parts that you don't need in creating your cockpit - you only need to keep the fuselage itself plus any relevant Misc Bodies. After import, your cockpit uses the same texture file as your plane, ie airplane_paint. Choose Image → Replace in a UV/Image Editor window to use a different texture file, which can be named anything you like (but no spaces) and which should live in the same folder as your plane's .acf file.

To make your 3D cockpit appear in X-Plane, on the Standard → Viewpoint → View screen in PlaneMaker, check the show cockpit object in: INSIDE views, exact forwards option. To hide the 2D cockpit altogether, also check the show cockpit object in: PANEL views, exact forwards option; hiding the 2D cockpit means that you no longer have to leave a large part of the Panel Texture transparent to represent the windscreen, which gives you more room on the Panel Texture for instruments.

Cockpit instruments

To construct moving cockpit instruments paint the .../cockpit/-PANELS-/Panel.png texture in an image editor application and place instruments in PlaneMaker as as you would for a 2D panel (but bear in mind that only the top 768 lines of the Panel Texture can be used in the 3D cockpit in X-Plane versions prior to 8.20). The Panel Texture can be 1024×any size in X-Plane v8, and any size up to 2048×2048 in X-Plane v9. Normally you can only use a single file to texture your X-Plane objects. But when constructing a 3D cockpit you can additionally use this .../cockpit/-PANELS-/Panel.png file - the instruments that X-Plane draws on the 2D panel will also appear in your 3D model.

The .../cockpit/-PANELS-/Panel.png texture doesn't contain any instruments when you load it into Blender (unless you've painted them on yourself). This makes it hard to tell in Blender where X-Plane will draw the instruments. So it's easier if you use a screenshot of the panel with the instruments drawn on it, instead of the real Panel Texture. If your display is larger than your Panel Texture, then this is simple:

- Run PlaneMaker.
- Choose Background → Rendering Options and set the size to be equal to the size of your Panel Texture.
- Restart PlaneMaker

- Choose Standard → Panel
- Take a screenshot: Press Alt PrintScreen (PC) or Command Shift 3 (Mac).
- Paste (PC) or load (Mac) the screenshot into an image editor application.
- Crop the window borders etc from the screenshot so that the image is exactly the same size as your Panel Texture.
- Save the screenshot as ScreenshotPanel.png (or any filename ending in *panel*.) in the same folder as your plane's .acf file.
- Use the ScreenshotPanel.png texture on those faces that you want to display moving cockpit instruments in X-Plane.
- The screenshot file does not need to be distributed with your finished plane.

If your Panel Texture is larger than your display then you cannot take a screenshot of the whole panel. In this case you'll need to take multiple screenshots of the panel in PlaneMaker and stitch them together in an image editing application.

If you later want to resize your Panel Texture then use the procedure described below.

Note that X-Plane versions prior to 8.20 only display the 3D cockpit when running at the default 1024x768 resolution. You may want to mention this in the Readme with your plane if your plane is intended to work in X-Plane versions prior to 8.20.

v9: Cockpit Panel Regions

The cockpit Panel Texture uses a lot of video memory, much of which is wasted when the 3D cockpit is being displayed:

- X-Plane has to round up the height and width of your Panel Texture to be powers of two. eg if your Panel Texture is 1600×1200 pixels then X-Plane rounds this up to 2048×2048 pixels, which requires 16MB of video memory. More if you also supply a LIT Panel Texture.
- Typically up to half of your Panel Texture represents your plane's windscreen, which is fully transparent. You can't make use of this part of the texture in any useful way in a 3D cockpit, so the memory that it consumes is wasted. (Note: You can construct a tinted windscreen in your 3D cockpit quite cheaply by using a small semi-transparent part of the non-panel texture).
- The Panel Texture contains an alpha channel for transparency. The alpha channel accounts for ¼ of the memory that the texture consumes. But often your only need for transparency in the Panel Texture is to represent the 2D windscreen, which is of no use in a 3D cockpit, so the memory that the alpha channel consumes is wasted.

A "Panel Region" is a new texture which is cut out from your Panel Texture:

- You can create up to 4 Panel Regions (which can overlap).

- The height and width of a Panel Region texture must be a power of two eg 128, 256, 512, 1024 or 2048, but it doesn't have to be square.
- Panel Region textures are opaque - they don't contain an alpha channel.

When you use Panel Regions instead of the Panel Texture to texture your 3D cockpit, X-Plane discards the Panel Texture's alpha channel and also discards all areas of the Panel Texture other than the pieces that you cut out to make the Panel Regions. This reduces video memory requirements and improves performance.

Creating a Panel Region

- In the UV/Image Editor window, select your Panel Texture from the pop-up menu.
- Choose Image → X-Plane panel regions → Create new region
- Enter the co-ordinates in your Panel Texture where you want the bottom-left pixel of the new Panel Region to start, and the width and height of the new Panel Region.

Any faces that you've textured using the Panel Texture which are contained inside the new Panel Region are transferred over to use the new Panel Region.

Any areas that are fully transparent in the Panel Texture are coloured sky blue in the new Panel Region. You'll get undefined (ie weird) results in X-Plane if you use these sky blue areas to texture your faces.

(Note: When you create a Panel Region, Blender also creates a hidden object named "PanellRegionHandler" to store accounting information. Don't mess with this object).

Deleting a Panel Region

- In the UV/Image Editor window, select your Panel Region from the pop-up menu.
- Choose Image → X-Plane panel regions → Delete this region

Any faces that you've textured using this Panel Region are transferred back to using the Panel Texture.

The deleted Panel Region will remain in the UV/Image Editor window's pop-up menu for a while until Blender figures out that it can remove it. But the deleted Panel Region won't count towards your maximum of four Panel Regions.

Re-loading the Panel Regions

The Panel Regions aren't automatically updated when you edit your Panel Texture in an image editor application and then reload it in Blender, or when you reload your .blend file.

- In the UV/Image Editor window, select your Panel Texture from the pop-up menu.

- Choose Image → X-Plane panel regions → Reload all regions

Using Blender to create X-Plane objects

Only Lamps and Meshes are exported to X-Plane. You can use other Blender object types, eg Curves and Surfaces, to construct your scenery as long as you convert them to meshes before exporting to X-Plane.

Lamps

Only Lamp objects of type "Lamp" are exported to X-Plane. Lamp objects of types "Area", "Spot", "Semi" and "Hemi" are ignored (and so can be used to illuminate your model in Blender).

v8/v9: Lamp objects with certain words in their names have special behaviours when exported to an X-Plane v8 or v9 object:

- *Lamp* - Normal (legacy) light. The colour is determined by the R,G,B sliders on the Lamp panel (**F5**).
- *Flash* - Flashing (legacy) light. The colour is determined by the R,G,B sliders on the Lamp panel (**F5**).
- *Traffic*, *smoke_black*, *smoke_white* - As for X-Plane v7 objects; see below. (R,G,B settings are ignored).
- other - X-Plane pre-defined "named" or "custom" light. (Supported by X-Plane 8.50 and later. R,G,B settings are ignored). The name of the X-Plane light is taken from the value of a property named name if this exists, otherwise from the name of the lamp object.

v7: Lamp objects with certain words in their names have special behaviours when exported to an X-Plane v7 object:

- *Flash* - Flashing light. The colour is determined by the R,G,B sliders on the Lamp panel (**F5**).
- *airplane_beacon* - Red pulsing anti-collision light. (R,G,B settings are ignored).
- *airplane_strobe* - White strobe light. (R,G,B settings are ignored).
- *Traffic* - Cycles red, orange, green. (R,G,B settings are ignored).
- *smoke_black* or *smoke_white* - Not really a light; emits smoke. The size of the smoke puffs is determined by the Energy slider on the Lamp panel (**F5**) (R,G,B settings are ignored).
- other - Normal light. The colour is determined by the R,G,B sliders on the Lamp panel (**F5**).

csi: Lamp objects with certain names have special behaviours when exported to an X-Plane CSL object. The XSquawkBox documentation strongly recommends that you use these special lights:

- *airplane_landing* - White landing light. (R,G,B settings are ignored).
- *airplane_taxi* - White taxi light. (R,G,B settings are ignored).
- *airplane_nav_left* - Red navigation/position light. (R,G,B settings are ignored).
- *airplane_nav_right* - Green navigation/position light. (R,G,B settings are ignored).
- *airplane_beacon* - Red pulsing anti-collision light, on when engines are running. (R,G,B settings are ignored).
- *airplane_strobe* - White strobe light. (R,G,B settings are ignored).
- *other* - Normal light. The colour is determined by the R,G,B sliders on the Lamp panel (**F5**).

v8/v9: Custom lights (supported by X-Plane 8.50 and later) are created using the vertices from a Mesh object: In the Material buttons panels (**F5**) add a new material to the mesh, then press the Halo button on the Links and Pipeline panel. You should use just one material.

- The Halo button and the R,G,B,A sliders on the Material panel control the light's R,G,B and A values. Alternatively you can create [property](#) named *R*, *G*, *B* and/or *A* to set these values.
- The HaloSize control on the Shaders panel controls the light's S value. Also press the HaloTex button on this panel to make Blender render the light correctly.

In the Texture buttons panels (**F6**) add a new texture to the material, and change the Texture Type to *Image*. You should use just one texture.

- On the Image panel load the texture file that contains the light that you want to use.
- On the Map Image panel press the UseAlpha button. Use the MinX, MinY, MaxX, MaxY settings to select a subset of the texture.

To drive the custom light using a dataref add a String [property](#) named *name*.

Meshes

Create faces with 3 or 4 edges (called "*tri*"s and "*quad*"s in X-Plane).

In the Link and Materials Editing panel (**F9**):

- Set Smooth and Set Solid control whether to smooth edges of faces in a mesh. This is useful when using multiple faces to simulate a curved surface. Go to "Object Mode", select the mesh and press Set Smooth. The effect is only visible in Blender 3D View windows when the Viewport Shading button is set to Solid or Shaded.

v7: Only faces that are part of a [Strip](#) will be smoothed when displayed in X-Plane.

In the Texture Face Editing panel (**F9**) available in UV Face Select mode:

- **Tex** button controls whether the face has a texture:
In an image editor application create one texture file that is like a "collage" of all of the textures that you need in your model. The height and width of the texture file must be a power of two eg 128, 256, 512, 1024 or 2048 (X-Plane v9 only), but it doesn't have to be square. (See `.../Custom Scenery/KSBD Demo Area/KSBD_example.png` for an example). Save the texture file in 32bit or 24bit PNG format (ie with or without an "Alpha" channel) in the aircraft or scenery folder where you want the X-Plane `.obj` file to end up. Use the UV/Image Editor window to control mapping of the textures to the face.
- **Tiles** button controls whether the face is rendered with "polygon offsetting" (`ATTR_poly_os`) in X-Plane. Press this button for faces that lie flat on the ground to prevent Z-buffer thrashing in X-Plane. Don't press this button for other faces. For best results with X-Plane versions prior to 8.50 you should ensure that objects that use polygon offsetting are listed first in your `.env` or `.dsf` scenery file.
v7: "polygon offsetting" does not produce reliable results in X-Plane versions prior to 8.20.
cs! This button has no effect when exporting CSL objects.
- **Collision** button indicates that the face is not "hard" (ie not "landable on") in X-Plane. Making faces hard is very expensive, so this button should normally be pressed. Unpress this button only for things like helicopter landing pads.
(Note: The meaning of this button was changed in v1.50. Use [this](#) script to turn the Collision button back on for all faces).
v9: You can control whether it is possible to fly under this hard face; add a Bool [property](#) named `deck` and give it the value `True`. You can't fly under hard faces in X-Plane v8.
v8/v9: You can specify the surface type; add a String [property](#) named `surface` and give it the value `water`, `concrete`, `asphalt`, `grass`, `dirt`, `gravel`, `lakebed`, `snow`, `shoulder` or `blastpad`. The surface type is ignored by X-Plane versions prior to 8.50.
v7: Only faces with 4 edges (ie "quads") are exported as "hard".
v7: X-Plane versions prior to 8.20 have a bug where `.obj` files that contain hard faces must be placed in WorldMaker with an "object heading" of 0. Otherwise the "hard" part of the surface ends up in the wrong place.
cs! This button has no effect when exporting CSL objects. ***Twoside** button controls whether one or both sides of the face are displayed. Unless you have a lot of double-sided faces it is cheaper to avoid this button and to use two single-sided faces back-to-back instead.
- **Alpha** button is a hint to Blender and to X-Plane that the face is transparent or translucent. Use this for outwards-facing transparent or translucent faces to instruct X-Plane to draw these faces last so that you can see other faces through them. Don't

press this button for opaque (normal) faces. See [drawing order](#) for more fine-grained control over drawing order.

v8/v9: In the Material buttons panels (**F5**) you can change the way that the mesh reacts to light by specifying a material. Changing between materials in X-Plane is expensive so you should ensure that you only use a few materials in your model. Press Add New to create a new material, or choose an existing material (if any) from the drop-down list. You should use just one material. Only a few of Blender's many material buttons affect X-Plane:

- Col button and the R,G,B sliders on the Material panel control the diffuse colour of the faces. X-Plane combines the colours that are specified by the texture (if any) with this setting. The default X-Plane setting is *1, 1, 1* (white). However the default setting of a new material in Blender is *0.8, 0.8, 0.8*. You should set this to *1, 1, 1* - control the diffuse colour of the faces by editing the texture file instead.
- Spec slider on the Shaders panel controls the specularly (shininess) of the faces. The default X-Plane setting is 0 (matt). However the default setting of a new material in Blender is 0.5.
- Emit slider on the Shaders panel controls the emissive brightness of the faces. Emissive faces give off light so the effects of this setting are most obvious at night. It is usually cheaper and easy to use **LIT_ textures to control night-time brightness instead of using this setting. However using this setting allows you to create faces that also give off light during the daytime, eg on overcast days, which LIT_ textures do not.** The default X-Plane and Blender setting is 0 (not emissive).
- Mir button and the R,G,B sliders on the Material panel control the emissive colour of the faces.

Use [this](#) script if you need to reset all faces in the scene to standard settings (ie no polygon offsetting, not hard, single sided, not transparent).

You can add "modifiers" in the Modifiers panel to change the way that the Mesh appers. Some useful modifiers when modelling for X-Plane are:

- EdgeSplit - automatically sharpen edges between mesh faces (this only has an effect if you've used the Set Smooth button)
- Subsurf - produce a more detailed version of the mesh by subdividing faces
- Curve - bend the mesh along a curved path

Lines

Blender doesn't support Lines directly. Use a mesh with one 4-edged face instead. The pair of vertices at each end of the "line" must be within 0.1 units of each other. The face must be the only face in its mesh and must not have a texture assigned to it.

Assign a material to the face and use the Col button and the R,G,B sliders on the Material panel to control the colour of the line. Faces not linked to a material will be exported coloured grey.

v8/v9: Animation

You can make lamps, meshes and lines animate in X-Plane according to the value of any of the simulator datarefs listed [here](#) that have type "int", "float" or "double".

Basic animation

Create an "Armature" object. Make the lamps and/or meshes that you want to animate the children of the armature's "bone":

1. Click on the lamps and/or meshes (the "children")
2. Shift-click on the armature
3. **Choose Pose Mode from the Mode menu in the 3D View window's toolbar**
4. Click on the bone (the "parent")
5. Press **Ctrl-P** and select Bone from the popup menu

Once you have assigned a parent bone to your lamps/meshes, you can specify the simulator [dataref](#) that will drive the animation:

1. Choose Object Mode from the Mode menu in the 3D View window's toolbar
2. Select the child lamp or mesh
3. From the 3D View window's menubar choose Object → Scripts → X-Plane Animation
4. In the Parent Bone panel, use the pop-up menu to select the [dataref](#) (or you can type in just the "leaf" name of the dataref into the text field, or the full name of a custom dataref)
5. Some datarefs require you to specify a "Part number"; eg the dataref *sim/flightmodel/engine/ENGN_thro* represents the engine throttle settings, so you need to specify which engine you're referring to. Specify a "Part number" of 0 for the first engine, 1 for the second engine etc
6. Press the Apply button

Use frames to represent the desired position of the lamps/meshes at various dataref values - X-Plane will [interpolate linearly](#) between the positions:

1. In the Animation Frame field, in the Buttons window's menubar, select frame 1
2. Click on the armature
3. Choose Pose Mode from the Mode menu in the 3D View window's toolbar
4. Click on a bone
5. Move and/or rotate the bone
6. Press **i** and specify a *LocRot* key (ie location and rotation)

7. Repeat for any other bones in the armature
8. Select animation frame 2 and repeat

(Note: X-Plane's animation syntax is quite simple; so don't use IPOcurves, Vertex Groups, Deformations, Shape Keys or any other advanced Blender animation techniques since these will be ignored by the exporter - only the positions specified by keys in the first n frames are significant to X-Plane.)

v8: X-Plane v8 only supports two frames, so if you want your *.obj* file to work in X-Plane v8 then you should insert "LocRot" keys only in frames 1 and 2. If you insert keys in frame 3 and above then your animation will not work at all in X-Plane v8. Use the Delete button in the X-Plane Animation dialog to delete any keys from additional frames.

v9: You can add "LocRot" keys in as many frames as you like. If you skip a frame then Blender and X-Plane will use the pose from the previous frame.

The X-Plane Animation dialog renames the parent bone to the "leaf" name of the dataref. So pressing the Draw Names button in the Armature panel can be helpful to see what's going on when you have lots of animations.

Use the Action Editor window to get an overview of which bones in the selected armature have keys inserted into which frames.

Controlling animation response to dataref values

By default, X-Plane will display the meshes in the frame 1 position when the dataref has a value of 0, and in the last frame position when the dataref has a value of 1. You can change these values:

1. Choose Object Mode from the Mode menu in the 3D View window's toolbar
2. Select the child lamp or mesh
3. From the 3D View window's menubar choose Object → Scripts → X-Plane Animation
4. Specify in the Frame #n fields the dataref values that correspond to each frame; eg the yoke pitch dataref *yolk_pitch_ratio* takes values between -1 (forward) and +1 (back) in X-Plane. So, to make X-Plane display the position in frame 1 when the yoke is pushed fully forward specify *-1* in the Frame #1 field
5. Press the Apply button

v9: X-Plane will [extrapolate](#) your animation when the dataref has a value outside of the range that you specified in the *Frame #n* fields. You can stop the extrapolation and "clamp" your animation's position by repeating the poses and *Frame #n* values in the first two and/or the last two frames. Or you can cause your animation to loop back to frame 1 when the dataref value exceeds a certain number by specifying this number in the *Loop* field.

Using multiple datarefs

You can animate your lamps/meshes using multiple bones, each bone representing a different dataref:

1. In Edit Mode add additional bones to the armature.
2. Still in Edit Mode, in the Armature Bones panel, create parent/child relationships between each bone. (Note: This panel also lets you rename bones. Don't do this - use the X-Plane Animation dialog to name the bones after the datarefs that they represent).
3. Use the technique described [above](#) to make your lamps/meshes the children of the "youngest" bone in the chain.
4. Insert *LocRot* keys for every bone in the chain (each bone can have a different number of keys).

The X-Plane Animation dialog displays the settings for the lamp/mesh's parent bone, grandparent bone etc. (Note: Don't change the parent/child relationships between your lamps/meshes and their parent bones while the X-Plane Animation dialog is being displayed).

Hiding lamps and meshes

You can make all of the lamps and meshes in an animation disappear when a dataref is within a certain range:

1. Use the technique described [above](#) to make your lamps/meshes the children of an armature bone (If you don't want to animate your lamps/meshes then don't insert any animation keys for this bone, and the bone doesn't have to be a valid dataref) Choose Object Mode from the Mode menu in the 3D View window's toolbar Select a lamp or mesh that you want to hide From the 3D View window's menubar choose Object → Scripts → X-Plane Animation In the last panel, press the Add New button Specify the dataref and the range of values (it's OK to use datarefs that are not otherwise used in the animation)

You can make hidden lamps/meshes re-appear when a dataref value is within a certain range by adding another entry, and changing the type from Hide to Show. The dataref that you use to "show" the animation can be the same or different than the datarefs that you used to "hide" the animation. (The animation is always shown by default, so you only need to use a Show entry if you have used one or more Hide entries and you want to override them).

The order of Hide and Show entries is significant; the animation will be hidden if any of the Hide dataref values are in range, unless a subsequent Show dataref value is also in range. You can use the Up and Down buttons to change the order of the entries.

Note that the Hide and Show entries apply to all children of all bones in the armature. You can make an armature the child of a bone in a different armature; in which case all children are affected by any Hide and Show entries in parent armatures.

v8/v9: Drawing order

The order in which X-Plane draws the animations, lights, lines and triangles in your scenery or cockpit object usually has no effect on the appearance. So the exporter optimises the order of animations, lights, lines and triangles in your object to minimise the number of OpenGL state changes and therefore maximise X-Plane's framerate.

However drawing order does become important if you use transparent and/or translucent textures on some of your faces - transparent and translucent faces must be drawn last, otherwise other faces and lights will not be visible through them. You should therefore tell Blender which faces are transparent/translucent using the Alpha button in the Texture Face Editing panel (F9) in UV Face Select mode. The exporter will ensure that X-Plane draws these faces last.

But sometimes you need even more control over the drawing order - eg modelling a cockpit with a transparent HUD and (obviously) a transparent canopy; the HUD must be drawn after the canopy.

You can specify the relative order in which lamps, meshes etc should be drawn by assigning them to "Groups" on the Object and links panel (F7):

Objects that don't belong to a group are drawn first. The groups are sorted by alphabetical order, and then objects that belong to the first group are drawn next. ... objects that belong to the last group are drawn last.

eg in the case of the cockpit with transparent canopy and HUD, we could put the canopy and HUD into separate groups named GroupA and GroupB respectively. Since A is before B in the alphabet, the canopy would be drawn before the HUD (and both would be drawn after the rest of the cockpit).

To add objects to a new group:

Select the meshes that you want to be drawn late. Press the Add to Group button on the Object and links panel (F7) (or press Ctrl-G). Choose ADD NEW. Give the new group a name.

Objects that belong to a group are highlighted in green instead of pink so that you can easily distinguish them.

v8/v9: Drawing group

You can specify when X-Plane should draw your scenery object relative to other scenery elements:

Add a Blender object of type "Empty" to your scene. Add a property to the Empty object named `group_terrain`, `group_beaches`, `group_shoulders`, `group_taxiways`, `group_runways`, `group_markings`, `group_airports`, `group_roads`, `group_objects`, `group_light_objects` or `group_cars`. Give the property a value between -1 and -5 to make X-Plane draw your object before this group, 0 to draw your object with this group, or between 1 and 5 to draw your object after this group.

eg to make X-Plane draw your object at the same time that it draws runway markings add a property named `group_markings` to an Empty object and give it the value 0.

v8/v9: Slung load weight

You can specify the weight of an object for use in X-Plane's physics engine if the object is being carried by a plane or helicopter:

Add a Blender object of type "Empty" to your scene. Add a property to the Empty object named `slung_load_weight` and specify the weight in pounds.

Optimising for X-Plane

v7: Strips

As well as basic 3- and 4-edged faces ("tri"s and "quad"s), X-Plane v7 objects support two compound types - "tri_fan" and "quad_strip". These are strips of two or more tris and quads that share common edges. Because the faces in these strips share common edges, X-Plane and the underlying OpenGL renderer have a third or a half as much work to do to render them compared to individual tris and quads. This gives higher frame rates.

In order for a pair of faces to be considered for inclusion in one of these strips, the following conditions need to be true:

The faces must be facing the same way. Each pair of faces must share a common edge (apart from the first and last face). Each shared edge must have the same texture coordinates in both faces.

In practice this means that the texture must be reversed in alternate faces in the strip. In the case of tris this can also be achieved by mapping a single area of the texture across all the tris, with the tip of the tris at the centre of the texture area. Use UVs → Copy & Paste from the UV/Image Editor window to automate the creation of strips.

These compound types aren't supported directly by Blender. However, the export script automatically tries to spot when it can use them.

v8, v9 & csl: The performance gains from using strips are more modest in X-Plane v8 and v9, and for CSL objects. These compound types aren't supported in v8/v9 objects (X-Plane v8 and v9 use more advanced techniques) and the only saving from using strips over some other UV mapping methods is in reduced "vertex count". It probably isn't worth going out of your way to look for opportunities of making strips.

Level Of Detail

X-Plane has borrowed the concept of "Level Of Detail" from 3D games. This works on the principle that when you're viewing an object from a large distance it can be displayed with reduced detail without you noticing the difference. By displaying distant objects with reduced detail we can simulate a more complex scene than would be possible if all objects were drawn at maximum detail.

Use Layers to draw scenery and CSLs (but not aircraft Cockpits or Misc Objects) with multiple Levels Of Detail. Objects in layers 1-3 are visible in X-Plane at the following distances:

Layer

Distance

1

< 1000m

2

1000-4000m

3

4000m-10000m

Changing the texture size

If you run out of space in your texture file then you can increase the size. The panel texture can be 1024×any size in X-Plane v8, and any size up to 2048×2048 in X-Plane v9. The height and width of a non-panel texture file must be a power of two eg 128, 256, 512, 1024 or 2048 (X-Plane v9 only), but it doesn't have to be square.

You should use the following procedure to ensure that your UV mappings and/or PlaneMaker instrument layouts are preserved:

Create a new, larger, texture in an image editor application. Panel texture: Paste the original texture into the lower left corner of the new texture. Non-panel texture: Paste the original texture into one of the corners of the new texture, or aligned on a multiple of the original

texture's width and height. In Blender, in the UV/Image Editor window, select the original texture from the pop-up menu. Choose Image → Replace and fixup UV mapping... Select the new image and press Replace image. In the Fixup UV mapping dialog, press the button in the cluster of buttons that represents where you placed the original texture in the new texture.

You can also use this technique if you want to combine 3D models that use different texture files; paste all of the textures used by the 3D models into a single new texture file, then use Image → Replace and fixup UV mapping... on each of the original textures.

Blender tips

Blender has a quirky user interface:

By default, Right mouse button does what you would expect Left mouse button to do. I strongly recommend that you change this; drag down the top menubar to reveal the User Preferences window, select the View & Controls tab and choose Select with: Left Mouse. See this tutorial for a fuller explanation. Space does what you might expect Right mouse button to do. The user interface is "modal". Buttons and keys do different things depending on whether you are in "Object", "Edit", "UV Face Select" or "Pose" mode. (And I thought modal UIs went out with "vi"). Tab changes mode.

Auto-saving

Blender can automatically archive old copies of your .blend files when you save. This is useful if you later want to revert to an older version. In the User Preferences window, on the Auto Save tab, choose the number of Save Versions. The older versions of your file will be called .blend1, .blend2 etc.

Blender can also periodically save a copy of your work in case Blender or your computer crashes, or in case you quit Blender without saving. On the Auto Save tab press Auto Save Temp Files. Also, on the File Paths tab, ensure that Temp is set to a folder that exists on your machine. This is where the auto-saved files will be created. (The default value of /tmp is fine for Mac and Linux, but /tmp doesn't exist on most Windows machines so you should change this setting).

Properties

"Properties" are used to specify animation parameters, "named" lights and the drawing group. You can see an object's properties by selecting it and pressing F4 to display the Logic panel. Press the Add Property to create a new property.

Each property has three fields; Type, Name and Value. If you want to supply a non-numeric value, eg the name of a "named" light, you'll first need to change the Type from Float to String.

Limitations

Import object

smoke_black and smoke_white X-Plane primitives are ignored. ambient, blend and specular attributes are ignored. The importer usually can't work out which faces are partially or wholly transparent. You should tell Blender which faces are transparent using the Alpha button in UV Face Select mode after import.

Import plane

Importing planes made with PlaneMaker 7.30 or earlier is not supported. Convert an older plane to v7 by opening it in PlaneMaker 7.63 and re-saving it. Wings are simplified to reduce polygon count. Any wing curvature (ie the customize chords option in PlaneMaker) is ignored. Adjacent wing segments may not be exactly joined-up. If not corrected manually this can create shadow effects in XSquawkBox and X-IvAp. Cockpit objects are ignored. You can use File → Import → X-Plane Object to import cockpits. Planes (and their weapons, misc objects and cockpit objects) use multiple texture files. These are not automatically merged into one file during the import. The importer can't work out which faces are partially or wholly transparent. You should tell Blender which faces are transparent by pressing the Alpha button in UV Face Select mode after import. LOD algorithm can be insufficiently aggressive in layer 2.

Export object

Only Lamps and Mesh Faces (including "lines") are exported. All faces must share a single texture file apart from cockpit panel faces which can additionally use the cockpit panel texture (this is a limitation of the X-Plane .obj file formats). You can use Image → Consolidate into one image to merge multiple texture files into one file.

Problems

Please email any questions, problems etc to danklaue. If reporting a problem, please:

First, check that you are using the latest version of the scripts. Try to send me the the .obj, .acf or .blend file (where relevant) and the contents of the Console window. (In Mac OS X, this is the "Console" program. In Windows this the window titled "Blender" that looks like a Command Prompt).

License

These tools are licensed under the Creative Commons Attribution-Noncommercial-ShareAlike license.

Use of these tools does not impose any requirement on you to license your work under a Creative Commons license. For the avoidance of doubt, this means that you can license any scenery that you make using these tools under commercial terms (subject to any licensing restrictions on any imported or library objects that you use).

The team that is carrying on the development and upkeep of this script would appreciate a donation for the time and efforts spent on this script. Although you are under no obligation to do so, it would help in keeping the team motivated to continue updating and improving the script. If you wish to contribute, please [click here](#) to donate an amount of your choice via PayPal.

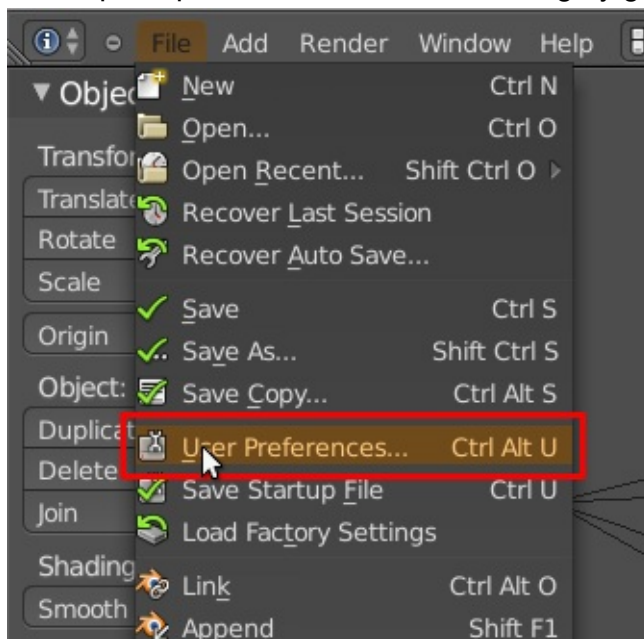
Documentation for XPlane2Blender v3.2

Install Blender

1. [Download Blender](#) for your system.
2. Install Blender
3. (Optional) Download and Install [Python 3.x](#). On some Systems Blender comes with a bundled Python and thus Python does not need to be installed one your system. You can see this in the system console when Blender starts. If Python is bundled it will write something like "Found bundled Python ...".

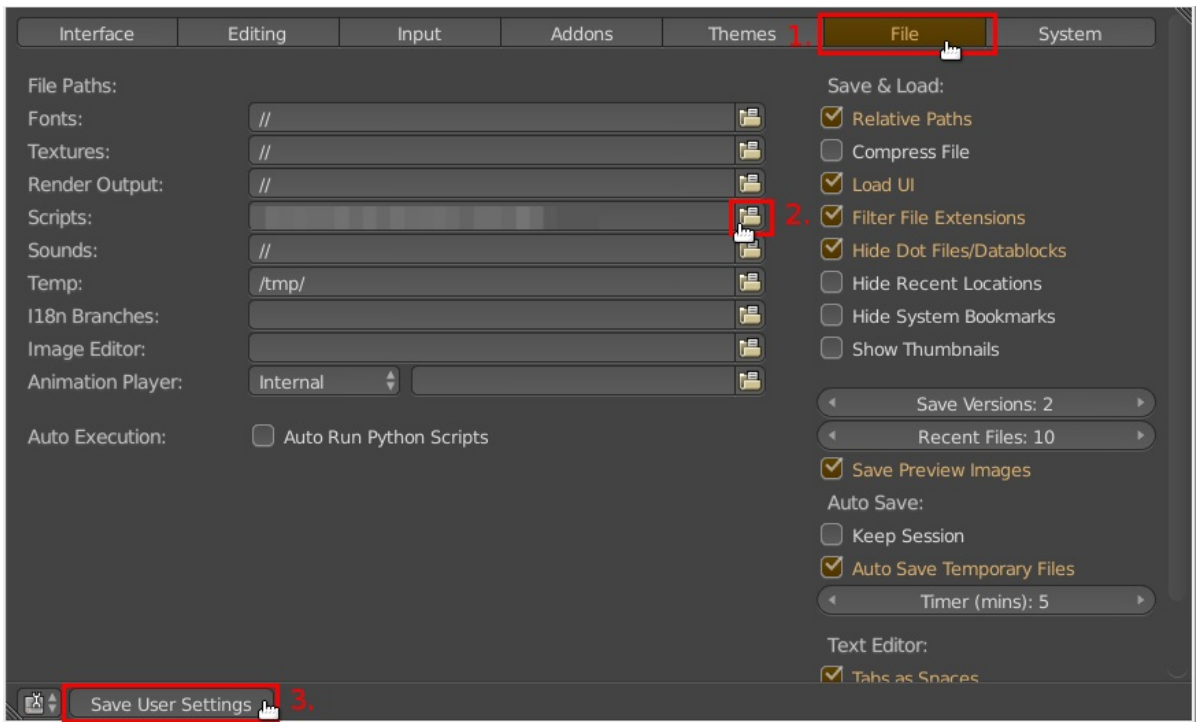
Setup blenders external scripts directory

1. Somewhere on your harddrive create a **blender-scripts** directory. Within that directory create an **"addons"** folder.
2. Open Blender and open up the File-Preferences Dialog by going to *File > User*



Preferences.

3. Select the *File* tab and hit the button next to the *Scripts* input field and locate the *blender_scripts* directory and hit *Save User Settings*.



4. Close blender.

Blender will now look for additional addons/scripts within your *blender-scripts* directory.

Install XPlane2Blender

1. Download the latest version of XPlane2Blender 3.2x [here](#) by selecting one of the listed downloads clicking the *zip* or *tar.gz* icons.



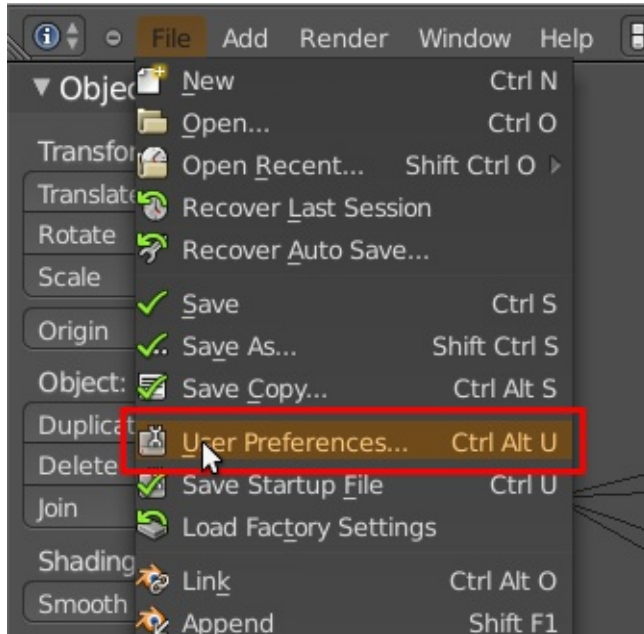
Releases / Tags



2. Extract the contents of the archive somewhere on your computer. You will get a folder named something like *XPlane2Blender-3.20.x*. Within this folder navigate to **blender_25/** and copy/move the folder **io_xplane2blender** into your **blender-scripts/addons** folder. (Afterwards you can safely delete the left behind folders, they are meant for developers.)

```
XPlane2Blender-3.20.x  
> blender_25  
> io_xplane2blender
```

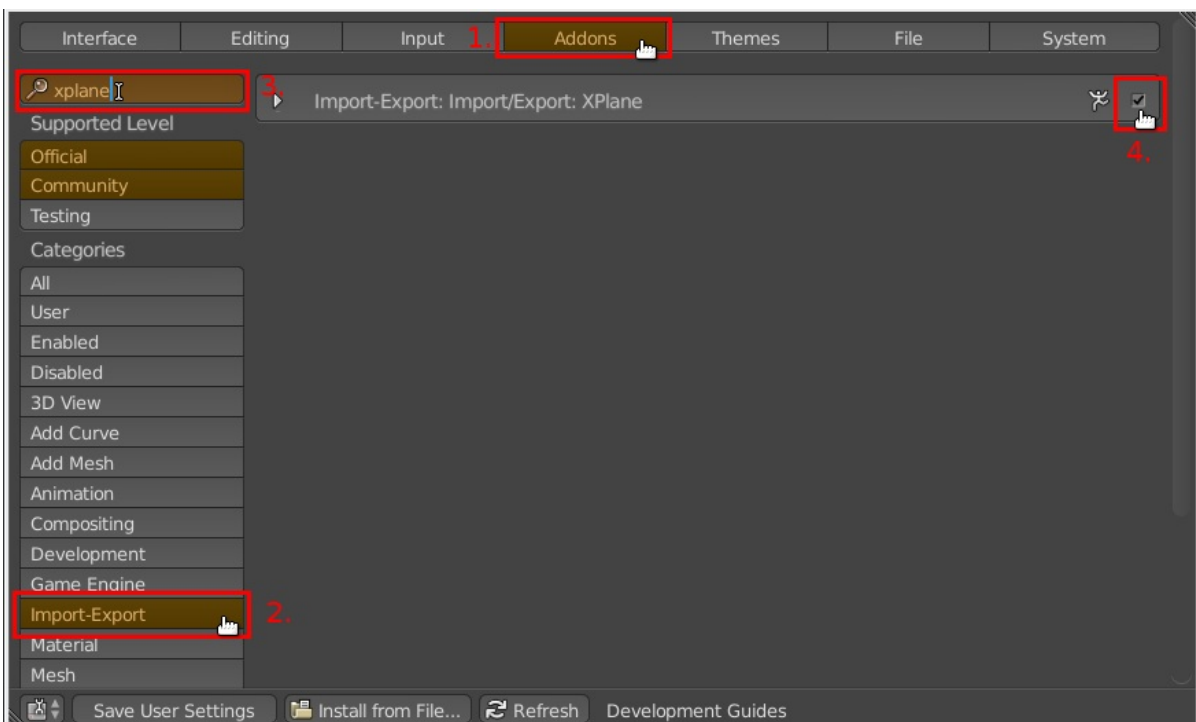
3. Open Blender and open up the User-Preferences Dialog by going to *File > User*



Preferences.

4. Go to the **Addons** tab select **Import-Export** as Category and start typing *XPlane* into the search box in the top left.

The addon *Import-Export: XPlane* should appear. Now Hit the checkbox next to it, to activate it. (Sometimes you must hit it twice to work). Finally save this configuration by clicking on *Save User Settings*. XPlane2Blender is now permanently activated in your Blender.



If you think this is a good thing [buy me a beer](#).

Please learn Blender first

If you read up to this point and never used Blender 2.6x before, please first [learn some on Blender basics](#).

Recommendation: Understand the OBJ file format

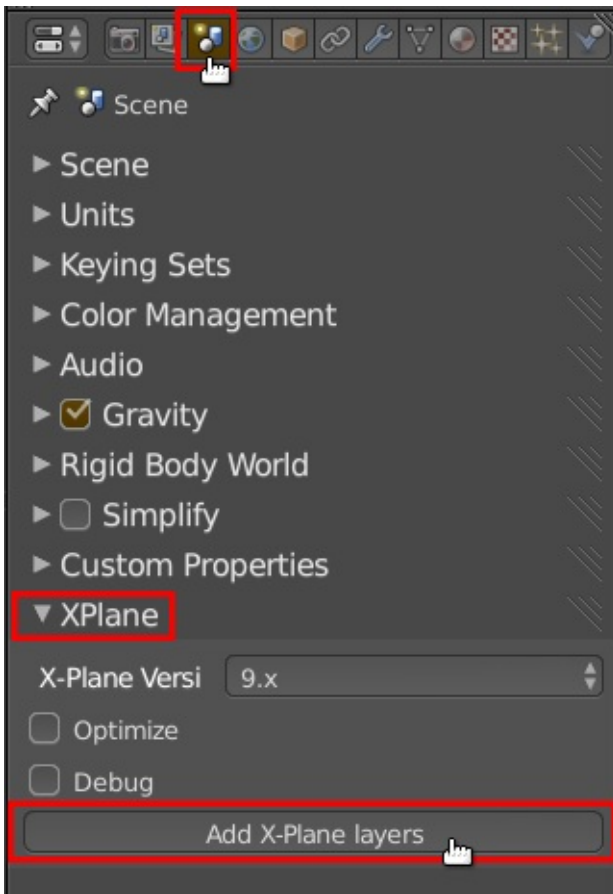
I also recommend you to [learn the concept of OBJ files](#). They are simple text files, you can investigate in any text-editor. Understanding the file format will help you in debugging your OBJ-Files and understanding XPlane2Blender better.

Steps to export an OBJ file

XPlane2Blender 3.2 uses Blenders layers for export of OBJ-files. Each layer represents one OBJ file. This way you can easily export multiple OBJ files at once and organize your OBJ files.

1. Activate X-Plane Layers

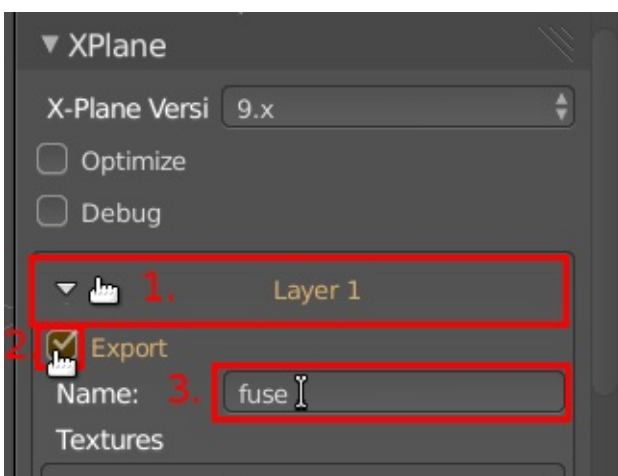
However to make the export work we need to initiate the Blender layers to make them work as OBJ-files. For this go to the *Scene* Tab in the *Properties* window. Now scroll down until you see a *XPlane* panel. Open it up, if it is not already. Hit the button *Add X-Plane layers*.



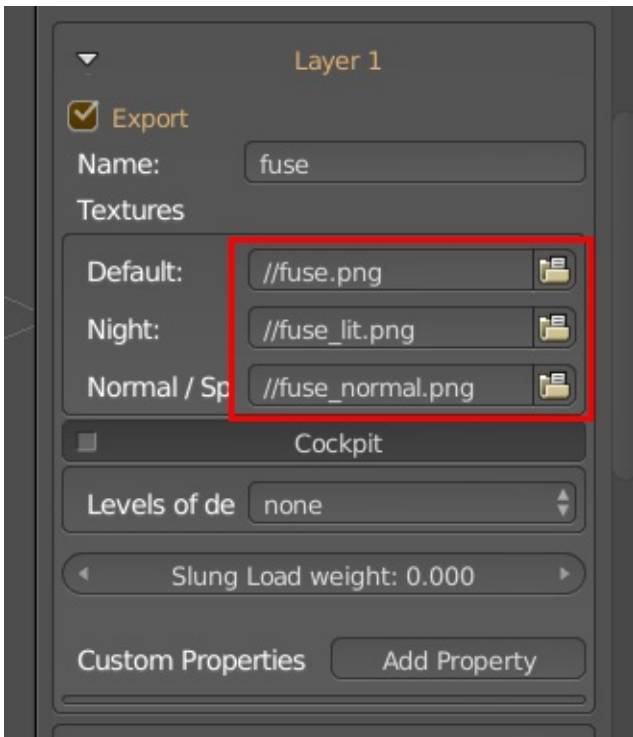
2. Configure a X-Plane layer

Now you have one subpanel for each Blender layer. Open up the first one labeled *Layer 1*.

If not enabled, check *Export* to make the layer exportable. Within the *Name* field type in the **name of the OBJ file** this layer should create, without the .obj extension. You can enter a relative path here.

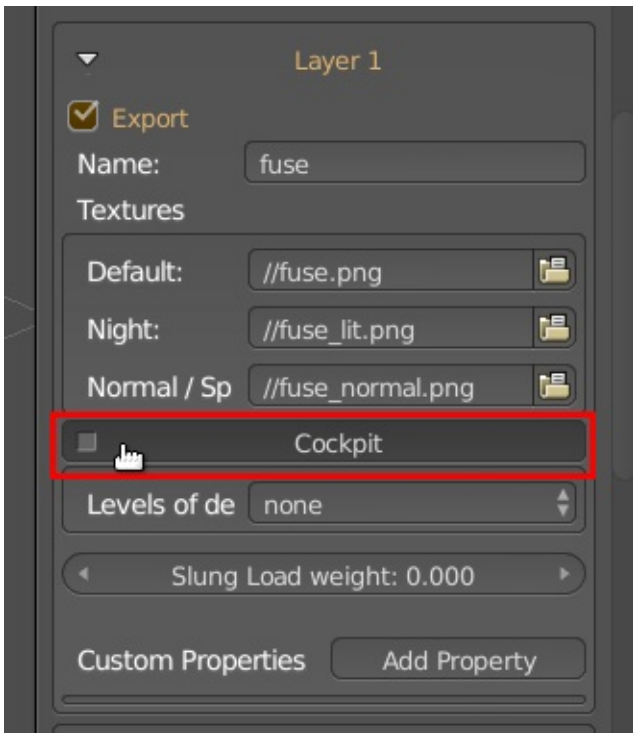


In the *Textures* group you can select the texture files the OBJ should use. One OBJ file can only use one texture file.



- The "Default" texture is the day texture,
- the "Night" texture will be used as the lit/night texture and emits light depending on the texture brightness,
- the "Normal/Specular" texture will be used for tangent space normal maps where the alpha channel defines the specular intensity.

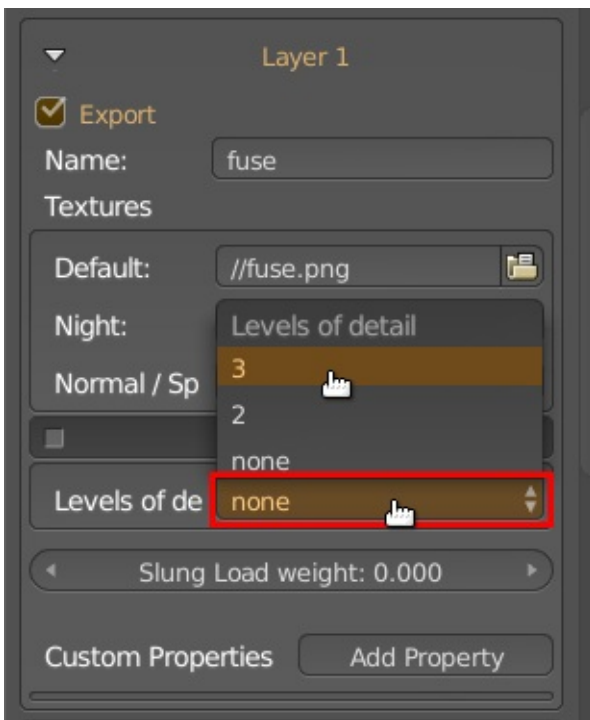
If the OBJ file is the **cockpit Object**, check *Cockpit*.



3. Advanced X-Plane layer configuration

If the OBJ will be attached to the aircraft as a slung load you can set it's weight in pounds using the *Slung load weight* slider.

If the OBJ will have levels of detail select how many in the *Levels of detail* dropdown.



Each level of detail (= **LOD**) has a near and far clipping distance. The LOD will only be visible within these distances. When dealing with multiple LODs be sure to set the Near clipping distance of the higher LOD to the Far distance of the preceding LOD as shown in the next image.



This will make your LODs compatible with X-Plane 9, as only X-Plane 10 allows for overlapping LODs.

You can add additional OBJ-Header Properties by hitting the *Add Property* button.

4. Check your Materials

XPlane2Blender will not export objects without a material. So check if the objects within that layer all have a material assigned.

5. Check your normals

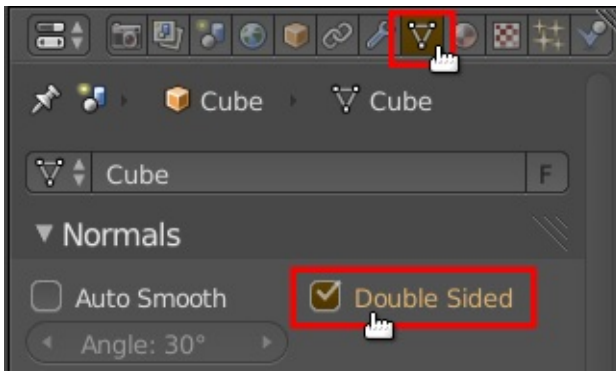
By default all objects exported to X-Plane are rendered with "single-sided" faces. This means that only the front side of a face will be visible. If the face is facing backwards it will simply be hidden. To be sure all faces point to the correct side you have to make the face normals of each object point outwards.

You can view the face normals of an object when in *Edit-Mode*. In the *View Properties* (hit N while hovering over the viewport) under *Mesh Display* activate the the display of face normals as lines.



If the lines are too short increase the *Size* next to this button.

The other option is to make all faces of an object "double-sided". This can be done in the mesh tab in the properties window.



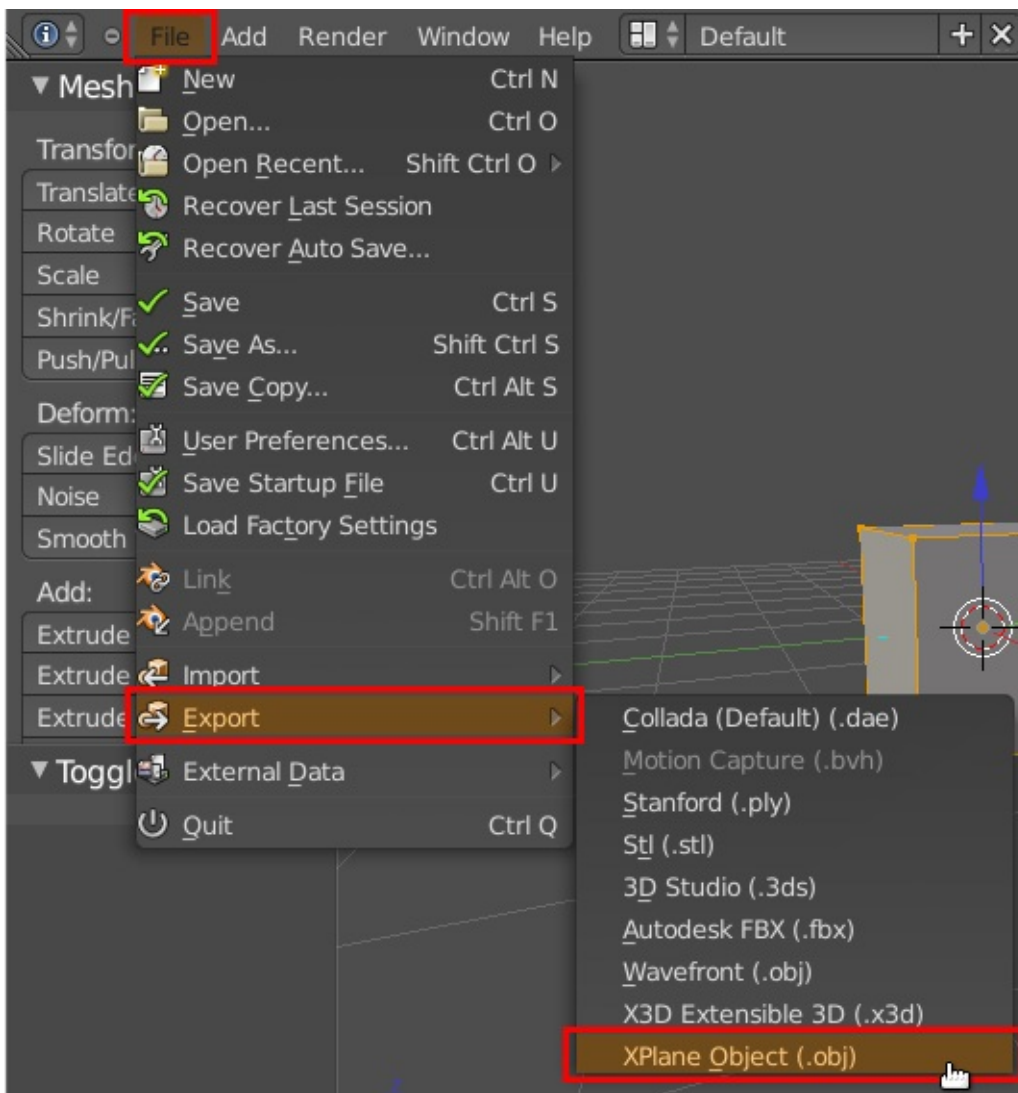
However this is not recommended as it means a performance loss in the game and possibly wrong lighting of faces pointing backwards.

6. Check your UVs

Objects without a UV-Layer will get no texture. Be sure to [UV unwrap](#) all your objects before you export. XPlane2Blender ignores material textures. It will only apply the texture to the OBJ file, you've chosen in the *Scene > XPlane* panel for the specific layer.

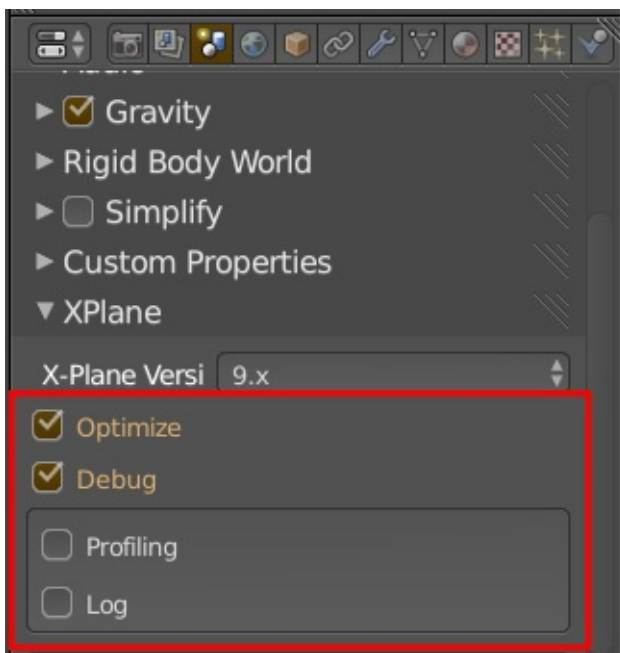
7. Export

To finally export go to *File > Export > XPlane Object (.obj)* and choose the directory you want to export all active and exportable layers to. Remember, that you can enter relative paths into the XPlane-Layer Name field. You can use that to also export the cockpit OBJ together with all other OBJs by exporting it to your airplanes *objects* folder, but setting the *Name* of the cockpit Layer to *../my_aircraft_cockpit*.



Optimize/Debug

In the properties window under the *Scene* tab in the *XPlane* panel you can find two options you can enable: Optimize and Debug.



Optimize will reduce the size of your OBJ files by removing duplicate vertices, however this can take up to several minutes depending on the total number of vertices your objects have.

When enabling **Debug** it will write debugging information like Material and Object names right into the OBJ file and print out a lot of additional information in the console.

You can then also enable **Profiling** which will create a time log about the export process (how long what took). This is actually more a feature used in development of XPlane2Blender.

Enabling **Log** will create a log file containing all debugging information within the directory of the .blend file. Enable this if you have trouble exporting, or things like animations do not show up in the simulator.

Also send these log files together with your bug reports.

If you think this is a good thing [buy me a beer](#).

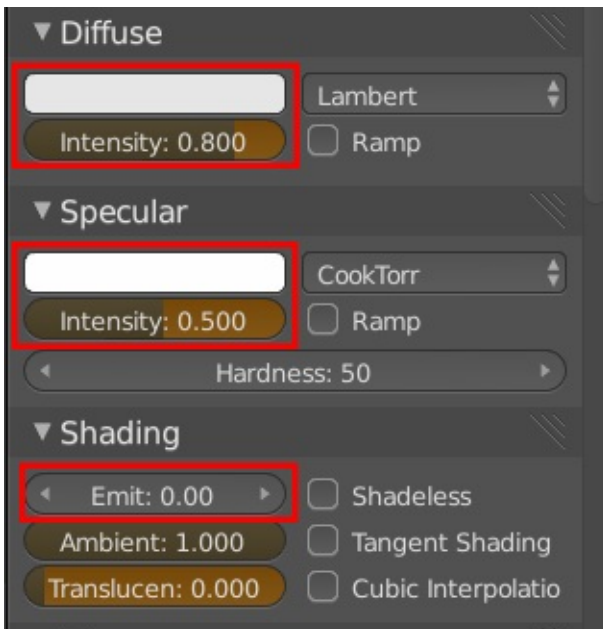
Materials

As already mentioned in the previous chapter all objects that you want to export need materials assigned to them. It will only export the first material assigned to an object.

What will be exported?

Xplane2blender can only export the following material properties to Xplane:

- diffuse color
- diffuse intensity
- specular intensity (specular size will always be 128)
- Emit value



X-plane related material properties

You have additional material properties under the Xplane panel in each material:



- **draw:** when disabled the surface will be invisible in xplane, but it will still be physically there. Use this for low poly proxy manipulator handles for example. Using high poly objects for manipulators is not recommended, as it's pretty expensive.
- **override specularity:** when enabled you can set specularity values above 1.0 with the shiny ratio to simulate very glossy surfaces.
- **use alpha cutoff:** If enabled transparent parts of the texture that are below a certain alpha value will be cut away/hidden and all other parts will be fully opaque.
- **surface type:** The material can represent a physical surface of some kind. This is especially useful for Scenery objects.
- **camera collision:** If checked, the objects having the material assigned will collide with the camera. Use this to restrict camera movement in the cockpit.
- **polygon offset:** This is useful for scenery objects that lie on top of other geometry like a runway for example. In that case you can set the polygon offset to a value greater than 0 to force it to be drawn on top of the underlying geometry.
- **override light level:** If checked you can override the amount/brightness of the night texture to be used for the material when a certain dataref reaches values. Value 1 tells

the sim to set the light level to 0 when the dataref has that value. Value 2 sets the light level to 1 (full) when the dataref has that value. The light level will interpolate between value 1 and value 2. You can use this to for example create glowing buttons or custom cockpit lighting.

- **part of cockpit panel:** if checked objects with the material will be treated as part of the cockpit panel texture. For historical reasons you still can select which one of 4 regions the material should use, however as that was a workaround for very old hardware you can leave it at "none" to use the full panel texture in the material. If you want to use single regions you need to set them in the X-Plane Scene settings for the particular X-Plane layer.
- **custom properties:** You can add any additional property to the material that will be written to the OBJ file. This is usefull to for example include properties in an OBJ file that XPlane2Blender currently does not support, but you don't want to edit the OBJ file after every export to implement these new properties. A property has the following settings:
 - name** - the name of the property, **value** - the value of the property, **reset** - another property reseting/disabling the given property, this is important because otherwise all things that are exported after this material will still have the given property.

Textures

The textures you assign to a material will not be used for export.

This has a very simple reason:

As X-Plane's OBJ files only support the use of **one texture per OBJ file** and XPlane2Blender exports **one layer as a single OBJ file**, we define the texture to be used for an OBJ file in the X-Plane scene settings for the individual layers.

Setting texture(s) for a layer

[See the previous chapter for how to assign a texture to a X-Plane layer.](#)

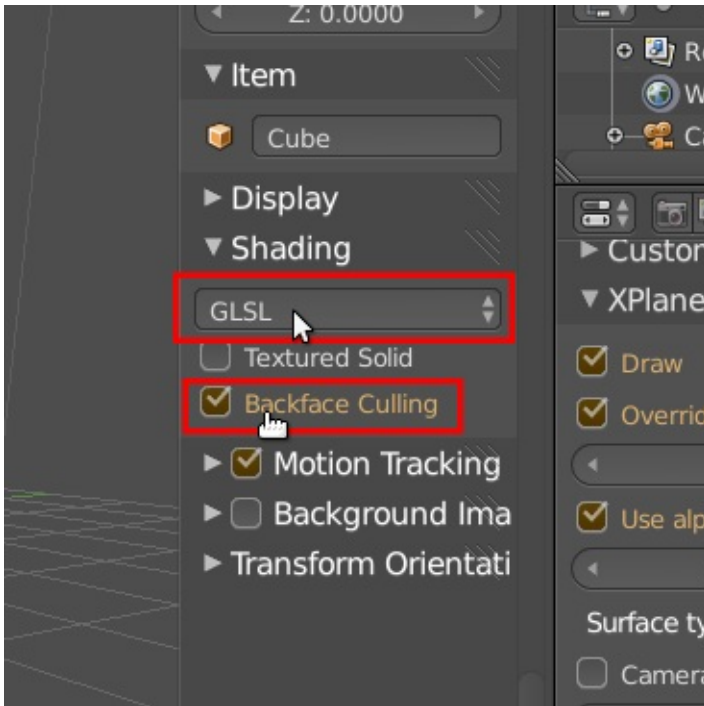
The texture(s) will only be visible on objects that are UV unwrapped. To learn how to UV unwrap in Blender read the following pages:

<http://wiki.blender.org/index.php/Doc:2.6/Manual/Textures/Mapping/UV>

Getting a live preview for the in-game textures

To get a direct feedback of how a texture will look like in X-Plane within the Blender viewport you can do the following:

1. In the viewport open the Properties shelf by hitting "N"
2. Scroll to the panel labeled **Shading**
3. Set the value of the dropdown to **GLSL**. This will display the textured view mode of the viewport with advanced OpenGL shading called GLSL, which is very close to the shading within X-Plane with "per pixel lighting" turned on.
4. Enable "Backface Culling" to hide faces facing backwards as X-Plane does.
5. Set the viewport shading to **Texture** by hitting ALT+Z
6. Now you need to assign the correct textures to all materials you want to preview



Texture preview for the default texture / day texture

1. Add a texture of type "Image or Movie" to the material you want to preview in the viewport

2. Select the texture file you've set as the Default / Day texture as the image
3. Under **Mapping** select **UV** as Coordinates and keep Projection at "Flat".
4. Under **influence** check **Color** and set it's value to 1.0

5. Leave the **color space** at **sRGB** as this fits for most cases and will prevent color shift in the viewport

Texture preview for the normal/specular texture

1. Add a texture of type "Image or Movie" to the material you want to preview in the viewport

```

```

2. Select the texture file you've set as the normal / specular texture as the image
3. Under **Mapping** select **UV** as Coordinates and keep Projection at "Flat".
4. Under **influence** uncheck **Color** and check **Normal** and set it's value to 1.0

```

```

5. Under **Image Sampling** uncheck **Alpha: Use** and check **Normal Map** and as normal map type select **Tangent**
6. Leave the **color space** at **sRGB** as this fits for most cases and will prevent color shift in the viewport

```

```

Texture preview for the specular texture

In X-Plane the normal textures alpha channel is used as the specular texture defining the specular intensity. In Blender we cannot use the Alpha channel like that and thus have to extract the alpha channel from the normal / specular texture as a grayscale image. You can do this using your Image-Editor or also using Blenders compositor nodes. However I guess you will probably already have a grayscale specular image and actually need to manually set it as alpha channel for the normal map.

1. Add a texture of type "Image or Movie" to the material you want to preview in the viewport
2. Select the grayscale image with the specular texture you've used for the alpha channel in the normal / specular texture
3. Under **Mapping** select **UV** as Coordinates and keep Projection at "Flat".
4. Under **influence** uncheck **Color** and check **Specular Intensity** and set it's value to 1.0
5. Leave the **color space** at **sRGB** as this fits for most cases and will prevent color shift in the viewport

Some Material / Texturing tips

One material, multiple OBJ textures

If you use the same material within layers/OBJ files that do not use the same texture and still want to preview both textures in the viewport you need to duplicate the material for the other layer. I've found it best practise to prefix the duplicates with the OBJ file name. For example fuse_chrome and wings_chrome.

UV unwrapping of multiple objects for a single texture

As each OBJ file can only contain one texture but consists of multiple individual objects that all share this texture one needs to unwrap all those objects in a way so their UV coordinates do not overlap. Blender has a useful feature that makes this easier.

1. To make this work you need to set the same Image in the UV/Image-Editor in Edit-Mode for every object in the layer
2. Select all objects within the layer and then additionally select the object you want to UV unwrap. This way the object will be the **active** one.
3. Go into edit-mode and in the UV/Image-Editor under **View** check **Draw other objects**. This will display the UVs of all other selected objects that have the same image assigned in the UV/Image-Editor together with the UVs of the currently edited object.

Texturing workflow

Sometimes its hard to get a texture look realistic just by painting it in your Image-Editor. Things can easily look distorted due to the UV distortion, baked in shading like ambient occlusion can look artificial if you simply paint it by hand, baked in reflections can look artificial.

To overcome this problem you can aim for a texturing workflow that makes use of Blenders texture baking feature and compositor nodes.

In general the workflow looks like this:

1. Create duplicates of the UV unwrapped low-poly in-game objects within one layer you want to texture and move them to a layer that is not used for OBJ export
2. You can now add subsurface modifiers or multires modifiers to the objects to add additional detail / smoothness to the duplicates. Subsurf can be controlled by edge crease values. The important thing is to preserve the UV mapping in the duplicates.
3. You now can add multiple materials with multiple textures to the duplicates and make use of Blenders internal renderer to create realistic looking surfaces for your duplicates
4. Finally you render bake the objects to a single texture by selecting the duplicate and in the render properties under **Bake** select **Full render**, uncheck **Selected to Active** and hit **Bake**. For more information about this read the following pages:

<http://wiki.blender.org/index.php/Doc:2.6/Manual/Render/Bake>

5. To bake normal maps you need to bake from **Selected to Active** from you high poly duplicate to the low-poly in-game object or you can take my preset at BlendSwap for normal map baking without the common problems:

<http://www.blendswap.com/blends/node-setups/tangent-space-normal-map-baking-from-two-object-space-normal-maps/>

There is a lot of try and error with render baking and you will need to fine adjust your materials allot, but basically it will make your texturing work look more realistic and easier.

Some tips for render baking

- mirror reflections work best if you set Fresnel to 0.0 and the Reflectivity higher then you would in a regular render
- for better outer reflections add an image texture containing an environment map containing a simple sky and ground and set its mapping to **Reflection**, its influence to **Color** and the Blend to **Screen**.
- render baking is way slower then regular rendering, so try to avoid the expensive mirror reflections with a gloss amount lower then 1.0
- render baking does not use anti-aliasing! That's why you need higher sample rates for ambient occlusion, Area-Lights, Ray-traced soft shadows and things like mirror reflections. It's best to post-process the render baked images with a image-based antialiasing in your Image-Editor to reduce noise in the texture and to be able to reduce sample rates during render baking.

If you think this is a good thing [buy me a beer](#).

Animations in X-Plane are all driven by datarefs. The numeric value a dataref currently has will be linearly mapped on a previously defined animation sequence.

For example if you animate your yoke from back to forth you must bind this animation to the dataref *sim/joystick/yoke_pitch_ratio* which can have values from -1.0 (yoke full forward) and 1.0 (yoke full backward). Your animation-to-dataref-binding has to take this into account and map the keyframe for the yoke fully backward and fully forward to the corresponding dataref values of -1.0 and 1.0.

Exportable animations

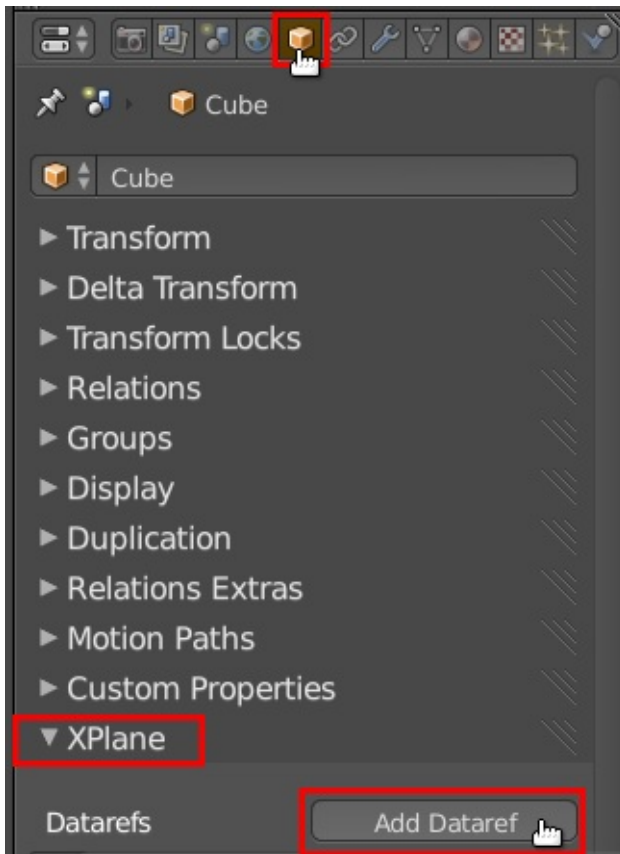
You **can** only export **Object and Bone** animations for **location and rotation**. You **cannot** export **scale** animation.

Object animation

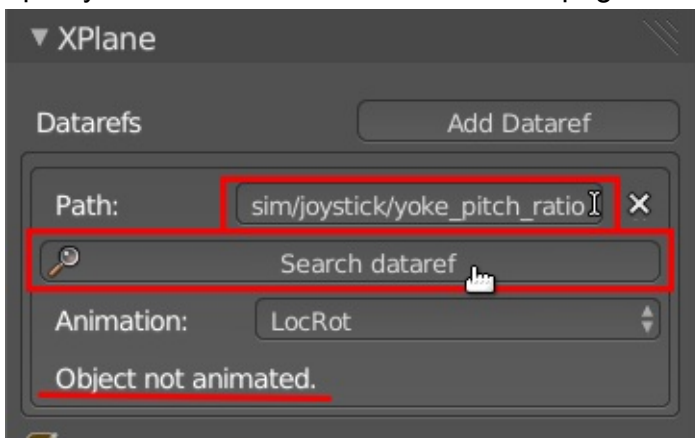
To export an objects animation:

If the object is not yet animated, animate it. You can animate the object in any way, not just with keyframes.

1. Go to the **XPlane** tab in the Object-Properties.
2. Under *Datarefs* hit **Add dataref** to add a new dataref binding.

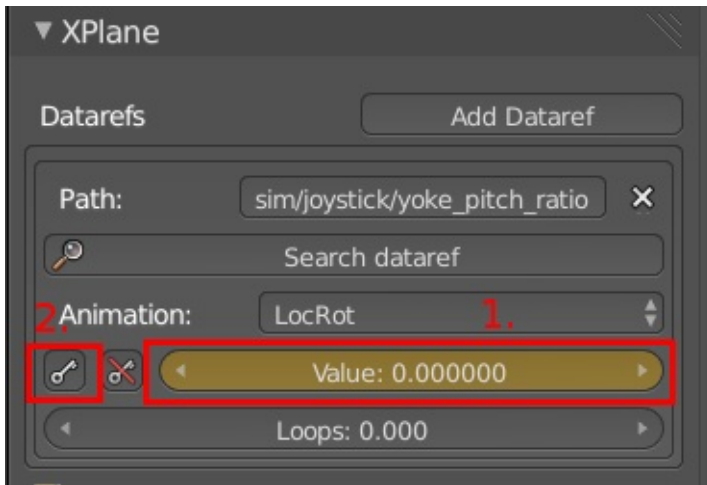


3. Enter the **Path** to the dateref. If you don't know the path hit **Search dateref**. This will open your browser with a dateref search page.

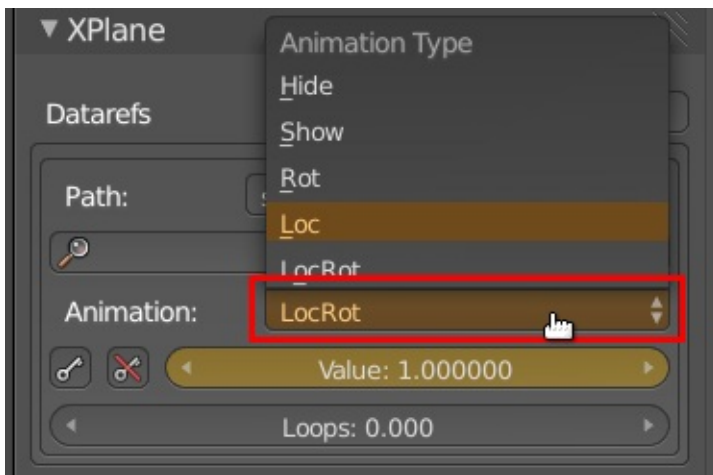


If the object is not yet animated animate it now and continue with the next steps.

4. Go to a frame you want to export as a keyframe to X-Plane.
5. Now enter the **Value** of the dateref and hit the **key-icon** next to the Value-input to create a dateref-keyframe. The dateref value will turn green indicating a keyframe. When the dateref in X-Plane reaches this value, the animation will be exactly at this keyframe.

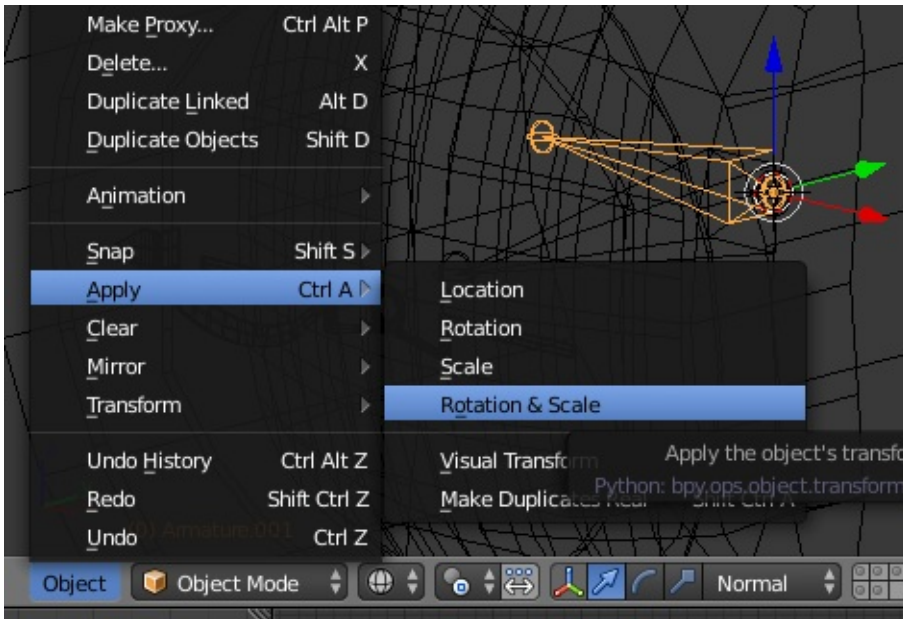


6. Go to the next frame you want to export.
7. Enter another **Value** of the dataref and hit the key-icon again to create another dataref-keyframe. The dataref value will turn green again and yellow in all frames between two dataref-keyframes. The animation of the object will now be interpolated between these two keyframes.
8. Repeat step 5 for each frame you want to export.
9. If your animation only contains location changes change it's type to "Loc" under **Animation**, if it only contains rotation change it to **Rot**. If it contains both keep it at "LocRot". If it is a hide/show animation see "[Show / Hide animations](#)" below.

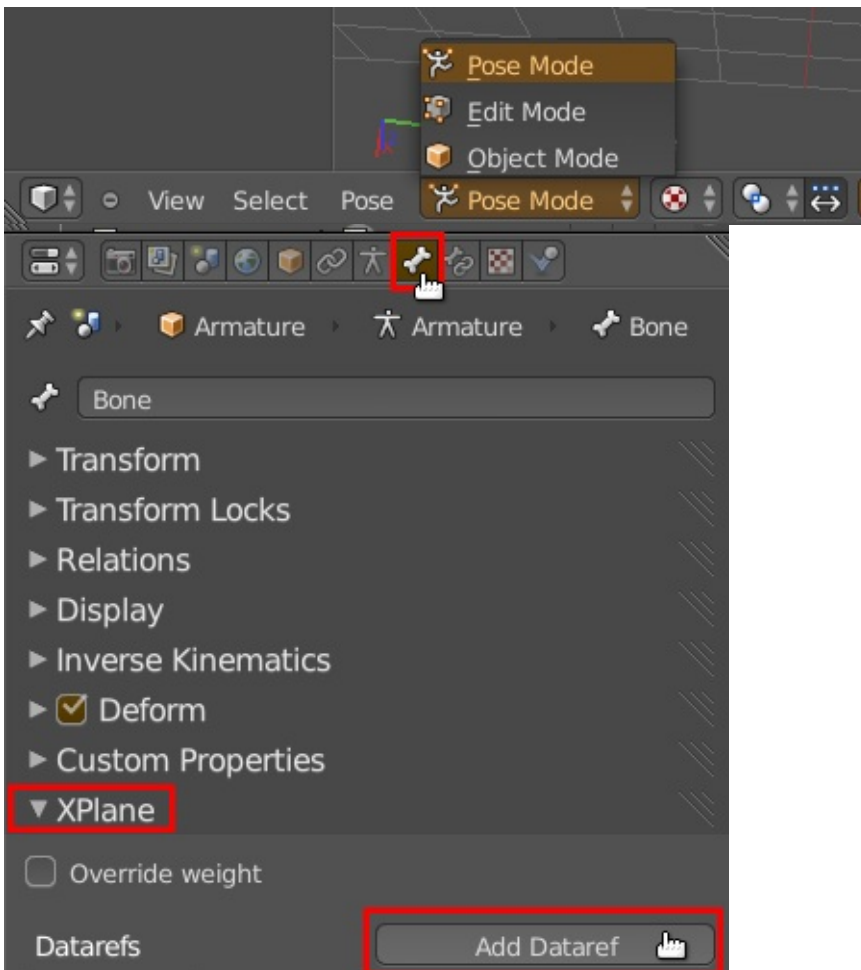


Bone animation

Important: Before parenting the Bone to the Object you want to animate, make sure you applied **Rotation & Scale** both to the Object and the Armature (in Object mode)!



Animating bones is working almost the same as animating objects. The only difference is that you have to switch to the **Bone-Properties** while in **Pose-Mode** and there you can add datarefs to the Bone.



Note: The key-framed dataref value does not get colored yellow or green as with object-datarefs. This is due to a long lasting Blender bug.

Attention: If you add Datarefs in the Object-Properties you will actually be animating the **Armature** Object which holds the bones.

Gotchas

Animation export can become wacky in some special cases:

1. Rotations larger or equal to 180° will mostly lead to a rotation below 180° .
To prevent this insert another keyframe at 90° for example.
2. Rotations along an arbitrary axis (not the global X,Y or Z axis) can look flipped.
There is no real solution to this problem which exists since the first release of XPlane2Blender 3.20. Sometimes it's helpful to "nest" the problematic object into another animated object.
3. Nested animations only work if every object in the nesting chain is animated as described in "Object animations".
4. Smooth animations in Blender appear clunky or robotic in X-Plane.
Blender by default uses Bezier-Curves to interpolate animation between keyframes. X-Plane uses linear-interpolation. To make the animation look smoother in X-Plane you need to add additional dataref-keyframes. However you will never reach the same smoothness as in Blender even when you keyframe every frame.
5. Animation keyframes should always start on the first frame, with the dataref value 0, otherwise animation can end up unwanted positions/result
6. Rotations with negative degrees in the keyframes that go above 180° along the global Y-Axis can start turning in the wrong direction. As a workaround for this bug try to avoid negative rotations above 180° by setting the objects axis in such a way that the rotation can become positive again.

Show / Hide animations

If you set the **Animation**-type of a Dataref to "Show" or "Hide" the object will be showed or hidden when the dataref reaches a certain value. Show/Hide Animations do not need the object to be animated.

Show/Hide animations require two values. The show or hide will take effect between these two values.

You'll mostly need the dataref twice to make it appear and disappear correctly. For example if your dataref value ranges from 0.0 to 1.0 and you want your object to be hidden at 0.0 and visible at 1.0 you need to add the dataref twice. The first one needs a "Hide" animation with it's first value 0.0 and second value 1.0. The second dataref needs a "Show" animation with it's first value 1.0 and second value 1.0 or higher.

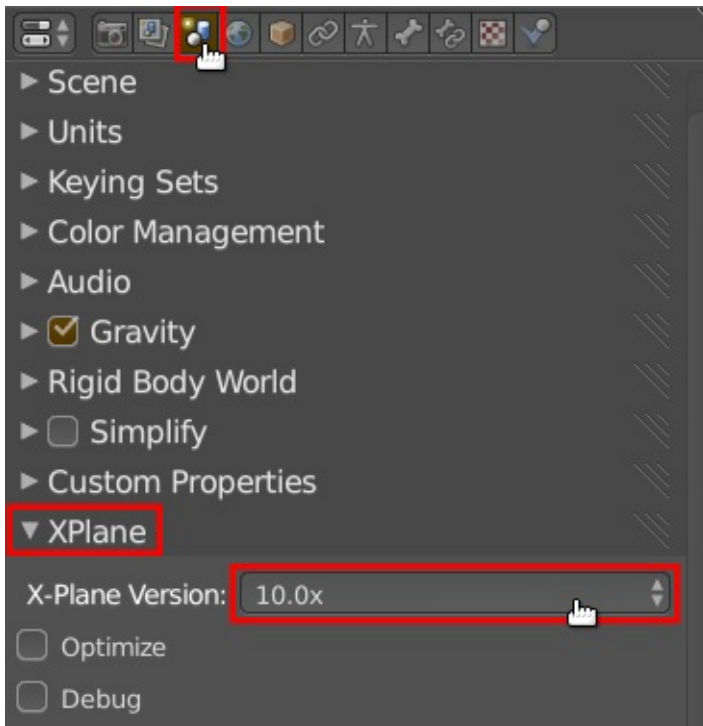
If you think this is a good thing [buy me a beer](#).

Conditions within OBJ files are supported since X-Plane 10 only. However OBJ files containing conditions also work in X-Plane 9 but simply get ignored.

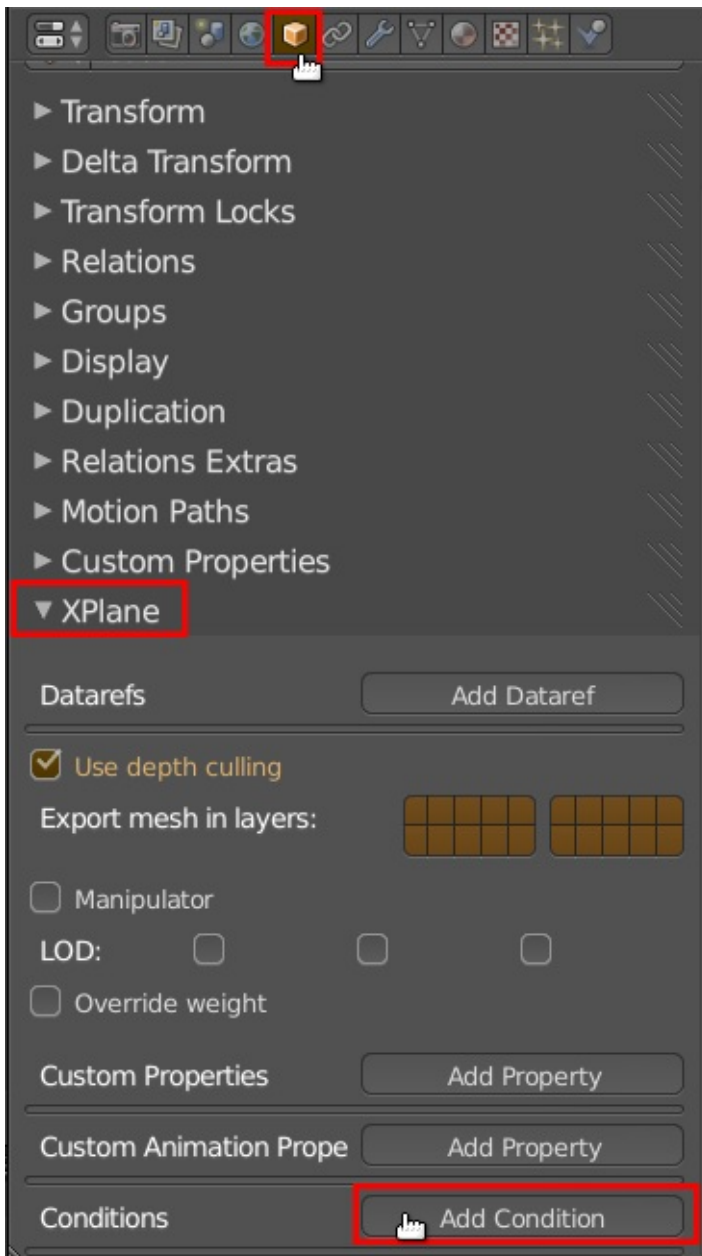
X-Plane 10 supports the conditionalization of OBJ files. Conditionalization is a process whereby the contents of a file are skipped or used depending on rendering settings. Typical uses of conditionalization are to change the appearance of an object when shadows are enabled, or when HDR is on.

To make use of conditions in XPlane2Blender do the following:

1. In the Scene tab under XPlane set the X-Plane version to at least 10.0x



2. **To conditionally hide/show an object:** Select an object and within the Object tab under XPlane click on **"Add Condition"**.



To conditionally hide/show all object with a material: Select an object and within the Material tab under XPlane click on **"Add Condition"**.

3. Now select a condition from the dropdown and choose if this condition should be true or not. The object(s) will be only visible if the condition is met.

If you think this is a good thing [buy me a beer](#).

- In my X-Plane layer I've pointed to the right texture, but the exported object contains a wrong texture path like ".././../...texture.png".
- I have a mesh that is transparent (glass). To correctly display it, it must be drawn at last. That means it needs to be written last in the OBJ file.
- How do I add cockpit regions?
- XPlane2Blender does not support feature X from the OBJ-file format specification and I need to write my *hacks* into the OBJ-file after every export.
- I've assigned a texture to my materials but I can't see it in Plane-Maker or X-Plane.
- I have an aileron/rudder/... that need's to be rotated along an arbitrary axis. How can I do that?
- How can I quickly see if my animation exported correctly?
- I've opened files created with an older version of XPlane2Blender and now (all) manipulators have the wrong type.
- My objects look very dark when brought into X-Plane.

In my X-Plane layer I've pointed to the right texture, but the exported object contains a wrong texture path like ".././../...texture.png".

This is an issue with older versions of XPlane2Blender. Be sure to uncheck "Relative Paths" when choosing the texture file. XPlane2Blender was only capable to resolve **absolute** file paths back then.

I have a mesh that is transparent (glass). To correctly display it, it must be drawn at last. That means it needs to be written last in the OBJ file.

You can override the *weight* of each object, material or bone. The *heavier* it is, the farer to the end of the OBJ-file it drops. Glass objects however should be within their own OBJ-file and marked as "glass" in Plane-Maker.

How do I add cockpit regions?

Simply said, don't do that. It is deprecated.

XPlane2Blender does not support feature X from the OBJ-file format specification and I need to write my *hacks* into the OBJ-file after every export.

You can attach any attribute to any object (including lights and bones and also materials) within Blender under X-Plane > Custom Attributes. This will save you from writing those into the exported obj file every time.

I've assigned a texture to my materials but I can't see it in Plane-Maker or X-Plane.

You have to choose your textures from within the X-Plane layer settings under Scene > XPlane. Textures assigned to materials are not taken into account. Also be sure to UV-unwrap your objects. Generated UV-coordinates are not exported.

I have an aileron/rudder/... that need's to be rotated along an arbitrary axis. How can I do that?

Create an armature with a single bone, then edit the bones position to match the axis. Then parent the object to the bone (not the armature), animate the bone in pose mode and do not forgett to give it a dataref and your object will be rotated along the bones axis.

How can I quickly see if my animation exported correctly?

Planemaker shows all animations, no matter if the dataref exists or not. This is a good way of testing if an animation looks right or got exported at all. However choosing the correct dataref is another thing.

I've opened files created with an older version of XPlane2Blender and now (all) manipulators have the wrong type.

If you created your files with the version 3.20.6 [use this script](#) to fix your manipulators.

If you've created your files with a version older then 3.20.6 [use this script](#) to fix your manipulators.

Usage: select the objects with the wrong manipulator type and then run the script within blender's text-editor.

My objects look very dark when brought into X-Plane.

In the Material tab, set the diffuse intensity of the material to 1.0 and the diffuse color to a full white. For X-Plane the diffuse color will be multiplied with the intesity and the resulting diffuse color will be multiplied with the texture color.

Example: a red diffuse color of $R = 1.0$, $G = 0.0$, $B = 0.0$ with an intensity of 0.8 will result in a X-Plane material color of $R = 0.8$, $G = 0.0$ and $B = 0.0$. A fully white pixel in the texture will then appear in exactly this color.

This behaviour is however usefull to create colored variations of objects using the same texture.

Also check that the "Emit" value under "Shading" is set to 0.0 to not make the object "glow" in X-Plane.

Note to Cycles: If you're using Cycles to bake the textures, don't forget to switch back to "Blender Render" and assign non-cycles materials to all the objects.

If you think this is a good thing [buy me a beer](#).

Documentation for XPlane2Blender v3.3

Installation - Detailed Notes

Download Blender App

Go to <https://www.blender.org/download/>

download .zip for OS X or .msi for Windows

OSX: unzip downloaded zip file. Copy entire unzipped folder into "Applications"

Set up Blender for scripts

1. In the Blender folder, create a folder named "scripts"
2. In that folder, make another folder named "addons"
3. Launch Blender
4. Pick File -> User Preferences
5. Go to "File" tab in preferences dialog box.
6. Click the folder icon next to the line that says scripts.
7. Navigate to your scripts folder.
8. When done you should see something like /Applications/blender-2.76b-OSX_10.6-x86_64/scripts/ in the line.
9. Click "save user settings"
10. Close the preferences dialog box.
11. Quit Blender

Power user note: the scripts folder can be anywhere on your hard drive and have any name. However, the "addons" folder within scripts *must* be named addons all lower-case, no spaces.

Installing X-Plane2blender

1. Download the latest version of X-Plane2Blender. **Open Issue: where do we download it from? This link needs to be finalized.**
2. Unzip the download.
3. Move the *ioxplane2blender_*into your "addons" folder you created.
4. Start Blender.
5. Pick File->User Preferences

6. Go to the "Add-Ons" tab
7. Type "xplane" into the search field (no hyphen). You will see only one add-on: Import-Export: Import/Export: X-Plane
8. Check the checkbox next to the icon of the dancing person on the right. The exporter is now enabled.
9. Click "save user settings".
10. Close the preferences dialog box.

Customizing Blender

Preferences to change

From File->Preferences, a few settings I like to customize to make Blender easier to use:

- Interface Tab: check "Zoom To Mouse Position." This lets you zoom in and out of the thing you are pointing at.
- Input: select with: pick left.

Cleaning out the Startup Configuration

Blender has a "startup file" - a file that is loaded every time you ask for a new blank canvas. By default it has a light, a cube, and a camera. I find that annoying and always have to delete them to get going. We can *permanently* delete them by:

- For each item in the scene, right-click it in the outliner and pick "delete".
- Pick File -> Save Startu File

Blender will do something you have never seen before: it will create a tiny popup menu under your mouse that says: Ok? Save Startup File

- If you click "Save Startup File" the operation is confirmed and the save happens.
- If you move the mouse away, the operation is canceled.

This kind of "menu-based confirm" is all over Blender. It is tricky to get used to at first, but can be very fast when working.

Customizing the Screen

Blender has a huge number of screens and lots of options to customize how the screen is organized.

Blender's default layout is good for general modeling, but I'm not a fan of the default UV editing view. Here'es how to customize the view for texturing:

1. Start Blender.
2. If you are not already in the "default" screen layout, pick that.
3. Click the + sign next to the screen layout menu. A new item Default.001 appears.
4. Rename it to "texturing"
5. Put the mouse over the divide between the 3-d window and the main menu bar. A horizontal pair of arrows will appear.
6. Right click and pick "split area" - a vertical line appears.
7. Move the mouse to put the line about at the midpoint of the main window and click.
8. There are now two main editing areas.
9. In the lower left corner of the right side window there is an icon with a cube and up-and-down arrows. Pick it and select "UV/image editor"
10. Put the main screen layout back to default.
11. Pick file->save user startup file to save this change for future documents.

Materials and Textures

Materials

As already mentioned in the previous chapter all objects that you want to export need materials assigned to them. It will only export the first material assigned to an object.

What will be exported?

Xplane2blender can only export the following material properties to Xplane:

- diffuse color
- diffuse intensity
- specular intensity (specular size will always be 128)
- Emit value



X-plane related material properties

You have additional material properties under the Xplane panel in each material:



- **draw**: when disabled the surface will be invisible in xplane, but it will still be physically there. Use this for low poly proxy manipulator handles for example. Using high poly objects for manipulators is not recommended, as it's pretty expensive.
- **override specularity**: when enabled you can set specularity values above 1.0 with the shiny ratio to simulate very glossy surfaces.
- **use alpha cutoff**: If enabled transparent parts of the texture that are below a certain alpha value will be cut away/hidden and all other parts will be fully opaque.
- **surface type**: The material can represent a physical surface of some kind. This is especially usefull for Scenery objects.
- **camera collision**: If checked, the objects having the material assigned will collide with the camera. Use this to restrict camera movement in the cockpit.
- **polygon offset**: This is usefull for scenery objects that lie on top of other geometry like a runway for example. In that case you can set the polygon offset to a value greater then 0 to force it to be drawn on top the underllying geometry.
- **override light level**: If checked you can override the amount/brightness of the night texture to be used for the material when a certain dataref reaches values. Value 1 tells

the sim to set the light level to 0 when the dataref has that value. Value 2 sets the light level to 1 (full) when the dataref has that value. The light level will interpolate between value 1 and value 2. You can use this to for example create glowing buttons or custom cockpit lighting.

- **part of cockpit panel:** if checked objects with the material will be treated as part of the cockpit panel texture. For historical reasons you still can select which one of 4 regions the material should use, however as that was a workaround for very old hardware you can leave it at "none" to use the full panel texture in the material. If you want to use single regions you need to set them in the X-Plane Scene settings for the particular X-Plane layer.
- **custom properties:** You can add any additional property to the material that will be written to the OBJ file. This is usefull to for example include properties in an OBJ file that XPlane2Blender currently does not support, but you don't want to edit the OBJ file after every export to implement these new properties. A property has the following settings:
 - name** - the name of the property, **value** - the value of the property, **reset** - another property resetting/disabling the given property, this is important because otherwise all things that are exported after this material will still have the given property.

Textures

The textures you assign to a material will not be used for export.

This has a very simple reason:

As X-Plane's OBJ files only support the use of **one texture per OBJ file** and XPlane2Blender exports **one layer as a single OBJ file**, we define the texture to be used for an OBJ file in the X-Plane scene settings for the individual layers.

Setting texture(s) for a layer

[See the previous chapter for how to assign a texture to a X-Plane layer.](#)

The texture(s) will only be visible on objects that are UV unwrapped. To learn how to UV unwrap in Blender read the following pages:

<http://wiki.blender.org/index.php/Doc:2.6/Manual/Textures/Mapping/UV>

Getting a live preview for the in-game textures

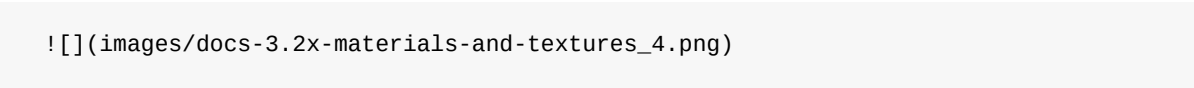
To get a direct feedback of how a texture will look like in X-Plane within the Blender viewport you can do the following:

1. In the viewport open the Properties shelf by hitting "N"
2. Scroll to the panel labeled **Shading**
3. Set the value of the dropdown to **GLSL**. This will display the textured view mode of the viewport with advanced OpenGL shading called GLSL, which is very close to the shading within X-Plane with "per pixel lighting" turned on.
4. Enable "Backface Culling" to hide faces facing backwards as X-Plane does.
5. Set the viewport shading to **Texture** by hitting ALT+Z
6. Now you need to assign the correct textures to all materials you want to preview

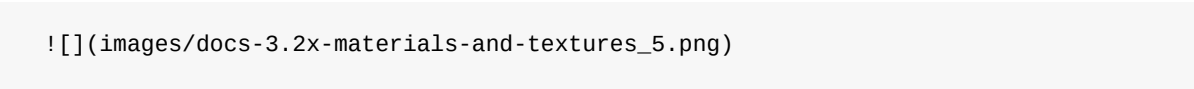


Texture preview for the default texture / day texture

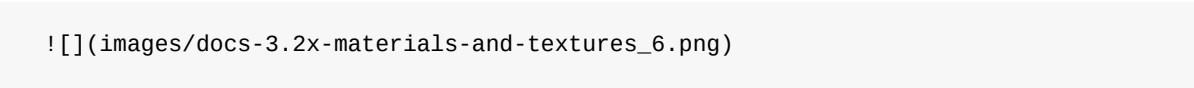
1. Add a texture of type "Image or Movie" to the material you want to preview in the viewport



2. Select the texture file you've set as the Default / Day texture as the image
3. Under **Mapping** select **UV** as Coordinates and keep Projection at "Flat".
4. Under **influence** check **Color** and set it's value to 1.0

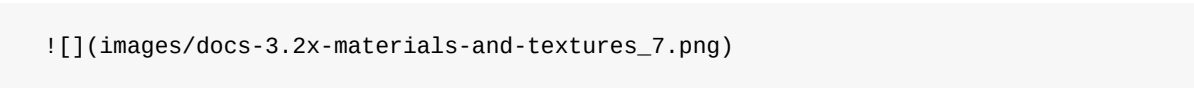


5. Leave the **color space** at **sRGB** as this fits for most cases and will prevent color shift in the viewport



Texture preview for the normal/specular texture

1. Add a texture of type "Image or Movie" to the material you want to preview in the viewport



2. Select the texture file you've set as the normal / specular texture as the image
3. Under **Mapping** select **UV** as Coordinates and keep Projection at "Flat".
4. Under **influence** uncheck **Color** and check **Normal** and set it's value to 1.0



5. Under **Image Sampling** uncheck **Alpha: Use** and check **Normal Map** and as normal

map type select **Tangent**

6. Leave the **color space** at **sRGB** as this fits for most cases and will prevent color shift in the viewport

Texture preview for the specular texture

In X-Plane the normal textures alpha channel is used as the specular texture defining the specular intensity. In Blender we cannot use the Alpha channel like that and thus have to extract the alpha channel from the normal / specular texture as a grayscale image. You can do this using your Image-Editor or also using Blenders compositor nodes. However I guess you will probably already have a grayscale specular image and actually need to manually set it as alpha channel for the normal map.

1. Add a texture of type "Image or Movie" to the material you want to preview in the viewport
2. Select the grayscale image with the specular texture you've used for the alpha channel in the normal / specular texture
3. Under **Mapping** select **UV** as Coordinates and keep Projection at "Flat".
4. Under **influence** uncheck **Color** and check **Specular Intensity** and set it's value to 1.0
5. Leave the **color space** at **sRGB** as this fits for most cases and will prevent color shift in the viewport

Some Material / Texturing tips

One material, multiple OBJ textures

If you use the same material within layers/OBJ files that do not use the same texture and still want to preview both textures in the viewport you need to duplicate the material for the other layer. I've found it best practise to prefix the duplicates with the OBJ file name. For example fuse_chrome and wings_chrome.

UV unwrapping of multiple objects for a single texture

As each OBJ file can only contain one texture but consists of multiple individual objects that all share this texture one needs to unwrap all those objects in a way so their UV coordinates do not overlap. Blender has a usefull feature that makes this easier.

1. To make this work you need to set the same Image in the UV/Image-Editor in Edit-Mode for every object in the layer

2. Select all objects within the layer and then additionally select the object you want to UV unwrap. This way the object will be the **active** one.
3. Go into edit-mode and in the UV/Image-Editor under **View** check **Draw other objects**. This will display the UVs of all other selected objects that have the same image assigned in the UV/Image-Editor together with the UVs of the currently edited object.

Texturing workflow

Sometimes its hard to get a texture look realistic just by painting it in your Image-Editor. Things can easily look distorted due to the UV distortion, baked in shading like ambient occlusion can look artificial if you simply paint it by hand, baked in reflections can look artificial.

To overcome this problem you can aim for a texturing workflow that makes use of Blenders texture baking feature and compositor nodes.

In general the workflow looks like this:

1. Create duplicates of the UV unwrapped low-poly in-game objects within one layer you want to texture and move them to a layer that is not used for OBJ export
2. You can now add subsurface modifiers or multires modifiers to the objects to add additional detail / smoothness to the duplicates. Subsurf can be controlled by edge crease values. The important thing is to preserve the UV mapping in the duplicates.
3. You now can add multiple materials with multiple textures to the duplicates and make use of Blenders internal renderer to create realistic looking surfaces for your duplicates
4. Finally you render bake the objects to a single texture by selecting the duplicate and in the render properties under **Bake** select **Full render**, uncheck **Selected to Active** and hit **Bake**. For more information about this read the following pages:

<http://wiki.blender.org/index.php/Doc:2.6/Manual/Render/Bake>

5. To bake normal maps you need to bake from **Selected to Active** from you high poly duplicate to the low-poly in-game object or you can take my preset at BlendSwap for normal map baking without the common problems:

<http://www.blendswap.com/blends/node-setups/tangent-space-normal-map-baking-from-two-object-space-normal-maps/>

There is a lot of try and error with render baking and you will need to fine adjust your materials allot, but basically it will make your texturing work look more realistic and easier.

If you think this is a good thing [buy me a beer](#).

Export of OBJ files

Please learn Blender first

If you read up to this point and never used Blender 2.6x before, please first [learn some on Blender basics](#).

Recommendation: Understand the OBJ file format

I also recommend you to [learn the concept of OBJ files](#). They are simple text files, you can investigate in any text-editor. Understanding the file format will help you in debugging your OBJ-Files and understanding XPlane2Blender better.

Steps to export an OBJ file

XPlane2Blender 3.2 uses Blenders layers for export of OBJ-files. Each layer represents one OBJ file. This way you can easily export multiple OBJ files at once and organize your OBJ files.

1. Activate X-Plane Layers

However to make the export work we need to initiate the Blender layers to make them work as OBJ-files. For this go to the *Scene* Tab in the *Properties* window. Now scroll down until you see a *XPlane* panel. Open it up, if it is not already. Hit the button *Add X-Plane layers*.



2. Configure a X-Plane layer

Now you have one subpanel for each Blender layer. Open up the first one labeled *Layer 1*.

If not enabled, check *Export* to make the layer exportable. Within the *Name* field type in the **name of the OBJ file** this layer should create, without the .obj extension. You can enter a relative path here.



In the *Textures* group you can select the texture files the OBJ should use. One OBJ file can only use one texture file.

- The "Default" texture is the day texture,
- the "Night" texture will be used as the lit/night texture and emits light depending on the texture brightness,
- the "Normal/Specular" texture will be used for tangent space normal maps where the alpha channel defines the specular intensity.

If the OBJ file is the **cockpit Object**, check *Cockpit*.

3. Advanced X-Plane layer configuration

If the OBJ will be attached to the aircraft as a slung load you can set it's weight in pounds using the *Slung load weight* slider.

If the OBJ will have levels of detail select how many in the *Levels of detail* dropdown.

Each level of detail (= **LOD**) has a near and far clipping distance. The LOD will only be visible within these distances. When dealing with multiple LODs be sure to set the Near clipping distance of the higher LOD to the Far distance of the preceding LOD as shown in the next image.

This will make your LODs compatible with X-Plane 9, as only X-Plane 10 allows for overlapping LODs.

You can add additional OBJ-Header Properties by hitting the *Add Property* button.

4. Check your Materials

XPlane2Blender will not export objects without a material. So check if the objects within that layer all have a material assigned.

5. Check your normals

By default all objects exported to X-Plane are rendered with "single-sided" faces. This means that only the front side of a face will be visible. If the face is facing backwards it will simply be hidden. To be sure all faces point to the correct side you have to make the face

normals of each object point outwards.

You can view the face normals of an object when in *Edit-Mode*. In the *View Properties* (hit N while hovering over the viewport) under *Mesh Display* activate the the display of face normals as lines.



If the lines are too short increase the *Size* next to this button.

The other option is to make all faces of an object "double-sided". This can be done in the mesh tab in the properties window.



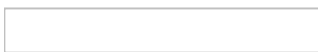
However this is not recommended as it means a performance loss in the game and possibly wrong lighting of faces pointing backwards.

6. Check your UVs

Objects without a UV-Layer will get no texture. Be sure to [UV unwrap](#) all your objects before you export. XPlane2Blender ignores material textures. It will only apply the texture to the OBJ file, you've chosen in the *Scene > XPlane* panel for the specific layer.

7. Export

To finally export go to *File > Export > XPlane Object (.obj)* and choose the directory you want to export all active and exportable layers to. Remember, that you can enter relative paths into the XPlane-Layer Name field. You can use that to also export the cockpit OBJ together with all other OBJs by exporting it to your airplanes *objects* folder, but setting the *Name* of the cockpit Layer to *../my_aircraft_cockpit*.



Optimize/Debug

In the properties window under the *Scene* tab in the *XPlane* panel you can find two options you can enable: *Optimize* and *Debug*.



Optimize will reduce the size of your OBJ files by removing duplicate vertices, however this can take up to several minutes depending on the total number of vertices your objects have.

When enabling **Debug** it will write debugging information like Material and Object names right into the OBJ file and print out a lot of additional information in the console.

You can then also enable **Profiling** which will create a time log about the export process (how long what took). This is actually more a feature used in development of XPlane2Blender.

Enabling **Log** will create a log file containing all debugging information within the directory of the .blend file. Enable this if you have trouble exporting, or things like animations do not show up in the simulator.

Also send these log files together with your bug reports.

If you think this is a good thing [buy me a beer](#).

Animations

Animations in X-Plane are all driven by datarefs. The numeric value a dataref currently has will be linearly mapped on a previously defined animation sequence.

For example if you animate your yoke from back to forth you must bind this animation to the dataref *sim/joystick/yoke_pitch_ratio* which can have values from -1.0 (yoke full forward) and 1.0 (yoke full backward). Your animation-to-dataref-binding has to take this into account and map the keyframe for the yoke fully backward and fully forward to the corresponding dataref values of -1.0 and 1.0.

Exportable animations

You **can** only export **Object and Bone** animations for **location and rotation**. You **cannot** export **scale** animation.

Object animation

To export an objects animation:

If the object is not yet animated, animate it. You can animate the object in any way, not just with keyframes.

1. Go to the **XPlane** tab in the Object-Properties.
2. Under *Datarefs* hit **Add dataref** to add a new dataref binding.
3. Enter the **Path** to the dataref. If you don't know the path hit **Search dataref**. This will open your browser with a dataref search page.

If the object is not yet animated animate it now and continue with the next steps.

4. Go to a frame you want to export as a keyframe to X-Plane.
5. Now enter the **Value** of the dataref and hit the **key-icon** next to the Value-input to create a dataref-keyframe. The dataref value will turn green indicating a keyframe. When the dataref in X-Plane reaches this value, the animation will be exactly at this keyframe.

6. Go to the next frame you want to export.
7. Enter another **Value** of the dataref and hit the key-icon again to create another dataref-keyframe. The dataref value will turn green again and yellow in all frames between two

dataref-keyframes. The animation of the object will now be interpolated between these two keyframes.

8. Repeat step 5 for each frame you want to export.
9. If your animation only contains location changes change it's type to "Loc" under **Animation**, if it only contains rotation change it to **Rot**. If it contains both keep it at "LocRot". If it is a hide/show animation [see "Show / Hide animations" below](#).



Bone animation

Important: Before parenting the Bone to the Object you want to animate, make sure you applied **Rotation & Scale** both to the Object and the Armature (in Object mode)!

Animating bones is working almost the same as animating objects. The only difference is that you have to switch to the **Bone-Properties** while in **Pose-Mode** and there you can add datarefs to the Bone.



Note: The key-framed dataref value does not get colored yellow or green as with object-datarefs. This is due to a long lasting Blender bug.

Attention: If you add Datarefs in the Object-Properties you will actually be animating the **Armature** Object which holds the bones.

Gotchas

Animation export can become wacky in some special cases:

1. Rotations larger or equal to 180° will mostly lead to a rotation below 180°. To prevent this insert another keyframe at 90° for example.
2. Rotations along an arbitrary axis (not the global X,Y or Z axis) can look flipped. There is no real solution to this problem which exists since the first release of XPlane2Blender 3.20. Sometimes it's helpfull to "nest" the problematic object into another animated object.
3. Nested animations only work if every object in the nesting chain is animated as described in "Object animations".
4. Smooth animations in Blender appear clunky or robotic in X-Plane. Blender by default uses Bezier-Curves to interpolate animation between keyframes. X-Plane uses linear-interpolation. To make the animation look smoother in X-Plane you

need to add additional dataref-keyframes. However you will never reach the same smoothness as in Blender even when you keyframe every frame.

5. Animation keyframes should always start on the first frame, with the dataref value 0, otherwise animation can end up unwanted positions/result
6. Rotations with negative degrees in the keyframes that go above 180° along the global Y-Axis can start turning in the wrong direction. As a workaround for this bug try to avoid negative rotations above 180° by setting the objects axis in such a way that the rotation can become positive again.

Show / Hide animations

If you set the **Animation**-type of a Dataref to "Show" or "Hide" the object will be showed or hidden when the dataref reaches a certain value. Show/Hide Animations do not need the object to be animated.

Show/Hide animations require two values. The show or hide will take effect between these two values.

You'll mostly need the dataref twice to make it appear and disappear correctly. For example if your dataref value ranges from 0.0 to 1.0 and you want your object to be hidden at 0.0 and visible at 1.0 you need to add the dataref twice. The first one needs a "Hide" animation with it's first value 0.0 and second value 1.0. The second dataref needs a "Show" animation with it's first value 1.0 and second value 1.0 or higher.

If you think this is a good thing [buy me a beer](#).

Conditions

Conditions within OBJ files are supported since X-Plane 10 only. However OBJ files containing conditions also work in X-Plane 9 but simply get ignored.

X-Plane 10 supports the conditionalization of OBJ files. Conditionalization is a process whereby the contents of a file are skipped or used depending on rendering settings. Typical uses of conditionalization are to change the appearance of an object when shadows are enabled, or when HDR is on.

To make use of conditions in XPlane2Blender do the following:

1. In the Scene tab under XPlane set the X-Plane version to at least 10.0x
2. **To conditionally hide/show an object:** Select an object and within the Object tab under XPlane click on "**Add Condition**".

To conditionally hide/show all object with a material: Select an object and within the Material tab under XPlane click on "**Add Condition**".

3. Now select a condition from the dropdown and choose if this condition should be true or not. The object(s) will be only visible if the condition is met.

If you think this is a good thing [buy me a beer](#).

Advanced Techniques

Some tips for render baking

- mirror reflections work best if you set Fresnel to 0.0 and the Reflectivity higher then you would in a regular render
- for better outer reflections add an image texture containing an environment map containing a simple sky and ground and set its mapping to **Reflection**, its influence to **Color** and the Blend to **Screen**.
- render baking is way slower then regular rendering, so try to avoid the expensive mirror reflections with a gloss amount lower then 1.0
- render baking does not use anti-aliasing! That's why you need higher sample rates for ambient occlusion, Area-Lights, Ray-traced soft shadows and things like mirror reflections. It's best to post-process the render baked images with a image-based antialiasing in your Image-Editor to reduce noise in the texture and to be able to reduce sample rates during render baking.

FAQ

- In my X-Plane layer I've pointed to the right texture, but the exported object contains a wrong texture path like `"../../../../...texture.png"`.
- I have a mesh that is transparent (glass). To correctly display it, it must be drawn at last. That means it needs to be written last in the OBJ file.
- How do I add cockpit regions?
- XPlane2Blender does not support feature X from the OBJ-file format specification and I need to write my *hacks* into the OBJ-file after every export.
- I've assigned a texture to my materials but I can't see it in Plane-Maker or X-Plane.
- I have an aileron/rudder/... that need's to be rotated along an arbitrary axis. How can I do that?
- How can I quickly see if my animation exported correctly?
- I've opened files created with an older version of XPlane2Blender and now (all) manipulators have the wrong type.
- My objects look very dark when brought into X-Plane.

In my X-Plane layer I've pointed to the right texture, but the exported object contains a wrong texture path like `"../../../../...texture.png"`.

This is an issue with older versions of XPlane2Blender. Be sure to uncheck "Relative Paths" when choosing the texture file. XPlane2Blender was only capable to resolve **absolute** file paths back then.

I have a mesh that is transparent (glass). To correctly display it, it must be drawn at last. That means it needs to be written last in the OBJ file.

You can override the *weight* of each object, material or bone. The *heavier* it is, the farer to the end of the OBJ-file it drops. Glass objects however should be within their own OBJ-file and marked as "glass" in Plane-Maker.

How do I add cockpit regions?

Simply said, don't do that. It is deprecated.

XPlane2Blender does not support feature X from the OBJ-file format specification and I need to write my *hacks* into the OBJ-file after every export.

You can attach any attribute to any object (including lights and bones and also materials) within Blender under X-Plane > Custom Attributes. This will save you from writing those into the exported obj file every time.

I've assigned a texture to my materials but I can't see it in Plane-Maker or X-Plane.

You have to choose your textures from within the X-Plane layer settings under Scene > XPlane. Textures assigned to materials are not taken into account. Also be sure to UV-unwrap your objects. Generated UV-coordinates are not exported.

I have an aileron/rudder/... that need's to be rotated along an arbitrary axis. How can I do that?

Create an armature with a single bone, then edit the bones position to match the axis. Then parent the object to the bone (not the armature), animate the bone in pose mode and do not forgett to give it a dataref and your object will be rotated along the bones axis.

How can I quickly see if my animation exported correctly?

Planemaker shows all animations, no matter if the dataref exists or not. This is a good way of testing if an animation looks right or got exported at all. However choosing the correct dataref is another thing.

I've opened files created with an older version of XPlane2Blender and now (all) manipulators have the wrong type.

If you created your files with the version 3.20.6 [use this script](#) to fix your manipulators.

If you've created your files with a version older then 3.20.6 [use this script](#) to fix your manipulators.

Usage: select the objects with the wrong manipulator type and then run the script within blender's text-editor.

My objects look very dark when brought into X-Plane.

In the Material tab, set the diffuse intensity of the material to 1.0 and the diffuse color to a full white. For X-Plane the diffuse color will be multiplied with the intensity and the resulting diffuse color will be multiplied with the texture color.

Example: a red diffuse color of $R = 1.0$, $G = 0.0$, $B = 0.0$ with an intensity of 0.8 will result in a X-Plane material color of $R = 0.8$, $G = 0.0$ and $B = 0.0$. A fully white pixel in the texture will then appear in exactly this color.

This behaviour is however usefull to create colored variations of objects using the same texture.

Also check that the "Emit" value under "Shading" is set to 0.0 to not make the object "glow" in X-Plane.

Note to Cycles: If you're using Cycles to bake the textures, don't forget to switch back to "Blender Render" and assign non-cycles materials to all the objects.

If you think this is a good thing [buy me a beer](#).

Documentation for XPlane2Blender v3.4

Installation

Download and Install Blender

Go to [the Blender home page](#) and download the latest version of Blender for your OS. (Note for Windows: If one would like to use multiple versions of Blender on the same computer, use the "portable" .zip version instead of the Windows Installer (.msi) version.)

OSX:

- Unzip downloaded zip file.
- Copy entire unzipped folder into "Applications"

Windows:

- Run the Windows Installer (.msi) and follow the prompts
- If using the .zip version unzip it to an easy to access location

Official Blender Documentation

- Installing on [macOS](#)
- Installing on [Windows](#)
- Installing on [Linux](#)

Download and Install XPlane2Blender

Automatic Installation

1. Download XPlane2Blender. The safest choice (see "[What Version Do I Choose?](#)") is the latest **non pre-release version**. Put it in a convenient place
2. Launch Blender and open "User Preferences" (**File->User Preferences**)
3. Select the "Add-ons" tab
4. Click the button at the bottom that says "Install from File..."
5. Use the Blender file picker to locate and pick the downloaded XPlane2Blender .zip
6. Click the checkmark next to the words "Import-Export: Export: X-Plane (.obj)". This will enable the addon

7. Click "Save User Settings" and exit the Window

Manual (Advanced) Installation

Blender recognizes that XPlane2Blender is an addon because of some files inside the folder 'io_xplane2blender' which is inside the zip file and because it gets placed into a special "addons" folder. You can manually install XPlane2Blender by dragging that unzipped `io_xplane2blender` folder into it, however it may be up to you to create the folder structure *exactly* by hand.

It is typically located (or should be located) in the following places

- macOS: `/Users/$USER/Library/Application Support/Blender/2.78/scripts/addons`
- Windows: `%USERPROFILE%\AppData\Roaming\Blender Foundation\Blender\2.78\scripts\addons`
- Linux: `$HOME/.config/blender/2.78/scripts/addons`

If it is not located in one of these places, you'll have to find it yourself. You can change the location of the scripts folder in User Preferences, however this could affect other addons. Proceed with caution.

Official Blender Documentation

- [Configuring Directories](#)

Customizing Blender

Preferences to change

From File->Preferences, a few settings I like to customize to make Blender easier to use:

- Interface Tab: check "Zoom To Mouse Position." This lets you zoom in and out of the thing you are pointing at.
- Input: Select with: pick left.

Cleaning out the Startup Configuration

Blender has a "Startup File". It is loaded everytime you select "New File". Since X-Plane has its own camera, and lighting systems you may find it to be a time saver to remove them from the default.

1. For each item in the scene, right-click it in the outliner and pick "delete".

2. Pick File -> Save Startup File (or press Ctrl + U)

A tiny popup menu will appear under your mouse that says:

``

(?) Ok?

Save Startup File CTRL + U ``

- If you click "Save Startup File" the operation is confirmed and the save happens.
- If you move the mouse away, the operation is canceled.

This kind of "menu-based confirm" is heavily used in Blender. It can be tricky to get used to at first, but can actually be very fast and natural.

Customizing the Screen

An important part of an efficient workflow is customing your screen layout. You can read more about it on Blender's manual

- [Screens Overview](#)
- [Area Splitting and Resizing](#)

XPlane2Blender Version Numbers

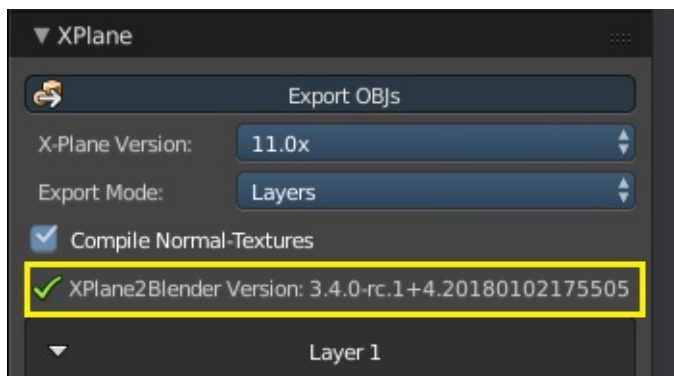
XPlane2Blender follows a fairly standard software development cycle and uses version numbers to keep track of it's progress and as metadata in .blend and .obj files to help debug problems. A fully history of every version number that a .blend file has been opened with is also kept (with restrictions).

What Version Do I Choose?

Use the [latest full release candidate from github](#) for the safest and most stable experience possible. Download the .zip file that starts with "io_xplane2blender" (not Source code.zip!) and read the [installation instructions](#).

What Version Do I Have?

The most complete place to check what version you have is at the top of the scene settings. It should look like this:



Broken down, its part's represent:

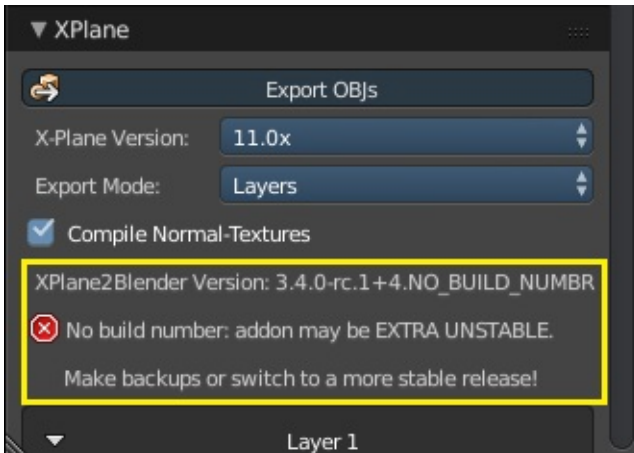
```
"3.4.0" - Major, minor, and revision number (part of any Blender Addon)
"rc" - The build type (see below)
"1" - What number release candidate this is
```

After the "+":

```
"4" - The data model version
"20180102175505" - When this build of the plugin was created (in the UTC timezone)
```

Using a build, even the latest release candidate, without a build number is dangerous!

It means any number of untracked changes could be infecting your .blend file with no dialog or record of changes! Using a build without a build number will receive a large warning and no green checkmark for release candidates.



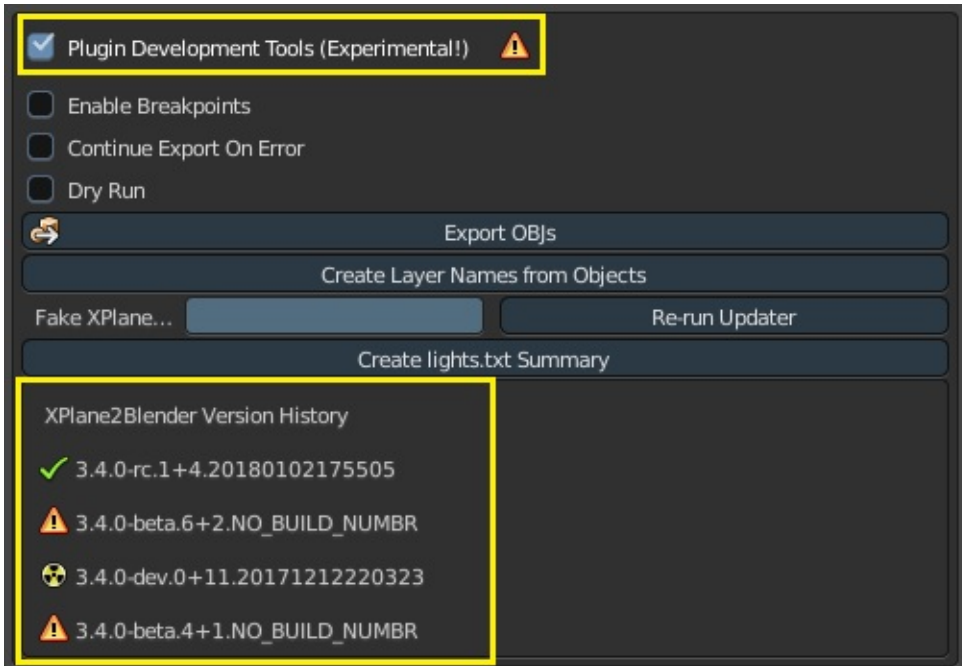
Build Types

New features and fixes are added over the course of a development cycle. A cycle is broken into segments explained in this table (organized from safest to least safe).

Build Type	Icon	State Of Addon	Dangers	Notes
Release Candidate		Contains all tested work from the beta: the most tested, stable, and documented. Rarely are new changes introduced here	As is code's nature; bugs could still exist	"Full Release" and "RC" are synonyms. There can be multiple RCs because our work is never done!
Beta		New features and fixes are being polished, tested, and integrated together	Bugs, new and old, can corrupt data	Make backups! You may have to redo or discard work. Follow development closely and communicate with us! Your voluntary participation is a huge help to XPlane2Blender!
Developer		Code is at its most malleable and unstable	Everything and anything could happen!	Only developers and users they're directly communicating with should use a dev build!

What Versions Has This .blend File Been Opened With?

By enabling the Plugin Development section in the bottom of the scene settings, you can see a history of every version of XPlane2Blender this .blend file has been used with (things get less accurate if you open a .blend file in pre-3.4.0-beta.5 versions). This is a read only list, shown in chronological order, developed to debug automatic updater problems. It is only updated when there is a difference (excluding the build number) between last and current version numbers.



What About Checking Out Code Directly From git?

This is **ABSOLUTELY NOT RECOMMENDED!** You will never have build numbers (unless you make a build yourself) and severely increase the chance of corrupting your work. If you feel the need to checkout code from git, **talk to us first** and we'll work it out your needs together!

Quick Example

Let's go through the process of making a textured cube in Blender and having it show up in X-Plane.

Launch Blender, Set Up Our Workspace

- Ensure lamps and cameras are out of the way
- Ensure scene settings are correct
- Save as Custom Scenery/TXWF/cubes/cube_tutorial.blend

Create a Basic Cube, Set Layer 1 settings

- Create a cube
- Open Scenery Panel
- Set Up Layer One Settings

Setting Material/Texture Properties

- Create Material
- Set Material Properties
- Open Image (we skip UV Mapping) Albedo
- Open Image NML

Create library.txt

Launch X-Plane!

Scene Settings

To access, navigate to Blender's Properties Pane and click on the Scene Icon [icon], and scroll to the X-Plane panel.

Basic Settings

Export Objs Button

This button will export the .blend file and save the exported .obj files relative to it. **The only difference between using this button and using the File->Export->X-Plane Object (.obj) menu option is that the menu option gives you an export dialog box to choose your export destination. Exported Objs will be the same using either method.**

X-Plane Version

Latest released version of X-Plane by default. A drop down menu with versions of X-Plane to target. The behavior of the plugin, and contents of exported OBJs may changed based on it. In addition, certain parts of the UI may only be shown under certain versions. For instance, "Blend Glass" will only appear as a Blend option when the X-Plane Version is set to `11.0x` or higher.

Export Mode

A drop down menu with two options: "Layers" and "Root Object", **"Layers" by default.** These represent how the exporter will collect blender objects for processing into OBJs.

Layers Mode

In "Layers" mode, each of Blender's [Object Layers](#) are combined with their corresponding X-Plane Layer (1st Blender layer goes with X-Plane layer 1, 2nd Blender layer goes with X-Plane layer 2, etc). The Blender layer's list of objects that are visible in it gets combined with the [X-Plane Layer's settings](#) and is exported as a complete OBJ. Non-visible Blender Layers are not combined with their corresponding X-Plane Layers and do not get exported. In "Layers Mode" the settings for each X-Plane Layer is shown in the scene settings.

Caveats

You are limited by Blender's maximum of 20 object layers. All objects being collected must be visible on the same layer as being exported, including children. For example: Cube A, B, and C are on layer 1. Cube C has a child Cube D. Cube D must be visible on layer 1 as well. Making objects visible in multiple layers is not recommended because it can get messy fast. [fact check]

Example Workflow

An example use of this workflow is a "warehouse collection" project. Models of warehouses (small, medium, large, and mirrored variants) are each organized into layers. Layers mode is a good choice because it would be a collection of less than 20 things to model, where having many being visible would not be a problem or a distraction - it is not a necessary part of an artist's workflow to use the Object Layer Visibility feature to repeatedly show or hide parts of the model since the warehouses can be spread apart.

Root Object Mode

In "Root Object" mode, Blender Object's parent-child relationships are used to establish which X-Plane layers groups of objects should be exported together with. In the [Blender Outliner](#), top level [Empty Data Blocks](#) [icon] can be set to be root objects. The tree of children of the root object are collected as the objects to be exported to the OBJ. Blender Object Layers are irrelevant in the process.

Caveats

A root object cannot contain a root object.

Example Workflow

An example use of this workflow is an airplane with many parts. An airplane may have more than 20 parts, and much of them are nestled together making it hard to work on the model without hiding many parts of it. It would be very annoying to have to continually enable and disable the Blender Object Visibility Layers to make each component export which, fortunately, is not necessary in "Root Object" mode. You are free to use that feature of Blender however you wish.

Compile Normal-Textures

Shown always, only affects layers where `Autodetect Textures` is on. On by default. A checkbox which turns on or off the compiling of normal textures from existing normal and/or specular textures as needed. Given a detected normal texture, specular texture, or both,

XPlane2Blender will create a combined image and save it to disk (with the suffix "_nm", "_nm_spec", or "_spec" depending on what was used in the generated file). This will be referenced in the OBJ as a replacement for the regular normal or draped normal file.

See [materials] for more information.

Advanced Settings

Optimize

On by default. A checkbox which turns on or off the remove duplicate vertices optimization. Removing duplicate vertices can increase performance in X-Plane, but may also increase export time. All OBJs will render the same regardless of optimization.

Print Debug Info To Output, OBJ

On by default. A checkbox which changes what debug information gets printed and where. During export, XPlane2Blender keeps a log of events, broken down into four categories: *ERROR*, *WARNING*, *INFO*, and *SUCCESS*. *ERROR* and *WARNING* events are always printed, *INFO* and *SUCCESS* events statements are printed only when the checkbox is checked.

Also, when this box is checked, information such as mesh names, vertices index numbers, and exporter version numbers are printed as comments in the .obj. They are invisible and meaningless to X-Plane, but extremely useful for humans reading or debugging them.

Create Log File

Requires `Print Debug Info To Output, OBJ` **turned on.** **Off by default.** A checkbox which indicates whether or not the internal log will also get saved to a text file `xplane2blender.log`. It is located in the same directory as the saved .blend file.

Object Settings

A Blender object's settings are accessed via the X-Plane panel under the Properties Pane's "Object" tab. Each Blender object has its settings set per object. Object settings are available for mesh, empty, armature, and lamp data-blocks.

Relevant X-Plane Settings

Blender Object's Transformation and Relation data can be important to exporting OBJs.

Transform

Setting	Description	Can Keyframe Values?
Location	The X, Y, Z co-ordinates of the object, relative to its parent or the world origin. Changing these values change where the model is located in X-Plane	Yes
Rotation	The degrees of rotation around the X, Y, Z (axis-angle and Quaternion also have a W) axis of rotation the object, relative to its parent or the world origin. Changing these values change where the model is rotated in X-Plane	Yes
Scale	The scaling factor of the model along its X, Y, Z axis	No, the change to the object's vertices due to scaling is only saved to the OBJ once

Rotation Mode - The rotation mode for Blender rotations. XPlane2Blender supports all modes and exports them equally

Relations

Layers - An array of true/false toggle switches that correspond to what Blender layers it is in. When **Export Mode** is set to "Layers Mode", this changes which X-Plane layer (and therefore OBJ) the object gets collected and placed into. In "Root Object" mode it has no affect on the output or execution of the exporter

XPlane2Blender Settings

Root Object

`Root Object` - Off by default. When `Export Mode` is set to "Root Object" this checkbox will appear. Checking it will set this to be a `Root Object`. Settings to control the X-Plane layer this and child objects will be collected into will appear underneath. These are the exact same settings used in Layers Mode and found under the "Scene" tab.

Datarefs

Blender's location and rotation keyframes export where the geometry of objects should be in an animation, dataref keyframes describe how X-Plane should play the animation. This is different from traditional animation, where the animation tells a rendering engine (such as a 3D movie exporter) how to move geometry over time. Based on the value of a dataref inside X-Plane, X-Plane's rendering engine will interpolate between keyframes and display your model at specified positions.

For instance, to animate a yoke moving forward and back based on a player's joystick, you would use the dataref "sim/joystick/yoke_pitch_ratio". In X-Plane, this dataref will have a value between -1.00 and 1.00 (inclusive). In Blender you would model the yoke (the geometry), and create at least 2 Blender location and rotation keyframes, say, all the way forward on frame 1 and all the back on frame 2. You would then create X-Plane dataref keyframes on frame 1 and frame 2, with a value of -1.00 and 1.00 respectively. In sim, if the dataref is 0, X-Plane will automatically figure out how to position and rotate the yoke halfway between the position specified at frame 1 and the position specified at frame 2 (aka half way). If the dataref is -.80, X-Plane will automatically position the yoke almost all the way towards the position specified at frame 1 (almost all the way forward).

The more frames you add in Blender, the more precisely you can control the animation. Suppose the yoke in your aircraft has a dead-zone near the half way point. By specifying additional frames to show a slow down and specifying at what value of "sim/joystick/yoke_pitch_ratio" the player should see that frame, you can produce complex animations!

Since Blender's frames per second based timeline and X-Plane's dataref based "timeline" are different, you can use Blender's animation tools to preview but not directly export animations.

Dataref Animation Details

Use **Add Dataref** to add a dataref animation section, and click the small white **x** in the top right corner of a dataref animation section to remove it.

Dataref Path - Empty by default, this is the dataref for this animation. The dataref must be a standard X-Plane dataref or one supported by a plugin. It is not validated by XPlane2Blender.

Search datarefs - When clicked, the default web browser will open the site <https://www.siminnovations.com/xplane/dataref/index.php> where one can search for applicable datarefs. www.siminnovations.com is a third-party website not supported by Laminar Research. It is provided as a convenience only and is not guaranteed to be up to date or correct.

Animation Type - The type of animation to export. The types X-Plane supports are "Show", "Hide", and "Transform" (the default).

- "Show"/"Hide" animations show or hide the object when a dataref is greater than or equal to **value 1** and less than or equal to **value 2**. Blender keyframes are not needed to enable show/hiding animations. Since the range between Value 1 and 2 is inclusive, both a show and hide animation are necessary.

For example, suppose you have a dataref that is between 0 and 1 and you want to create an object that is shown between 0 and 1 and hidden after that. One would select that object, create a "Show" dataref animation with Value 1 as 0 and Value 2 as 1, and a "Hide" animation with Value 1 as 1 and Value 2 being 1 or higher.

- "Transform" animations use Blender location and/or rotation keyframes to transform objects in space. Thusly, Blender animation keyframes on the selected object or armature are required before the following options are shown.

Transform Dataref Animations

Each frame on Blender's timeline is a place to add or remove a dataref keyframe.

Add/Update an X-Plane Keyframe and **Remove an X-Plane Keyframe** - Each button, with the icon of a key and key with a red slash through it, add/update or remove X-Plane Dataref keyframes to an object. These keyframes are combined with the Blender's keyframes and exported as animation inside the OBJ.

Value - When the dataref is this value, X-Plane will have interpolated the animation to this frame. It is 0 by default, but that does not mean that is a valid number for all datarefs. You must look up documentation for each dataref and see which datatype are range of numbers it requires. Every time you change the value, you must also click the **Add/Update an X-Plane Keyframe** button before switching frames. Otherwise the value will not get saved.

`Animation Loops Every` - Looping allows you to create an animation that repeats, even when the dataref driving the animation is not a fixed set of values. One could think of it as "when the dataref has advanced by this much, loop the animation back to the beginning". Only one loop value is allowed per animation, on the last keyframe in an animation or shortly thereafter.

If a dataref possible values are unbounded (increases infinitely) instead of in a range or set of values, it is still possible to get a repeating animation. When looping is not 0.00, use looping to still get repeating animation. , instead of **A float, incrementing by, .03, describing the amount of loops the animation will perform. 0.00 by default, making for 0% looping.** See [whatever] for more details.

Show/Hide Animations

Show/Hide has two values, where the object will be shown/hidden when the data is greater than or equal to `value 1` and less than or equal to `value 2` . Datarefs that conflict about showing and hiding are not validated by the exporter.

Export Animation In Layers

`Export Animation In Layers` - **An array of checkboxes, mirroring Blender's Object Layer visibility, all enabled by default. Only available in [Layers Mode](#).** If a layer checkbox is disabled, then only the Blender Object's mesh will be exported in the layer, but **not** any of it's animations.

Manipulator Settings

Manipulators are part of the cockpit objects. They let you assign ways to use inputs to control the object. The following are available after `Manipulator` is checked.

Depending on type, different options are available to set for the manipulator. See the Manipulator spec [here](#)

Types Of Manipulators

None

The default, clicking on an object does nothing. Non-cockpit objects always have no manipulator.

Panel Click

Includes

-

Requires object with panel texture or panel texture region. A click on the object is mapped through to the 2-D panel based on the panel texture's mapping.

Drag-Axis

2-d Drag

Command

Command-Axis

Opaque

This sets the manipulation to 'no-op' (no operation). Unlike "none" manipulation, opaque manipulation "swallows" the mouse, preventing it from clicking on other manipulators that may lie behind it. It would typically be used to act as a safety-guard over a switch.

Each Manipulator type will allow you to set one or more of the following.

Drag

Manipulator Values

Different manipulators will have you set different values to fill out. Rather than listing every possible combination, here is what each means individually. Each value type has the same meaning, regardless of `Manipulator Type` chosen.

If a value is connected to a dataref

Common Manipulator Values

- `Drag X` - Length of X axis (meters)
- `Drag Y` - Length of Y axis (meters)
- `Drag Z` - Length of Z axis (meters)
- `Value 1` - Value 1 of the range of values over which a Drag Axis manipulator changes its dataref
- `Value 2` - Value 2 or the range of values over which a Drag Axis manipulator changes its dataref
- `Value 1 Min` - Minimum for a 2D Drag manipulator's dataref 1 range (-x to +x)
- `Value 1 Max` - Maximum for a 2D Drag manipulator's dataref 1 range (-x to +x)
- `Value 2 Min` - Minimum for a 2D Drag manipulator's dataref 2 range (-y to +y)

- `Value 2 Max` - Maximum for a 2D Drag manipulator's dataref 2 range (-y to +y)
- `On Value` - On value for toggle (must match value of dataref)
- `Off Value` - Off value for toggle (must match value of dataref)

Command Manipulator Values

- `Command` - The command to fire when manipulator is used
- `Positive Command` - Command to be executed when cursor is in positive region of Command Axis manipulator and mouse is being held down
- `Negative Command` - Command to be executed when cursor is in negative region of Command Axis manipulator and mouse is being held down
- `Dataref 1` - The first dataref to be modified by the manipulator
- `Dataref 2` - The second dataref to be modified by the manipulator
- `Step` - Dataref increment

Mouse Manipulator Values

- `Value On Mouse Down` - Value to set dataref on mouse down
- `Value On Mouse Up` - Value to set dataref on mouse up
- `Value On Mouse Hold` - Value to set dataref on mouse hold
- `Click Step` - Value change on click
- `Hold Step` - Value change on hold
- `Wheel Delta` - Value change on mouse wheel tick
- `Exp` - Power of an exponential curve that controls the speed at which the dataref changes. Higher numbers cause a more “non-linear” response, where small drags are very precise and large drags are very fast

LODs

`LODs` - **Four checkboxes that determine which levels of detail X-Plane should place this object. By default all boxes empty, meaning place in every LOD category** LOD ranges are defined in the [layer](#) settings.

Override Weight

`Weight` - **An int used to offset the order in which similar elements are written to the Obj, by default 0.** Needs better desc.

Custom Object Properties

Objects can have custom properties (another name for attribute) as well. These will get written directly to the OBJ in the form of

`Name` - The name of the custom property `value` - The value of the property `Reset` - OBJ attributes often come in pairs, such as `ATTR_light_level` and `ATTR_no_light_level`. An attribute may be turned on at the beginning of an object in an OBJ, and turned off (aka reset) in time for the next one. In the case of `ATTR_whatever`, if it was never turned off every object afterward would take on that attribute. The value for `Reset` will be written before the next object starts. `Weight` - An int. The larger the weight the later it gets written in the OBJ

Custom Animation Properties

Objects can also have custom animation properties. These also get written directly to the OBJ in the form of `Name Value` with no validation.

`Name` - **The custom animation directive to be written out. ANIM_ is *not* prepended**
`Value` - **The value for the custom animation directive.** `Weight` - **An int. Attempts to change the order of the properties, with larger numbers written later.**

Object Conditionalization

Objects X-Plane 10 supports the conditionalization of OBJ files. Conditionalization is a process whereby the contents of a file are skipped or used depending on rendering settings. Typical uses of conditionalization are to change the appearance of an object when shadows are enabled, or when HDR is on.

`Variable` - **A dropdown menu of "Global Shadows", "HDR", and "Version 10x", "Global Shadows".** `Must Be On` - **A checkbox which, when checked, tells X-Plane this rendering variable must be on to draw the variable. If unchecked, it tells X-Plane this rendering variable must be off to draw the variable. Removing the conditional lets X-Plane make its own decisions.**

Layer Settings

During the export process, Blender data (meshes, animations, materials, etc) are combined with X-Plane Layers and turned into .OBJs. Depending on your export mode defined in the [scene settings](#), the layer settings will appear in the Blender Properties Pane's Scene tab, or in selected root Object's Object's pane. All options are the same, regardless of export mode and where they appear in the UI. The only difference you may see is that in Layers mode, an X-Plane Layer settings section will have a header of Layer #, and in Root Objects mode will simply have the header of "Root Objects". This is purely cosmetic.

Basic Settings

Name - This is the name of the layer and ultimately the file name of the OBJ. When empty and in "Layers" export mode, the name "layer_x" is automatically applied, where x is the layer number. When empty and in "Root Object" export mode, the root object's name will become the layer name. If the name is a relative path, the exporter will automatically create the folders necessary.

Type - The type of X-Plane OBJ being exporter. This greatly changes what options are available and the behavior of the exporter.

- Aircraft (Part): For parts of an aircraft to be tied together with Plane Maker. Supports panel textures. The default choice
- Cockpit: For parts of an airplane's cockpit. Shows Cockpit Regions options and hides LODs options. Cockpit objects also support panel textures
- Scenery Object: Objects that are drawn as scenery. They can include draped geometry that hugs the ground
- Instanced Scenery Object: Like a scenery object, but is instance-able. Instancing is the ability for X-Plane draw more than one OBJ in a single stroke with the GPU. The less complicated the OBJ, the more likely it will pass for being instance-able. A massive performance booster for your model

Textures

X-Plane textures must be a .dds or .png file. Blender supports both. A .png will suffice for most applications. See [Texture Settings](#) for more information.

Autodetect Textures | On | Turns on or off autodetecting an OBJ's textures from Blender's material's texture slots. When unchecked, text fields will appear and you will be able to manually specify which textures should be used. **Beware: Some material and textures validations will be turned off!** In addition, Composite Normal-Textures will be ignored for this layer.

Setting	Default	Description	Requirement(s)
Default	""	The file path to the daytime texture to be used for this object	
Night	""	The file path to the night time texture to be used for this object	
Normal	""	The file path to the normal/specular map to be used for this object	For more details about what types of textures can be used in this slot, see [more info here].
Draped	""	The file path to the draped texture	Layer type must not be aircraft or Cockpit
Draped Normal	""	The file path to the draped normal texture	Layer type must not be aircraft or Cockpit

Objs can still be exported, even if **Autodetect Textures** is turned off, and any of these file paths is left blank.

Levels of Detail

Setting	Default	Description	Requirement(s)
LODs	"None"	Specifies the number of LOD bins to customize. When set to a number, that many sub LOD boxes will come up, allowing you to specify per bin. Each LOD bin has a Near and Far clipping distance (by default 0). The OBJ will only be visible at between those distances. When having 2 or more LODs, make sure they overlap or your model will flicker: (0,1000),(1000,3000), (3000,6000).	
Max. Draped LOD	0.00	Determines maximum LOD distance for draped geometry. 0 means "As far as possible."	Type must be "Scenery" or "Instanced Scenery".

Scenery Properties

Perhaps a bit misnamed, this group is not entirely exclusive to just Scenery and Instanced Scenery Exports.

Layer Grouping

Setting	Default	Description	Requirement(s)
Layer Group	"None"	Layer groups are a way to influence the draw order of scenery. It can be done at large by "Group" or at a smaller scale per group via the <code>Layer Group Offset</code> . The categories of each are (hopefully) self explanatory	
Layer Group Offset	0	A layer group offset attempts to fine tune the draw order for this OBJ within its layer group	
Draped Layer Group	"None"	Like <code>Layer Group</code> but for draped objects	Only available for "Scenery" and "Instanced Scenery" exports
Draped Layer Group Offset	0	Like <code>Layer Group Offset</code> but for draped objects	Only available for "Scenery" and "Instanced Scenery" exports

Shadows

Most shadow properties are set in the material settings, except for `Cast Shadow (Global)`.

Setting	Default	Description	Requirement(s)
Cast Shadow (Global)	On	If unchecked, the exported OBJ will not cast any shadows	

Slope Properties

These settings change when and how a scenery object rests on the ground.

Setting	Default	Description	Requirement(s)
Slope Limit	Off	Enables or disables an OBJ's slope limit, the minimum and maximum pitch and roll of a slope X-Plane can select the OBJ for. . Enabling <code>Slope Limit</code> opens additional options. See here	

`Tilted` |Off| When checked, X-Plane rotates the object before placing square on the ground even when the ground is a slope, as opposed to placing as if it were flat relative to gravity|

`Required Surface` |"Any"|Requires X-Plane to place this OBJ on only certain kinds of surfaces: "Dry", such as land or "Wet", such as oceans, rivers, and lakes, or "Any" for either of them|

Advanced Options

Setting	Default	Description	Requirement(s)

`Slung-Load Weight` - **A float indicating the weight in pounds the object is. 0 by default.** It is used by physic engine for things carried (and possibly dropped) by planes.

`Export` - **A checkbox which determines whether or not the layer is exported. On by default.** Only useful to turn off when you want to debug a scenery object by not having it there.

`Debug` - **A checkbox which determines whether or not the layer will emit debug information. On by default.** The debug information will only show up if the Scene Settings' `Debug` is also on. Turning this off will have no affect on OBJ performance.

Export Path Directives

Export Path Directives are unofficial OBJ directives that an internal Laminar Research tool uses to extract and automatically generate library.txt files from. EXPORT directives are ignored by X-Plane.

When the button `Add Export Path Directive` is pressed, a new Export Path directive slot is added. When the x-mark on the directive is clicked, it is removed. Empty export path directives are skipped.

Custom Properties

Custom properties are a key value pair, which get written directly to the header of the OBJ file, completely without validation. Some use this for 3rd party plug-ins or tooling or to gain access to new OBJ features that are not yet built into the exporter. For the latter case, this is NOT recommended and should be deprecated as soon as the exporter supports the OBJ directive! Without rigorous, spec compliant, and tested validation OBJs could become invalid or slow.

Material Settings

XPlane2Blender's material model is different from Blender's. In XPlane2Blender, all Blender objects must have a material. If you have multiple materials, only the first one is chosen. A material can have any number of texture slots that do the majority of the work changing how an OBJ looks in X-Plane.

Due to some inconveniences of Blender's data model, some UI elements will be shown even if using them would result in an invalid OBJ or would have no affect.

Relevant Blender Settings

Link

`Link` - The data-model linkage of the material **must be set to "Data"**. If you have materials that are linked by "Object", it is recommended to remove them to keep things neat

Specular

Setting	Default	Description	Requirement(s)
<code>Intensity</code>	0.50	How intense (bright) the specular reflection of the material is. All materials in an OBJ must have the same specularity	

XPlane2Blender Settings

Draw Linked Objects

These settings are used to decide when and how to draw the material.

Setting	Default	Description	Requirement(s)
Draw Linked Objects	Off	If checked, any object that uses this material will be drawn. If left unchecked, any object that uses this material will not be drawn	
Draped	Off	If checked, geometry with this material will hug the ground	Only valid for "Scenery" and "Instance Scenery" exports
Normal Metalness	Off	When checked, X-Plane will take the blue channel of this material's normal map and use it as base reflectance. This is commonly used for metal-like surfaces	Export Version must be greater or equal to "11.0x"
Blend	"Alpha Blend"	Controls how texture's alpha channels are blended with each other. See the Blend Mode Options Table for more details	
Alpha Cutoff Ratio	0.5	For this material's textures, alpha levels below the Alpha Cutoff Ratio are rendered as fully transparent. Alpha levels above the level are fully opaque. Values can range from 0.00 to 1.00	Blend must be set to "Alpha Cutoff"

Blend Mode Options

Blend Mode	Description	Requirement(s)
"Blend Shadow"	In shadow mode, shadows are not blended but primary drawing is	
"Alpha Cutoff"	Textures alpha channel will be used to cutoff areas above the Alpha Cutoff Ratio	
"Alpha Blend"	Textures alpha channel will be blended	

Blend Glass

The alpha channel of the albedo (day texture) will be used to create translucent rendering.

Export Version must be at least "11.0x"

Surface Behavior

Surface Behavior settings control how this material interacts with X-Plane in non-visual ways.

Setting	Default	Description	Requires
Surface Type	"None" - a smooth surface	A drop down menu of types of surfaces that X-Plane can simulate the bumpiness of. Grass is more bumpy than concrete, for instance.	
Deck	Off	When checked, objects with this material can be flown or moved under	Surface Type must not be "None"
Camera Collision	Off	When checked, the X-Plane camera will be prevented from moving through objects with this material	Always shown in UI but will only affect Cockpit objects

Light Levels

Setting	Default	Description	Requires
Override Light Level	Off	When checked, X-Plane's chosen object night lighting is overridden with custom dataref driven values	
Value 1	0.00	First value to be mapped to the dataref	Override Light Level is on
Value 2	0.00	Second value to be mapped to the dataref	Override Light Level is on
Dataref		The dataref to change the value of the material's night time lighting level	Override Light Level is on

Day-Night Preview Balance

Day-Night Preview Balance - This setting provides a convenient way to preview the material in Blender's 3D View. To enable the Day-Night Preview feature, add an albedo texture (uses Diffuse->Color) and a night texture (uses Shading->Emit) to the material. Adjusting this slider sliding this will automatically adjust their preview in the 3D Viewer. It does not change the content of exported OBJs.

Instancing Effects

These settings can only apply to **Instance Scenery** exports.

Setting	Default	Description	Requirement(s)
Tint	Off	If checked, Albedo Tint and Emissive Tint settings will be enabled.	
Albedo	0.0	A float between 0.0 and 1.0, where 0.0 is no darkening of the albedo (day time) texture and 1.0 is total darkening of the albedo texture	
Emissive	0.0	A float between 0.0 and 1.0, where 0.0 is no darkening of the emissive (day time) texture and 1.0 is total darkening of the emissive texture	

Polygon Offset

Setting	Default	Description	Requirement(s)
Polygon Offset	0	Indicates to X-Plane what order the object should attempt to be drawn in, from farthest away from the player camera to closets. 0 tells X-Plane to automatically chose drawing order. Very useful for an object with many parts that are close together. See the blog post The road to hell is paved with ATTR_poly_os for a more detail	

Cockpit Panel

Setting	Default	Description	Requirement(s)
Part Of Cockpit Panel	Off	If checked, objects with this material will use the OBJs panel texture and will be clickable	
Cockpit Region	None	Defines which region this material should appear in. If set to none, no region will be used. The region number refers to the regions defined in X-Plane Layer's Cockpit Regions	

Custom Properties

See Object->Custom Properties

Texture Settings

XPlane2Blender does not actually have any texture settings. However, certain Blender settings affect the export and even creation of new textures.

What Textures and Materials Mean to X-Plane

As stated before, very little Blender information related to materials actually gets written to an OBJ. Instead, most of what changes what an object looks like in Blender comes from Texture work.

An OBJ file does not contain the texture, but rather, the path to the image file. **Only .png and .dds file types are allowed.** .dds files have special considerations, see the Further Reading at the bottom.

Types of Textures

Albedo (day time) Texture

An the albedo texture

Lit (night time) Texture

Specular Texture

Normal Texture

Relevant Blender Settings

None

Texture Datablocks

`Material Texture Slots` - **A list of texture slots. XPlane2Blender will inspect these and attempt to use them during export. Each slot must be an "Image or Movie" type. The order of the slots is not important.**

Image

- `Source` - **Can be Generated or "Single Image"** Generated textures must be composit-able and saved as a real file to be referenced in an OBJ
- `Image/Movie File Name` - **If source is "Single Image", this is the file path of the image.** OBJs refer to their images relative to their own location, not an absolute path like `C:\Users\Me\plane_project\images\landing_gear.png` . You should keep paths relative to the .blend file to avoid distribution problems. Blender uses `//` in paths to refer to "the folder the .blend file is in". Blender and XPlane2Blender also supports the use of `..` (meaning *one parent folder*) and `.` (meaning *the current folder*).

For example: An image at `//images/landing_gear.png` would be read as "Starting at the folder the .blend file is in (`//`), looking in the folder `images` for the file `landing_gear.png` . Naming a layer `../test_objs/landing_gear.obj` would export the OBJ `landing_gear` to the folder `test_objs` , which is in the parent folder of the .blend file. No user folder or drive letter is referenced, making the project a lot more portable.

Mapping

- `Coordinates` - **Must be set to UV**
- `Map` - **Every texture must have a UV Map**

Influence

XPlane2Blender uses which influence options to determine the purpose of each texture. This is important for many parts XPlane2Blender, from Autodetect Textures, Composite Normal-Textures, which OBJ attributes to write, and certain validations.

Property	Texture Purpose	Use of Value
<code>Diffuse->Color</code>	Albedo (daytime) Texture	None
<code>Shading->Emit</code>	Lit (night time) Texture	None
<code>Specular->Intensity</code>	Specular Texture	Intensity value is used to accumulate the total specularity of all the material's texture slots (in addition to the material's own <code>Specular Intensity</code>)
<code>Geometry->Normal</code>	Normal Texture	None

Further Reading

Blender Manual

- [Relative Paths in Blender](#)

OBJ8 Spec Sections

- [TEXTURE APPLICATION](#)
- [TEXTURE](#)
- [TEXTURE_LIT](#)
- [TEXTURE_NORMAL](#)

DDS Format

Developer Blog Articles

- [DDS Revisited In X-Plane](#)
- [DDS and V10 - What Does It Do?](#)
- [Do Not use DDS as an Editing Format](#)
- [The Case For DDS](#)

Other Tools

- [DDSTool Manual](#)

Lamp Settings

The majority of light authoring in X-Plane is made through its large library of named and parameterized lights. These **"known"** Laminar Research lights come with an assortment of colors, directionality, and tie-ins with datarefs and airplane electrical systems to fit most artistic visions. Fully customizable lights are also available for your artist and complex plugin needs.

These "known" lights are listed in a file called "lights.txt", located in your X-Plane installation folder. A copy is also included in the addon folder "io_xplane2blender/resources/lights.txt". See the section **About lights.txt** for more about using this file.

Relevant Blender Settings

Lamp Type

`Type` - XPlane2Blender uses this to tell if the artist wants a "Point" or non-"Point" (such as a "Spot") light. "Point" lights have no meaningful direction and are therefore used for things like car headlights and blinking runway lights. They're more optimizable than non-"Point" spotlights. A "Spot" type has a meaningful direction to it and are good for representing things like spotlights. Further mentions of `Type` will refer to XPlane2Blender's setting unless otherwise noted.

`Color` - Custom Lights and old style lights use this for their RGB value

XPlane2Blender Settings

X-Plane Light Types

X-Plane's light `Type` refers to what type of information is required to use them.

- "Named" - Named lights use a specified pre-made light in X-Plane. They're the most common kind of light authors use
- "Param" - Param lights are like named lights, but also allow partial customization of their behavior such as color, direction, or brightness
- "Custom" - Custom lights are a textured billboards that use a size, a subsection of the OBJ texture, and, optionally, a dataref to control the look of them

"Default (deprecated)", "Flashing (deprecated)", "Pulsing (deprecated)", "Strobe (deprecated)", and "Traffic (deprecated)" are old X-Plane 7 lights, now deprecated.

Further Reading

Developer Blog Articles

- [Airplane Parameterized Light Guide](#)
- [Billboard and Spill Lights for OBJS](#)
- [Custom Lights](#)

OBJ8 Spec Sections

- [GEOMETRY_COMMANDS](#)

Forum Posts

- [Parsed contents of lights.txt](#)

lights.txt Comment Sections

- Lines 12-18, 38-62, 76-97

Common Settings

Setting	Default	Description	Requires
Type	"Named"	The X-Plane light type, not to be confused with Blender's Lamp <code>Type</code> . See the above "X-Plane Light Types" section for more details	
Name	""	The name of the light. It should match a "known" Lamina Research light	Type must be "Named" or "Param"
Params	""	The numerical parameters for the parameterized light, separated by 1 or more spaces. Comments should start with a "#" or "/"	Type must be "Param"

Light Name Vs Named Light

"Light Name" refers to the name of a pre-made light, such as "taxi_b" or "airplane_nav_left_size", see the setting `Name` above for more details. "Named Light" refers to the type of XPlane2Blender light which takes a light name and no parameters.

Custom Light Settings

Setting	Default	Description	Requires
Size	1.00	The size of the custom light, the larger the bigger. It isn't in meters or feet, it must be experimented with	Type must be "Custom"
Texture Coordinates	(0.0,0.0,1.0,1.0)	An area of the OBJ's texture where the texture for the light is, specified as a fraction of the texture's image between 0.0 and 1.0. The numbers represent the left, top, right, and bottom, respectively	Type must be "Custom"
Dateref	""	An optional dateref that will change the behavior of the light	Type must be "Custom"
Enable RGB Override	Off	When enabled, RGB Override Values will be used instead of the Blender color picker. Useful for certain datarefs	Type must be "Custom"
RGB Override Values	(0.0,0.0,0.0)	The values that will be used instead of the RGB picker	Enable RGB Override must be on

WYSIWYG Spot Lights

Most spot lights can be aimed directly, without animation. Simply rotate the light and it will appear in X-Plane as so! The exporter will use the lamp's rotation, even if a param list includes X, Y, Z or DX, DY, DZ components. (The param list must still be valid.)

This What-You-See-Is-What-You-Get behavior will be activated **if the Lamp is a "Spot" (or other non-"Point") lamp and a "Named" or "Param" type and the light name is included in the `lights.txt` file.**

Lights that do not meet the above requirements can be aimed with the "Useless Keyframe" trick, found in [Aiming Special Non-WYSIWYG Lights](#). **It is unnecessary in almost all cases.**

About lights.txt

lights.txt is essentially a massive table of information for X-Plane defining lights to be used in the sim. An artist only needs to be concerned with an extremely small portion of it. To read it, you'll need a good text editor (not Notepad or Word!) that can handle different line endings

and has an adjustable tabstop (8 seems to work). Comments are prefixed with a #. It is located inside the addon folder, `io_xplane2blender/resources/lights.txt`. It can be replaced by the lights.txt file included in X-Plane.

Do not make changes to this file! Changes may be incompatible with X-Plane and may, at best, make your object export incorrectly or, at worst, crash X-Plane!

The vast majority of this file is not relevant and may be misleading. The most important column is the second one: the light name. Multiple lines with the same light name represent different light drawing styles X-Plane can choose from, not different lights with the same name.

Anatomy of Param Lights

```
1.          2.          3.  4.
LIGHT_PARAM_DEF  apron_light_billboard  5  X Y Z W S
```

1. "LIGHT_PARAM_DEF" declares that the following is a description of a parameterized light
2. The light name
3. The number of parameters, meaningless to an artist
4. The parameters for this param light which must be replaced by the artist

Any light that doesn't have an associated LIGHT_PARAM_DEF is known simply as a "Named Light".

Hints For Finding A Light's Use

Though mostly undocumented, one can usually find the purpose of each light using some of these tips

- Experiment! Make a scenery object with lots of lights and view it in X-Plane!
- Light names are usually very descriptive
- Comments nearby may explain their purpose
- If a light name ends in "_bb" it usually means it is meant to be used with a billboard light. Look for a corresponding "_sp", such as "helipad_flood_bb" and "helipad_flood_sp"
- If a light name ends in "_sp" it usually means it's meant to be used as a pair with another light such as "taxi_g" and "taxi_g_sp" or "radio_const_bb" and "radio_const_sp"
- If it is a param light, the parameter names may be of help
- Though the first column is meaningless in terms of artistic and authoring decisions, it could give clues as to it's use. For instance, if a light name only has BILLBOARD types

associated with it, one can safely guess that the light will be a billboard.

"SPILL_HW_FLASh" lights such as "heli_morse_beacon" will **FLASh** at an interval

- If your file browser is sufficiently advanced, you can search for example uses of the light name in the text of existing .obj files in your X-Plane folder

SPILL..., BILLBOARD..., and other first column prefixes

The first column of each uncommented line of the light.txt is a defined light type. Each light name can have multiple types, and X-Plane will choose between them automatically. For instance, "taillight" includes 2 types for X-Plane to choose from:

```
BILLBOARD_HW    taillight...
SPILL_HW_DIR    taillight...
```

This information is only used by X-Plane and should not influence your decisions about whether to use Point or Spot lamps. The differences between SPILL_HW_DR, SPILL_SW, and etc have no impact on your work. For the overly curious, it is documented at the top of the file.

In Depth Articles

These articles provide in depth details about features or settings of XPlane2Blender. The information may contain content like tutorials, tables of information, or longer explanations of other content in the guide.

Aiming Special Non-WYSIWYG Lights

As mentioned in the [Lamp Settings](#), in XPlane2Blender 3.4, a certain type of "WYSIWYG" light was added:

If the Lamp is a "Spot" (or other non-"Point") lamp **and** a "Named" or "Param" type **and** the light name is included in the `lights.txt` file:

- XPlane2Blender will export the Lamp's rotation with What-You-See-Is-What-You-Get behavior
- The exporter uses Blender's rotation, even if a param list includes X, Y, Z or DX, DY, DZ components. (The param list must still be valid.)

This means "Custom" lights, unknown light names, and point lights cannot simply use their Blender rotation value to aim the light and an advanced hack must be used instead. While previous versions of this trick have been developed, new optimizations have broken some methods. **It is very unlikely that this trick applies to your project.** "Custom" lights are a discouraged (but not deprecated) feature; `lights.txt` will always be updated and should **never** be edited manually; and, since billboard lights rarely need **aiming** at all, it is far simpler to use a param light to [change the directionality of a billboard](#) than using this trick.

The "Useless Keyframe" Trick

After creating your Lamp, make an animation on it with the following details:

1. Datarref: "none" (or another non-existent datarref)
2. Rotation Keyframe 1: Whatever angle you want to aim the light at, a value of 0 for the datarref
3. Rotation Keyframe 2: A different angle than Keyframe 1's angle, a different value than Keyframe 1's value

How It Works

1. X-Plane selects which frame of an animation to interpolate to based on the value of the datarref associated with it. Since "none" is not a real datarref, its value will always be 0
2. Therefore, the animation will always be stuck on Keyframe 1, light pointing in the direction you want it to
3. Keyframe 2 exists and has "different rotation value/datarref value" pattern because:
 - An animation must have more than one keyframe or the exporter will not include it

in the OBJ

- An animation where all keyframes have the same rotation will make the exporter not include it in the OBJ
- All dataref values in an animation must be different or X-Plane considers it invalid with potentially unintended results

The "Useless Keyframe" Trick represents the simplest and safest way to meet the above requirements.

If you have a several of these special lights that are aiming in the same direction, you can save time and group them together. Parent all of them to an Empty or Bone, and apply the "Useless Keyframe" trick to only the parent.

Documentation for XPlane2Blender v3.5

Empty Settings

Empties have a special usage in XPlane2Blender. Having no mesh and no material, they are a fast and easy way to group and animate other Blender Objects. Post 3.5, they are also the way to specify special types of objects, such as particle emitters. Being a Blender Object, they share all the characteristics with what is covered [here](#).

XPlane2Blender Settings

Empty Special Type

Type	Use
None	No special usage
Particle Emitter	Empties location and rotation will be used to emit particles in X-Plane
Sound Emitter	A planned but unimplemented feature to set X-Plane sound emitters

Particle Emitter Settings

`Emitter Name` - A name from a .pss file that will be reference by the OBJ `Emitter Index` - The index of that Emitter Name to use, if there is one. Leave as 0 if not specified.

Object Settings

A Blender object's settings are accessed via the X-Plane panel under the Properties Pane's "Object" tab. Each Blender object has its settings set per object. Object settings are available for mesh, empty, armature, and lamp data-blocks.

Relevant X-Plane Settings

Blender Object's Transformation and Relation data can be important to exporting OBJs.

Transform

Setting	Description	Can Keyframe Values?
Location	The X, Y, Z co-ordinates of the object, relative to its parent or the world origin. Changing these values change where the model is located in X-Plane	Yes
Rotation	The degrees of rotation around the X, Y, Z (axis-angle and Quaternion also have a W) axis of rotation the object, relative to its parent or the world origin. Changing these values change where the model is rotated in X-Plane	Yes
Scale	The scaling factor of the model along its X, Y, Z axis	No, the change to the object's vertices due to scaling is only saved to the OBJ once

Rotation Mode - The rotation mode for Blender rotations. XPlane2Blender supports all modes and exports them equally

Relations

Layers - An array of true/false toggle switches that correspond to what Blender layers it is in. When **Export Mode** is set to "Layers Mode", this changes which X-Plane layer (and therefore OBJ) the object gets collected and placed into. In "Root Object" mode it has no affect on the output or execution of the exporter

XPlane2Blender Settings

Root Object

`Root Object` - Off by default. When `Export Mode` is set to "Root Object" this checkbox will appear. Checking it will set this to be a `Root Object`. Settings to control the X-Plane layer this and child objects will be collected into will appear underneath. These are the exact same settings used in Layers Mode and found under the "Scene" tab.

Datarefs

Blender's location and rotation keyframes export where the geometry of objects should be in an animation, dataref keyframes describe how X-Plane should play the animation. This is different from traditional animation, where the animation tells a rendering engine (such as a 3D movie exporter) how to move geometry over time. Based on the value of a dataref inside X-Plane, X-Plane's rendering engine will interpolate between keyframes and display your model at specified positions.

For instance, to animate a yoke moving forward and back based on a player's joystick, you would use the dataref "sim/joystick/yoke_pitch_ratio". In X-Plane, this dataref will have a value between -1.00 and 1.00 (inclusive). In Blender you would model the yoke (the geometry), and create at least 2 Blender location and rotation keyframes, say, all the way forward on frame 1 and all the back on frame 2. You would then create X-Plane dataref keyframes on frame 1 and frame 2, with a value of -1.00 and 1.00 respectively. In sim, if the dataref is 0, X-Plane will automatically figure out how to position and rotate the yoke halfway between the position specified at frame 1 and the position specified at frame 2 (aka half way). If the dataref is -.80, X-Plane will automatically position the yoke almost all the way towards the position specified at frame 1 (almost all the way forward).

The more frames you add in Blender, the more precisely you can control the animation. Suppose the yoke in your aircraft has a dead-zone near the half way point. By specifying additional frames to show a slow down and specifying at what value of "sim/joystick/yoke_pitch_ratio" the player should see that frame, you can produce complex animations!

Since Blender's frames per second based timeline and X-Plane's dataref based "timeline" are different, you can use Blender's animation tools to preview but not directly export animations.

Dataref Animation Details

Use **Add Dataref** to add a dataref animation section, and click the small white **x** in the top right corner of a dataref animation section to remove it.

Dataref Path - Empty by default, this is the dataref for this animation. The dataref must be a standard X-Plane dataref or one supported by a plugin. It is not validated by XPlane2Blender.

Search datarefs - When clicked, the default web browser will open the site <https://www.siminnovations.com/xplane/dataref/index.php> where one can search for applicable datarefs. www.siminnovations.com is a third-party website not supported by Laminar Research. It is provided as a convenience only and is not guaranteed to be up to date or correct.

Animation Type - The type of animation to export. The types X-Plane supports are "Show", "Hide", and "Transform" (the default).

- "Show"/"Hide" animations show or hide the object when a dataref is greater than or equal to **value 1** and less than or equal to **value 2**. Blender keyframes are not needed to enable show hiding animations. Since the range between Value 1 and 2 is inclusive, both a show and hide animation are necessary.

For example, suppose you have a dataref that is between 0 and 1 and you want to create an object that is shown between 0 and 1 and hidden after that. One would select that object, create a "Show" dataref animation with Value 1 as 0 and Value 2 as 1, and a "Hide" animation with Value 1 as 1 and Value 2 being 1 or higher.

- "Transform" animations use Blender location and/or rotation keyframes to transform objects in space. Thusly, Blender animation keyframes on the selected object or armature are required before the following options are shown.

Transform Dataref Animations

Each frame on Blender's timeline is a place to add or remove a dataref keyframe.

Add/Update an X-Plane Keyframe and **Remove an X-Plane Keyframe** - Each button, with the icon of a key and key with a red slash through it, add/update or remove X-Plane Dataref keyframes to an object. These keyframes are combined with the Blender's keyframes and exported as animation inside the OBJ.

Value - When the dataref is this value, X-Plane will have interpolated the animation to this frame. It is 0 by default, but that does not mean that is a valid number for all datarefs. You must look up documentation for each dataref and see which datatype are range of numbers it requires. Every time you change the value, you must also click the **Add/Update an X-Plane Keyframe** button before switching frames. Otherwise the value will not get saved.

`Animation Loops Every` - Looping allows you to create an animation that repeats, even when the dataref driving the animation is not a fixed set of values. One could think of it as "when the dataref has advanced by this much, loop the animation back to the beginning". Only one loop value is allowed per animation, on the last keyframe in an animation or shortly thereafter.

If a dataref possible values are unbounded (increases infinitely) instead of in a range or set of values, it is still possible to get a repeating animation. When looping is not 0.00, use looping to still get repeating animation. , instead of **A float, incrementing by, .03, describing the amount of loops the animation will preform. 0.00 by default, making for 0% looping.** See [whatever] for more details.

Show/Hide Animations

Show/Hide has two values, where the object will be shown/hidden when the data is greater than or equal to `value 1` and less than or equal to `value 2` . Datarefs that conflict about showing and hiding are not validated by the exporter.

Export Animation In Layers

`Export Animation In Layers` - **An array of checkboxes, mirroring Blender's Object Layer visibility, all enabled by default. Only available in [Layers Mode](#).** If a layer checkbox is disabled, then only the Blender Object's mesh will be exported in the layer, but **not** any of it's animations.

Manipulator Settings

Manipulators are part of the cockpit objects. They let you assign ways to use inputs to control the object. The following are available after `Manipulator` is checked.

Depending on type, different options are available to set for the manipulator. See the Manipulator spec [here](#)

Types Of Manipulators

None

The default, clicking on an object does nothing. Non-cockpit objects always have no manipulator.

Panel Click

Includes

-

Requires object with panel texture or panel texture region. A click on the object is mapped through to the 2-D panel based on the panel texture's mapping.

Drag-Axis

2-d Drag

Command

Command-Axis

Opaque

This sets the manipulation to 'no-op' (no operation). Unlike "none" manipulation, opaque manipulation "swallows" the mouse, preventing it from clicking on other manipulators that may lie behind it. It would typically be used to act as a safety-guard over a switch.

Each Manipulator type will allow you to set one or more of the following.

Drag

Manipulator Values

Different manipulators will have you set different values to fill out. Rather than listing every possible combination, here is what each means individually. Each value type has the same meaning, regardless of `Manipulator Type` chosen.

If a value is connected to a dataref

Common Manipulator Values

- `Drag X` - Length of X axis (meters)
- `Drag Y` - Length of Y axis (meters)
- `Drag Z` - Length of Z axis (meters)
- `Value 1` - Value 1 of the range of values over which a Drag Axis manipulator changes its dataref
- `Value 2` - Value 2 or the range of values over which a Drag Axis manipulator changes its dataref
- `Value 1 Min` - Minimum for a 2D Drag manipulator's dataref 1 range (-x to +x)
- `Value 1 Max` - Maximum for a 2D Drag manipulator's dataref 1 range (-x to +x)
- `Value 2 Min` - Minimum for a 2D Drag manipulator's dataref 2 range (-y to +y)

- `Value 2 Max` - Maximum for a 2D Drag manipulator's dataref 2 range (-y to +y)
- `On Value` - On value for toggle (must match value of dataref)
- `Off Value` - Off value for toggle (must match value of dataref)

Command Manipulator Values

- `Command` - The command to fire when manipulator is used
- `Positive Command` - Command to be executed when cursor is in positive region of Command Axis manipulator and mouse is being held down
- `Negative Command` - Command to be executed when cursor is in negative region of Command Axis manipulator and mouse is being held down
- `Dataref 1` - The first dataref to be modified by the manipulator
- `Dataref 2` - The second dataref to be modified by the manipulator
- `Step` - Dataref increment

Mouse Manipulator Values

- `Value On Mouse Down` - Value to set dataref on mouse down
- `Value On Mouse Up` - Value to set dataref on mouse up
- `Value On Mouse Hold` - Value to set dataref on mouse hold
- `Click Step` - Value change on click
- `Hold Step` - Value change on hold
- `Wheel Delta` - Value change on mouse wheel tick
- `Exp` - Power of an exponential curve that controls the speed at which the dataref changes. Higher numbers cause a more “non-linear” response, where small drags are very precise and large drags are very fast

LODs

`LODs` - **Four checkboxes that determine which levels of detail X-Plane should place this object. By default all boxes empty, meaning place in every LOD category** LOD ranges are defined in the [layer](#) settings.

Override Weight

`Weight` - **An int used to offset the order in which similar elements are written to the Obj, by default 0.** Needs better desc.

Custom Object Properties

Objects can have custom properties (another name for attribute) as well. These will get written directly to the OBJ in the form of

`Name` - The name of the custom property `value` - The value of the property `Reset` - OBJ attributes often come in pairs, such as `ATTR_light_level` and `ATTR_no_light_level`. An attribute may be turned on at the beginning of an object in an OBJ, and turned off (aka reset) in time for the next one. In the case of `ATTR_whatever`, if it was never turned off every object afterward would take on that attribute. The value for `Reset` will be written before the next object starts. `weight` - An int. The larger the weight the later it gets written in the OBJ

Custom Animation Properties

Objects can also have custom animation properties. These also get written directly to the OBJ in the form of `Name value` with no validation.

`Name` - **The custom animation directive to be written out. ANIM_ is *not* prepended**
`Value` - **The value for the custom animation directive.** `weight` - **An int. Attempts to change the order of the properties, with larger numbers written later.**

Object Conditionalization

Objects X-Plane 10 supports the conditionalization of OBJ files. Conditionalization is a process whereby the contents of a file are skipped or used depending on rendering settings. Typical uses of conditionalization are to change the appearance of an object when shadows are enabled, or when HDR is on.

`Variable` - **A dropdown menu of "Global Shadows", "HDR", and "Version 10x", "Global Shadows".** `Must Be On` - **A checkbox which, when checked, tells X-Plane this rendering variable must be on to draw the variable. If unchecked, it tells X-Plane this rendering variable must be off to draw the variable. Removing the conditional lets X-Plane make its own decisions.**

Animation Requirements For Smart Manipulators

Manipulators lets you specify how objects in Blender become interactive in X-Plane. In X-Plane 11.1x, you can now specify new manipulators which give a better experience in virtual reality. These new VR manipulators are

Manip Type	Potential Uses
Drag Axis *	Flap handles, parking brakes, simple air vents, or anything that move sin a linear fashion
Drag Axis With Detents	Any Drag Axis application, but with detents to overcome
Drag Rotate	Throttle quadrants, trim wheels, cowl flaps. Any mechanism that rotates around a fixed point
Drag Rotate With Detents	Any Drag Rotate application, but with detents to overcome

These manipulators use animation data to autodetect properties such as `X`, `Y`, `Z`, `v1 Min/Max`, etc. Dataref 1 and 2 can always be overridden manually. This article describes in detail the requirements for each.

*Drag Axis is an old manipulator now upgraded. Use the `Autodetect Settings` checkbox to opt in.

Further Reading

Manual Pages

- [Object Settings](#)

Developer Blog Posts

- [All About Clamping Animations](#)

Manipulators

- [Manipulators In-depth](#)
- [Manipulators](#)

VR

- [Manipulators and VR Part 1: From Mouse to Mechanism](#)

Example Blend Files

- [example_vr_manipulator_types.zip](#) This zip of blend files contains a variety of examples of working and not working smart manipulators, taken from the XPlane2Blender test suite. The last layer in each .blend file contains a correct working example. The rest are all the different ways you could get an error. Don't get the wrong idea, these are only some (simple) use cases of the feature.

About "Animation Sources"

The autodetection algorithm requires a certain number of animation sources, of certain types, in a certain order. The algorithm continues looking for animation sources until it fills its requirements, starting at the manipulator's mesh and then inspecting up its parents. This provides flexibility. Animations can appear anywhere in this chain from, including having gaps between and not having the manipulator mesh be the first animation source.

The ...-> indicates 0 or more non-animated meshes or bones are allowed in between.

Type	Animation Requirements (Listed In Expected Order)
Drag Axis	[Manipulator Mesh]...->Location Animation
Drag Axis With Detents	[Manipulator Mesh]...->Location (Detent) Animation...->Location Animation
Drag Rotate	[Manipulator Mesh]...->Rotation Animation
Drag Rotate With Detents	[Manipulator Mesh]...->Location (Detent) Animation...->Rotation Animation

Animation Source Rules

Common Animation Rules

- Must have one and only one dataref animation
- Must have actual movement (no animations of 0 degrees of rotation or 0 meters of movement)
- The mesh the manipulator is attached to must not have child datablocks or bones

Location Animation Rules

- Must have exactly two non-clamping Location keyframes and **no** Rotation keyframes

Rotate Animation Rules

- Must have exactly two non-clamping Rotation keyframes and **no** Location keyframes
- Must rotate around only one axis of rotation (in any rotation mode)
- Dataref keyframe values must be in ascending or descending order

Detent Location Animation Rules

Drag Axis With Detents:

- Must have a parent with a valid location animation source
- The detent axis must be at 90 degrees from the drag axis

Drag Rotate With Detents:

- Must have a parent with a valid location animation source
- The detent axis must not be along the rotational axis in any way. To understand why, imagine a wheel that has been welded onto an axle. The axle is along the X axis, therefore the wheel rotates around the X axis. When X-Plane simulates how a mouse drag or VR drag rotates this object, it only calculates how the wheel could possibly move. Therefore any dragging along the X axis is ignored and no detenting in that direction is possible
- Must have exactly two non-clamping Location keyframes and **no** Rotation keyframes
- Must have [Axis Detent Ranges](#)

Drag Rotate Regions

By adding additional keyframes between the start and end of a `Drag Rotate` OR `Drag Rotate With Detents` rotation, X-Plane will automatically adjust how long it takes to drag through those areas. In general, a large range will be slower to drag through, and a small range will be faster to drag through.

Keyframe #	Degrees	User Experience While Dragging At Constant Speed
1	0 - 90	Manipulator will rotate slowly through this section
2	90 - 150	Manipulator will rotate quickly through this section
3	150 - 180	Manipulator will rotate the quickest through this section

The speed is also affected by the ratio between degrees and size of dataref change.

Clamping is supported and recommended.

If you encounter anything that seems like a bug, please report it. However I won't investigate on bug reports that do not contain all relevant information.

How to report bugs

If you encounter a bug within XPlane2Blender please follow these steps to make it easier for the developer(s) to fix it:

1. Check the [issues-list at the GitHub-Project-Page](#), to see if your bug has not been reported yet
2. Open up Blender from a terminal/console. Under windows you have to launch blender with the Start > "run" command by typing "blender.exe" into it. Under Unix (Mac/Linux) you have to open up a terminal and type "blender".
3. Open up the file causing the bug.
4. Repeat the steps causing the bug.
5. Look into the terminal/console and copy the error-message that might have showed up in it. It's best If you copy everything, so the developer gets as much information as possible.
6. [Create a post at x-plane.org](#) / Or open up an [issue at Github](#) containing the following information: Your **Operating System**, your XPlane2Blender version (e.g. 3.20r1), your Blender version together with the **build-number** (if using Blender 2.5x this is the number after the "r"), the **exact steps for reproducing the error**, the **error message(s)**, if possible the file causing the error **cleared of all non-relevant parts for the bug**.
7. Wait and hope. We will look into it and fix it asap.

Please only send me private messages/email to send me the buggy files. Do not send me your bug report in a PM or email. Making the bugs/problems and their solutions public will help all other users of XPlane2Blender with the same problem.

If you think this is a good thing [buy me a beer](#).

If you think this is a good thing and want to support the developer buy me a beer using a donation.

