Arma Toolbox for Blender

Version 1.5

Table of Contents

Overview

Introduction

The ARMA Toolbox for Blender (ATB for short) is a set of scripts for the Blender 3D Modeler and Rendering software that allows you to create models for the Arma series of games. ATB makes it easier to export models you have created in Blender into P3D files suitable for including in Arma 2 or Arma 3.

Normally, you would export the model into a format like OBJ or 3DS and import them into Oxygen 2 for further processing. ATB makes this unnecessary, to a degree where you can most likely skip Oxygen 2 entirely.

This manual is intended to get you started. It does not assume too much knowledge of Blender, but you should have the basics covered.

What's New in 1.5

- Now looks for registry paths if the path of O2script.exe wasn't set manually.
- “Arma” tab if used with Blender 2.7 and beyond.
- Direct binary export. O2Script is only used to correct the normals.

What's New in 1.4

- Unreleased interim version

What's New in 1.3

- New Feature: Exporting of multiple UVSets
- New Feature: Toolbox Panel. The Toolbox panel contains a number of useful functions like bulk renaming and reparenting of textures, selection renaming, etc.

What's New in 1.2

1.2 is mainly a bugfix release. Some minor corrections have been made. The only notable new feature is the possibility to nominate a bone in an armature as the center of animation and calculate the RTM motion vector according to this bone.

What's New In 1.1

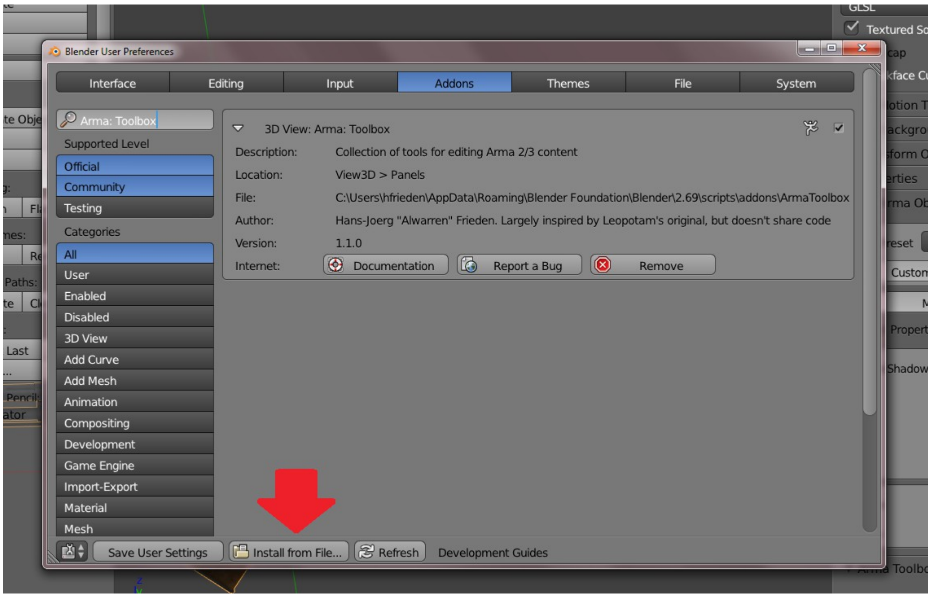
- The “Make ARMA Object” button has been replaced by a checkmark in the header of the panel (see “Making Arma Objects” on page 5)
- RTM Editing. ATB can now handle RTM animation files using a standard Blender Armature (see “Editing RTM Animations” on page 12)

Installation

This package comes with a ZIP archive that contains the Blender addon. In order to install it, please follow these steps:

1. Open the "User Preferences" Window in Blender. The User Preferences window can be found under File->User Preferences...

2. A window will open that will look somewhat like this



3D View: Arma: Toolbox ☑

Description: Collection of tools for editing Arma 2/3 content

Location: View3D > Panels

File: C:\Users\hfrieden\AppData\Roaming\Blender Foundation\Blender\2.69\scripts\addons\ArmaToolbox

Author: Hans-Joerg "Alwarren" Frieden. Largely inspired by Leopotam's original, but doesn't share code

Version: 1.1.0

Internet:

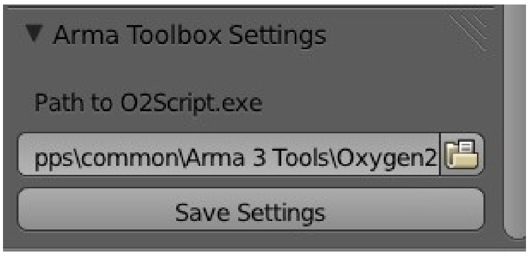
3. Go to the "Addons" tab, and click on the "Install from File" button at the bottom of the window. Browse to and select the ZIP file that came in this archive. It should result in the Arma Toolbox to be displayed in the large central area.

4. Clock on the checkbox to enable the addon.

5. *If the addon doesn't enable*, check the System console (use Window->Toggle System Console if it isn't visible) to see if there are any error messages.

6. If you want to keep the Arma toolbox enabled, click on "Save User Settings" to save your settings. If you don't want to keep it on, you will have to enable it every time you start up Blender. Close the User Preferences window.

7. In the Properties Shelf you should now have two new panels. One of them will not be visible unless a Mesh object is selected. The other one should always be visible, and if opened looks like this:



Arma Toolbox Settings

Path to O2Script.exe

pps\common\Arma 3 Tools\Oxygen2

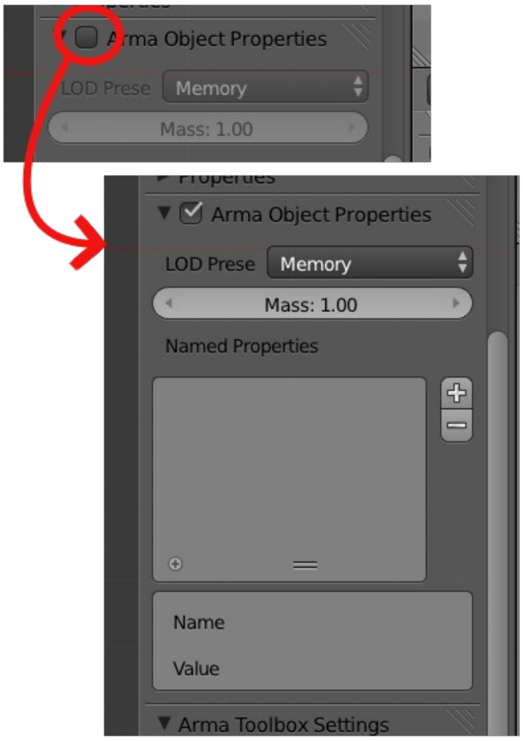
8. The "Arma Toolbox Settings" panel contains a single field, the path to the O2Script.exe binary. The location of this took depends on your installation. If you want to model for Arma 3 and have installed the tools via Steam, the most likely location is ”`C:\Program Files\Steam\SteamApps\common\Arma 3 Tools\Oxygen 2\`”.You can browse to this location (or wherever Oxygen 2 is installed) via the folder icon next to the field. Once you have the correct setting, press "Save Settings". You will not need to repeat this step, even if you disable the addon later. *Note that with 1.5 and later versions of the toolbox, the path is detected automatically and no manual interaction is required unless you have moved Oxygen or have a different installation you want to use.*

Getting Started with the Arma Toolbox for Blender

Making Arma objects

By default, the exporter ignores everything unless it is explicitly flagged as being an ARMA object. In order to export an object successfully, it needs a set of extra information. To reveal this extra information, it is necessary to make an object an Arma object first.

In order to show what this means, let's look at an example. Create an empty scene, then either select the default cube (if you have one) or create a new cube with SHIFT-A->Mesh->Cube and make sure it is selected. You will see the "Arma Objects Properties" panel in the Properties shelf. Like the “Background Images” panel, it has a checkmark next to the title. This checkmark represents the state of the currently selected object. To turn a normal object into an Arma object, click the checkmark. This will make the currently selected object an Arma object. Likewise, unchecking the checkmark will turn it back into a normal object. The object will retain all its settings, so you can temporarily turn off the checkmark to prevent the exporter from writing the object into the output file.



The only difference between an Arma object and a normal object is that the Arma object will be exported when you select Export to P3D file. Also, the settings will be grayed out if the object is not an Arma object and will hence be inaccessible.

The panel is roughly divided into three parts, marked A, B and C in the graphic. We'll go over these parts in more detail now.

LOD settings

LOD is short for "Level Of Detail", and in 3D games this is usually used to optimize performance. Depending on how far an object is away from the player's view, it is successively replaced by a lower resolution copy of the original. LOD is a very effective method; if you are standing half a kilometer away from a car, it is very unlikely that you will still be able to see the screws of the wheels, so there is no point in actually rendering them.

In Arma specifically, LOD has some extra meaning. Each P3D file contains at least one LOD, but usually contains much more than that. These LODs may be "traditional" LODs as described above, but may also be extra info or special versions of the object. For example, the "View Pilot" LOD is always used for the driver of a vehicle, or, in the case of a weapon, for the first person view. The "View Pilot" LOD is usually the most detailed, especially in weapons models, since it is likely that the player will see it very close and be able to see all the details (or lack thereof).

Another common LOD in Arma is the "Memory" LOD. The Memory LOD contains mostly single vertices that denote points of interest in the model's 3D space. In the weapons example, there are vertices in the Memory LOD were the gun's bullets will fly out, or where the empty brass is ejected from the ejection port.

Since the typical P3D file is composed of many LODs, the Arma Toolbox allows you to have any number of LODs in the same scene. Every object that is flagged as an Arma object will have exactly one LOD setting.

The upper field in the above figure (marked with 'A') is used to set this LOD. By default, it is set to "Custom" which means this is a "traditional" LOD. The number in the "Custom Distance" field is something of a scaling value. Typically the LOD with distance 1.0 is used for close objects, but it can get as big as you like as long as it stays below 1000. Typically, for a weapon, you'd have LOD's in the range 1.0 to 5.0, and typically they are at 1.0, 2.0, 3.0, 4.0 and 5.0.

If you select a different LOD in the LOD Preset combo box, the distance field will vanish. To mark the cube as a Geometry LOD for example, click on the Combo box and select "Geometry".

Mass Settings

The Geometry LOD needs a mass, and this field is only valid for a Geometry LOD. The value that you enter here is the total weight of the model. Typically, the Geometry LOD has named selections (vertex groups in Blender) that specify the weight distribution of the total weight.

Named Properties

The third field (C) is the "named properties" list. A LOD can have any number of named properties. These are bits of information for the game engine that are not interpreted in any way inside Blender (or O2).

The Plus and Minus buttons add or delete a named property. Clicking the Plus button will add an empty placeholder. You can change both the name and the value of the property by clicking the property and then the name or the value field.

UV Sets[1]

By default, there is only one UV coordinate set, usually named “UVSet”, under the object data tab in the properties window. However, Arma 2 and Arma 3 both accept up to 8 UV Sets for multi-material shaders. UV Maps in the Object Data tab are automatically exported as #UVSet# taggs in the resulting P3D file.

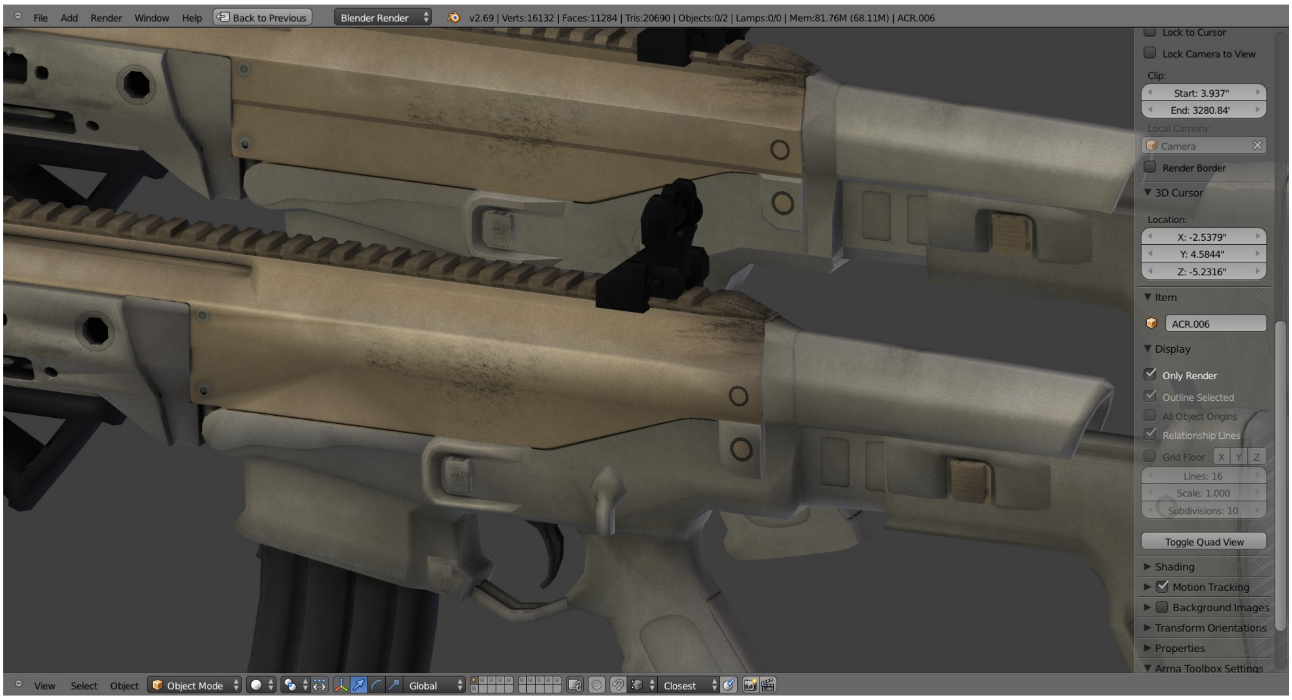
[1] This feature is only available in version 1.3 and later of the Arma Toolbox.

Sharp Edges

Blender will, by default, not show sharp edges correctly. You can mark an edge as a sharp edge by selecting the edge, pressing CTRL-E to bring up the edge menu, and clicking "Make Sharp" (or "Clear Sharp" to revert it).

In order to see the sharp edges, there is a trick that you can employ. Go to the "Modifiers" tab and add an "Edge Split" modifier to the object you are working with. In the modifier's settings, uncheck the "Edge Angle" and leave "Sharp Edges" enabled. Now, marking an edge as a sharp edge will display correctly.

IMPORTANT: Do not apply this modifier. Arma Toolbox can and will export sharp edges correctly. This is just a trick to get the correct shading showing.



v2.69 | Verts:16132 | Faces:11284 | Tris:20690 | Objects:0/2 | Lamps:0/0 | Mem:81.76M (68.11M) | ACR.006

□ Lock to Cursor
□ Lock Camera to View
Clip:
Start: 3.937"
End: 3280.84'
Local Camera:
Camera
□ Render Border

▼ 3D Cursor
Location:
X: -2.5379"
Y: 4.5844"
Z: -5.2316"

▼ Item
ACR.006

▼ Display
☑ Only Render
☑ Outline Selected
□ All Object Origins
☑ Relationship Lines
□ Grid Floor X Y Z
Lines: 16
Scale: 1.000
Subdivisions: 10

▶ ☑ Motion Tracking
▶ □ Background Images

The above image shows the same object with (top) and without (bottom) the Edge Split modifier . Note the shading errors on the receiver, mag release button guard, and other parts of the weapon.

Besides marking edges as sharp edges, the Arma Toolbox will also export flat shaded faces as sharp edges. If a face's shading is set to flat, all its edges will be exported as sharp. This method isn't used very frequently though since it gives very little control over the result – it's either all or nothing.

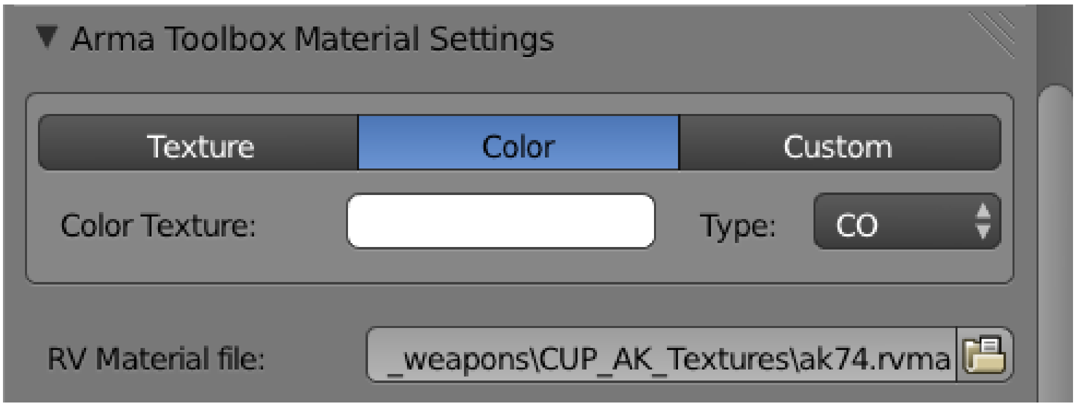
Materials and Textures

In a typical object in the Arma engine, each face carries two bits of information that determine how it is rendered: Texture, and Material.

Textures are usually in the PAA format that Blender does not understand. Likewise, Materials are stored in the RVMAT format.

If an object is flagged as an Arma object, an additional panel will show up on Blender Materials. This panel allows you to set texture and material for use in Arma, and it is independent of the textures and material settings used for this material, but the two items are saved with every face that has this Blender material assigned.

The Arma Texture/Material panel looks like this



Color Texture:

Type: CO

RV Material file: _weapons\CUP_AK_Textures\ak74.rvma

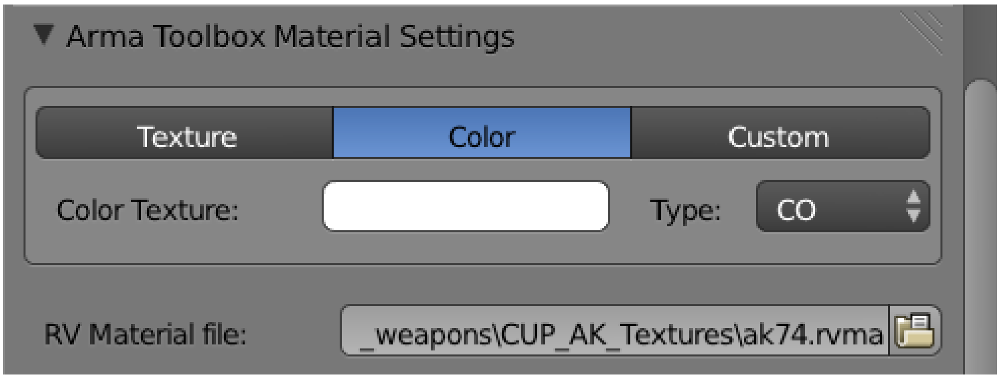
This panel has two regions, the top region for the texture, and the bottom region for the RV Material file.

The RV Material file is simply a file on your disk. You can browse to the file by using the folder button on the left. There are a few rules for this field that are supposed to make things easier (all of these also apply to texture files):

1. A leading drive letter ("P:" in this case) is stripped from the file name when exporting.
2. A leading backslash is removed as well since the names in a P3D are always relative to the root.

The top field is for the surface texture, and has three modes: Texture, Color, and Custom. The above figure displays the texture variant, and shows the Face Texture property field that works exactly like the RV Material field. It simply points to a PAA file that represents the surface texture.

If you select color mode, the display will change



Color Texture:

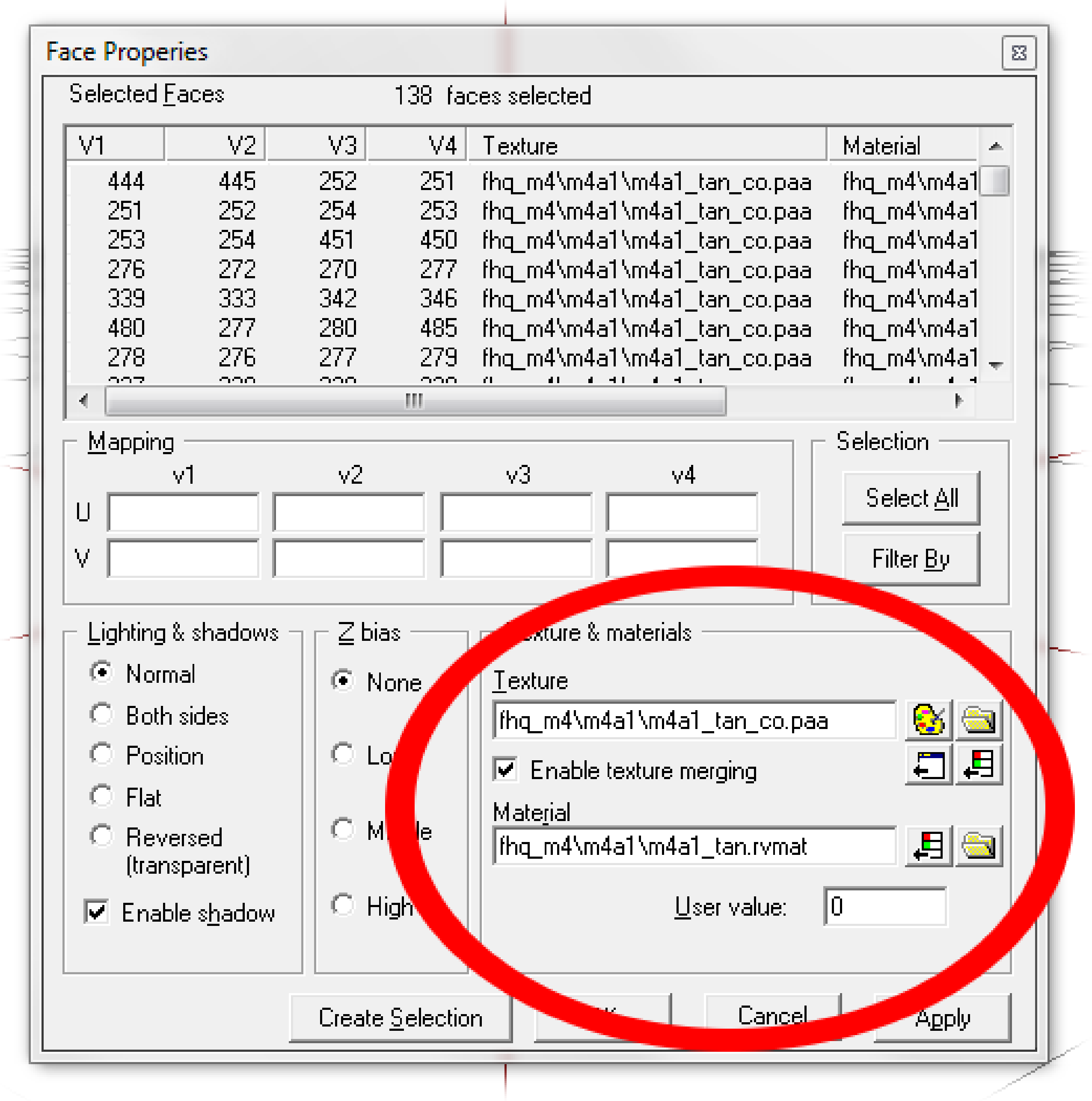
Type: CO

RV Material file: _weapons\CUP_AK_Textures\ak74.rvma

The Texture is replaced with a Color field and a type combo. The color field can be clicked on and will show an RGB dial. The Type field specifies a type of texture (CO for color, CA for color with Alpha, etc). Internally, this will generate a procedural texture (see the BIKI for more information on procedural textures).

Finally, setting the mode to Custom will show a text box similar to the Texture mode, just without the folder button. In this mode, the entered text is copied verbatim into the exported P3D file.

If you are used to Oxygen 2, the following dialog corresponds to the above panel:



Selected Faces 138 faces selected

V1	V2	V3	V4	Texture	Material
444	445	252	251	fhq_m4\m4a1\m4a1_tan_co.paa	fhq_m4\m4a1
251	252	254	253	fhq_m4\m4a1\m4a1_tan_co.paa	fhq_m4\m4a1
253	254	451	450	fhq_m4\m4a1\m4a1_tan_co.paa	fhq_m4\m4a1
276	272	270	277	fhq_m4\m4a1\m4a1_tan_co.paa	fhq_m4\m4a1
339	333	342	346	fhq_m4\m4a1\m4a1_tan_co.paa	fhq_m4\m4a1
480	277	280	485	fhq_m4\m4a1\m4a1_tan_co.paa	fhq_m4\m4a1
278	276	277	279	fhq_m4\m4a1\m4a1_tan_co.paa	fhq_m4\m4a1

	v1	v2	v3	v4
U				
V				

● Normal
○ Both sides
○ Position
○ Flat
○ Reversed (transparent)
☑ Enable shadow

● None
○ Lo
○ M le
○ High

Texture: fhq_m4\m4a1\m4a1_tan_co.paa
☑ Enable texture merging
Material: fhq_m4\m4a1\m4a1_tan.rvmat
User value: 0

Named Selections and Proxies

Named Selections

Named Selections are used, among other things, to animate things in Arma. Also, the memory LOD uses named selections to give meaning to the vertex cloud that it usually is. For example, the "eye" or "opticsView" named selection is usually used in weapons and attachments to mark the camera positions for the scoped and backup sight view.

Named selections consist of a number of vertices and associated weight. For skeletal animations, these weights are used to determine the amount of influence the matching bone has.

On the Blender side, Named Selections are vertex groups. You will find vertex groups on the "Object Data" panel.

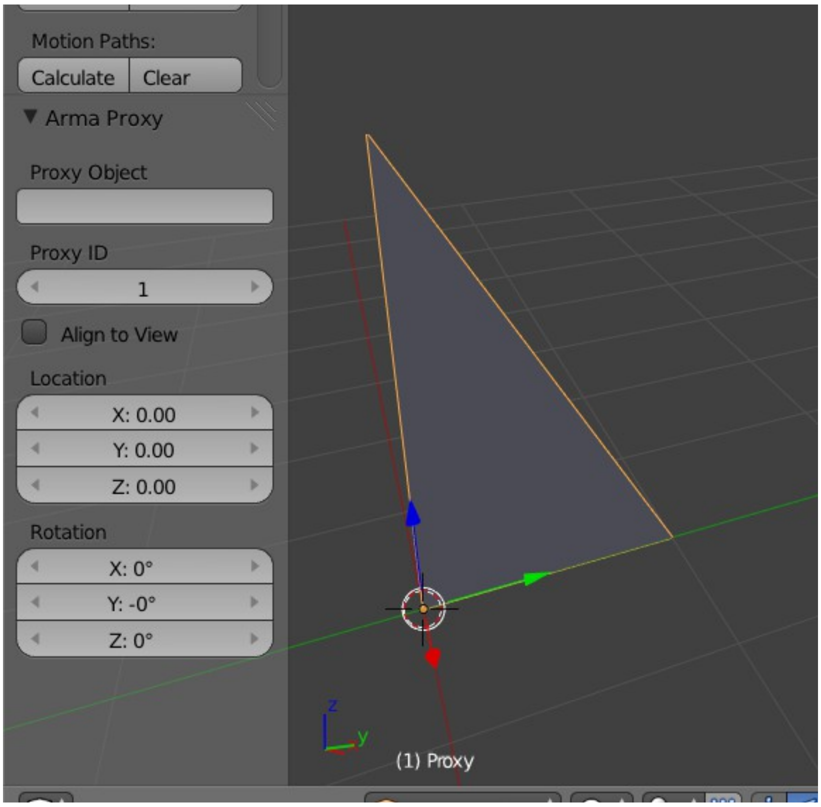
Just like normal usage in Blender, you can assign different weights to the vertices with the Weights slider under the vertex group list, or by going into weight paint mode in which blue denotes no weight (0.0) and red denotes 1.0. These weights are exported along with the name of the vertex group as a named selection.

Proxies[2]

Proxies are a special kind of named selection – they are usually replaced in game by something else. For example, the attachment system in Arma 3 is proxy based – you add a number of proxies to your weapon (all with a specific name) and the game knows where to attach a scope, silencer or laser pointer.

Proxies are named selections that start with the string "proxy:". As such, you can turn any triangle into a proxy by adding an aptly named vertex group. However, there is an easier method.

The 'Add Primitive' menu (SHIFT-A) contains an item "Arma Proxy" that will insert a proxy and associated named selection. Once you added the proxy, you can change its settings in the settings dialog in the tool shelf, or by pressing F6:



Proxy Object

Proxy ID: 1

□ Align to View

Location
X: 0.00
Y: 0.00
Z: 0.00

Rotation
X: 0°
Y: -0°
Z: 0°

The location and rotation can be changed here, but you can also move the proxy manually after you have placed it. The "Align to View" button does exactly this, align the proxy to your current view.

The two top fields "Proxy Object" and "Proxy ID" are the important bits. The "Proxy Object" is a path to a proxy model. For example, "`\a3\data_f\proxies\weapon_slots\side`" is the proxy for the side rail attachment of weapons. The Proxy ID is a number that starts at 1 and is used to identify multiple proxies of the same type (for example, for crew positions).

[2] This feature requires version 1.0.1 of the Arma Toolbox

At this time (Version 1.0.1 of the Toolbox) there are two major limitations on the proxy functions, none of which is deal-breaking.

For one thing, the Proxy ID must be manually assigned right now. This can get a bit finicky, but doesn't hamper the functionality itself.

The second limitation is the way that the 'Arma Proxy' operator works. Right now, you cannot use it in Edit mode. In order to get a proxy, you need to exit edit mode and add the proxy as a separate object, then join it with your mesh by selecting the proxy and then the object and pressing CTRL-J to join. The reason for this is that in edit mode you cannot create a vertex group. I don't know (yet) how to overcome this limitation.

Editing RTM Animations

Since version 1.1, the Arma Toolbox for Blender can handle direct export of animation into an RTM file. RTM files are used from within the Arma engine for all sorts of skeletal animation (as opposed to the simple animations possible with model.cfg).

Armatures

You will need a suitable armature for this. You can either make one yourself, or use one of the existing skeletons for Blender:

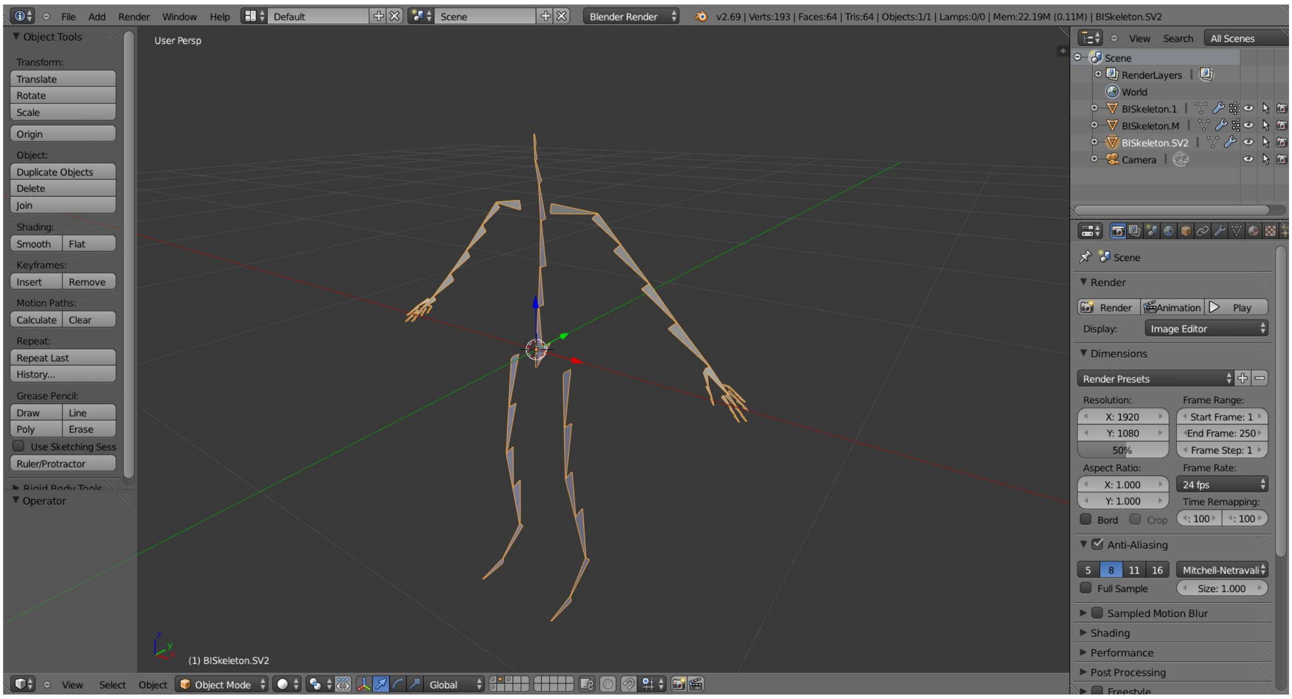
- ArmaRig by Maczer. At the time of writing, this is by far the most advanced rig available. His original rig won't work, but in the meantime he made an updated rig that works very well with the Toolbox.
- Blender Animation Rig by deanosbeano.
- There is also a thread by HJohnson on the BIS forum, but as far as I know, that one's using the same rig as deanosbeano.

Making your own Armature

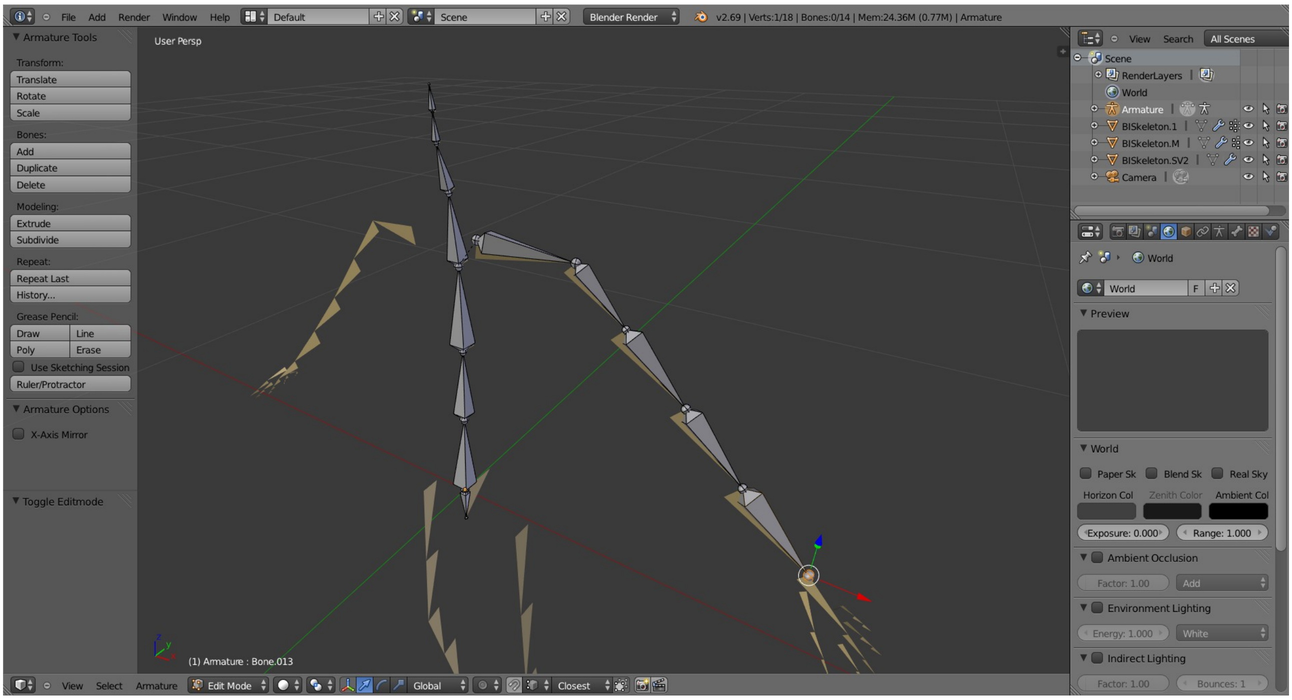
If you want your own rig, there are a few things to remember. The names of the bones are case sensitive. That means that any bone name must appear exactly as it appears in the Arma skeleton. For example, “LeftArm” is the name of the left upper arm bone of the standard BIS soldier skeleton, and it needs to be named like this. “leftarm”, “LEFTARM”, or “Leftarm” will not work.

If your rig needs control bones like IK controllers, it is best to start or end their name with an '@' character. This will prevent the exporter from writing them to the output file.

The easiest way is to import BISkeleton.p3d into Blender and check the “Edit” skeleton that will end up on the second layer (if you use default settings):



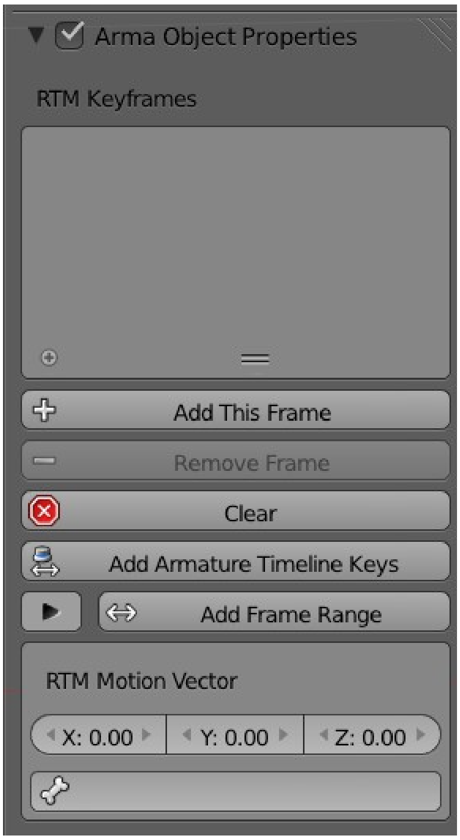
This is a geometry object, but it shows the “flow” of the bones, plus you can use the vertex positions for your bones. For that, enable vertex snapping, then press SHIFT-A to add a new armature with a single bone. Place the bone on the Pelvis in the center of the screen, then continue to extrude bones until you have a complete skeleton:



It is highly recommended that you name the bones as you go. Take care to use the correct spelling and capitalization of bone names.

Once you have created your rig, you can add other control bones and constraints. For animation purposes with Blender it is e.g. better to use a single upper and lower limb bone and use a copy rotation constraint on the other to prevent weird bending in the limbs.

Getting Started with RTM Animations



In order to get the option to export an animation, your armature needs to be turned into an Arma object just like with geometry objects by selecting the armature and clicking the checkmark on the Arma Object Properties panel. Note that the Arma Object Properties panel looks very different depending on whether your selected object is a mesh or an armature (see the picture to the right).

Since Blender's interpolation features are much more advanced than Arma's internal engine, the Arma toolbox allows you to define your own animation key frames that are independent of the armature key frames. In this respect, the term “keyframe” refers to the keyframes within the RTM file, not those defined on Blender's timeline.

The properties panel is divided into three parts. The upper part is the keyframe list. It lists all keyframes. The keyframe number in front refers to the frame on the timeline. The “time” value in brackets refers to the time index in the resulting RTM file. These time indices are always in the range 0 to 1 (representing 0 to 100 percent of the time used for the animation), and are calculated in relation to your animation timeline's start and end frame. Moving the start and/or end frame will update these values accordingly. Below that are buttons for manipulating that list (more on that below).

Finally, there is a field “RTM Motion Vector”. The motion vector, a part of the RTM file, is the vector by which the animated object moves during the animation. For a walk cycle, for example, this would be the length of one step.

This contains a three-part field with an X/Y/Z value, as well as a field with a stylized bone in front. The upper field only shows if the lower field is empty; if a selection is made in the lower field, the upper field will vanish since it is no longer relevant.

The upper field, if the bone selection is empty, will be written to the file as-is. You will have to compensate the motion of your rig yourself.

If you click on the bone field, you can select (or enter the name of) a bone from the armature. This bone becomes the center of the animation, and its movement, or rather, the resulting movement from the first to the last key frame (see below for the meaning of key frames) is the motion vector that gets written into the file.

Editing Key Frames

There are five buttons to edit key frames: Add This Frame, Remove Frame, Clear, Add Armature Timeline Keys, and Add Frame Range.

Pressing the “Add this Frame” button will add the currently selected frame on the timeline as a keyframe for the RTM file. The currently selected frame is the number on the center of the timeline's header (just right of the Start and End frame fields).

With any entry in the keyframe list selected, clicking the “Remove Frame” button will delete the key from the list. The “Clear” button will clear *all* the keyframes[3].

The remaining two buttons are a big more complex. The first one is labeled “Add Armature Timeline Keys” and will add the keyframes of the armature on the timeline to the list of RTM keyframes. Basically, this ensures that changes in the bones of the armature will always end up in the RTM file, and is quite useful since it saves you the hassle of having to add these manually.

The last button is called “Add Frame Range”. It has a split button next to it, meaning it has settings. If you click on the split button, the settings will be revealed.

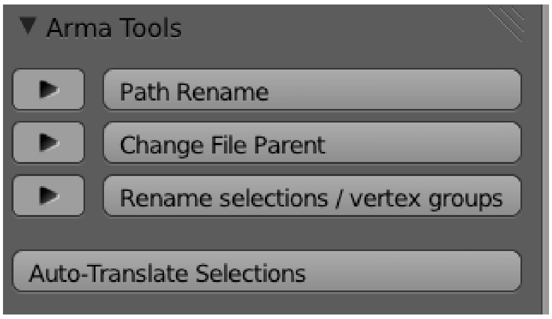
Basically, what this button does is to add a range of keys from the “Start” frame to the “End” frame in the operator button's settings, at the given step range. For example, setting start to “1” and end to “20” with a step of “5” will add the keys 1,6,11 and 16 to the RTM keyframe list. With a step of 1 and Start and End set to the start and end frame numbers of the timeline (the default), this will essentially add each and every frame to the list of RTM keyframes.

Again, please note that RTM Keyframes have nothing to do with any other keys in Blender. They are simply numbers that are used to write frames of bone transforms into the RTM file.

In order to write the animation to a file, see the chapter on Exporting RTM Animation Files on page 18.

[3] Note that deleting or adding keyframes in this list will have no effect to the keyframes defined for the armature in Blender

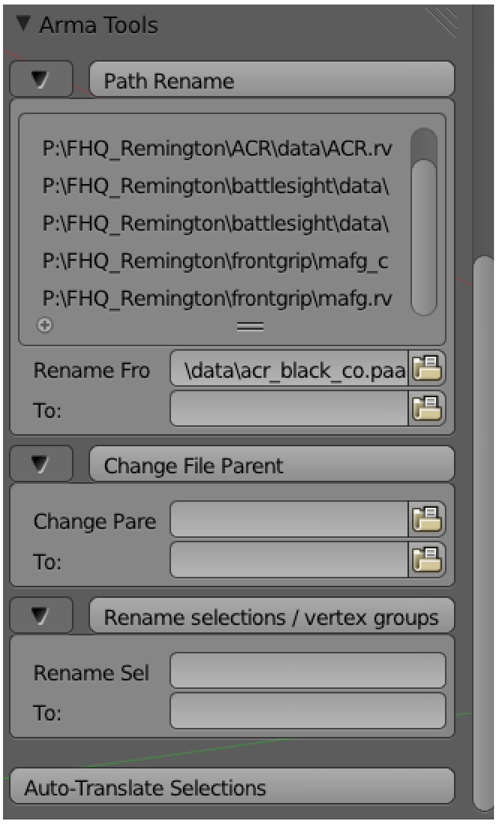
The Arma Tools Panel



Starting with version 1.3 of the toolbox, there is an additional UI panel in the “Tools” section of the UI. If you are using the (at the time of writing) development build of Blender, the tools panel will appear under a tab with category “Arma”. Under 2.69 and earlier versions, it will show up in the normal tool bar.

The image on the right shows the toolbox panel in its default state.

The arrows on the left side of the panel can be clicked to reveal the settings for the individual tools. For example, the “Path Rename” tool doesn't really do much when clicked, but opening the split arrow will reveal parameters for the function. When all split boxes are expanded, they look like in the image to the right. The panel might need more space, so if you use it you might want to resize the toolbar to be a bit wider than usual.



P:\FHQ_Remington\ACR\data\ACR.rv
P:\FHQ_Remington\battlesight\data\
P:\FHQ_Remington\battlesight\data\
P:\FHQ_Remington\frontgrip\mafg_c
P:\FHQ_Remington\frontgrip\mafg.rv

Rename Fro: \data\acr_black_co.paa
To:

We'll go over the tools in more detail now.

Path Rename

The Path Rename tool allows you to exchange the name of a texture or rvmat for a different one. In the listbox, all used file paths for textures and/or material files are displayed. Clicking on one of them will copy the name to the “Rename From” input box.

Click in the “To:” input box and type in a new name for the texture or material, then click on the “Path Rename” button on top of the panel to rename all occurrances of the “Rename From” name to the “To:” name.

Change File Parent

The “Change File Parent” tool allows you to exchange a prefix of a path on all texture or rvmat paths to another one. For example, suppose you have copied an MLOD that references texture path “ca\weapons\AK” and you want to rename it to something like “myweapons\ak_74”. You enter “ca\weapons\AK” in the “Change Parent” box, and “myweapons\ak_74” in the “To” box, and click “Change File Parent”. The result will be that all files that originally came from “ca\weapons\ak” will now reference the same subpath in “myweapons\ak_74”.

Rename selections / vertex groups

These two text fields allow you to globally rename all selections you enter in the “Rename Selection From” field to a new name you enter in the “To” field. The rename is global to all Arma objects in the current scene, and is case insensitive.

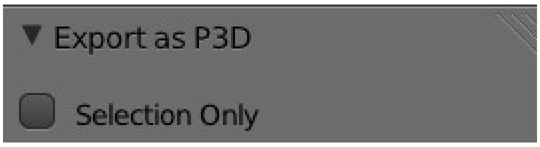
Auto-Translate Selections

This function has no options. Simply clicking the button will go through all selections/vertex groups in the current scene and try to replace the Czech names with English names. This currently limited to names that do not contain variable numbers – the lookup is static, using an internal table that contains official translations. In the future, I'll probably add the possibility to “fix” selections, like exchange the name of proxies for muzzle flashes and things like that.

File Operations

The meat of the toolbox are the import and export functions. As you would suspect, either can be found in the File->Import and File->Export menus.

Exporting a P3D



▼ Export as P3D

□ Selection Only

Exporting a P3D is done via the File->Export->Arma 2/3 P3D option. This will open Blender's standard file dialog. In the file dialog's options shelf, you will find another panel (you might need to collapse other panels to see it, or scroll down on the shelf). Starting with 1.5, there is only one option remaining. Ticking the “Selection Only” box will only export those objects that are flagged as Arma objects AND are selected when “Export” was invoked.

All other options were removed, they are either no longer needed (there is no bitxt file exported, the toolbox directly writes a P3D) or they haven't worked reliably.

Importing a P3D

The Import function is new in 1.0.1 of the Arma Toolbox. It is found under File->Import->Arma 2/3 P3D.

Note that Arma Toolbox can only import unbinarized files in the MLOD format. It cannot import binarized files (usually in ODOL format). If you have a binarized file, you should ask the author for an unbinarized version. In General, you should always have the consent of a model marker to use his or her creations.

Once selected, the import file dialog appears, which looks like a standard file dialog in Blender. Again, there is a special options panel at the left side of the screen. There is only one option right now (which is on by default): “Try to put each LOD on their own layer”. What this does is obvious – it tries to put each of the loaded LODs from the file on its own layer. Obviously, after 20 LOD's it will run out of layers, so it will start at 1 again. I usually keep my LODs on separate layers since it helps keep overview.

When Arma Toolbox imports a P3D, it names each of the objects it creates after a specific pattern. It starts with the base name of the file (say you import `M1_Abrams.p3d`, then the LODs will all start with “`M1_Abrams`”), followed by a period (“.”) and a one to four letter/digit code. For geometry LODs, this is the distance. For example, the highest resolution of above mentioned M1 Abrams would be called “`M1_Abrams.1`”, the second “`M1_Abrams.2`” and so on.

To find out what the codes mean, select the LOD and check their Arma Toolbox properties panel.

The importer will import all LODs, sharp edges, all UV sets, named selections, named properties, and mass settings. It will also import texture and material info and set up a number of materials automatically, one for each combination of face texture and material. These will be properly assigned to each face. In essence, exporting the file immediately should result in an identical file, within certain limits. The importer will also automatically add an Edge Split modifier with the right setup to the objects it creates, to ensure that the shading looks correct. Remember, do NOT apply this modifier at any time[4].

The naming of the materials may need some editing, since the names are chosen generically.

[4] If you should apply the modifier accidentally, you can simply select all vertices (using CTRL-A) select “Remove Doubles” from the Special menu. This should fuse all split edges again.

There are no blender-internal textures set up since Blender cannot read the paa files used within Arma. If you want to see the proper textures, you will have to convert them to PNG or a similar format and assign them yourself. A future version of the Toolbox might offer the chance to do this automatically.

Exporting RTM Animation Files

RTM files are always written in one of two modes – either as a static pose, or as a full animation.

For the purpose of this document, a static pose is an animation with exactly two frames, at time index 0 and time index 1, without any movement. Static poses are used e.g. for the hand postures on weapon models. If loaded into O2, the result will be three frames though: O2 always displays the time index -0.5 for the rest pose.

Exporting an RTM file is done by selecting File->Export->Arma 2/3 .RTM Animation from the menu. Note that this option will only be available when an armature is selected that has the Arma Object Panel checkbox checked.

The file select dialog allows you to enter a filename that should end with the .RTM extension. An export setting panel will appear on the left side of the file dialog (you might have to scroll/collapse other panels to see it). It currently has two checkboxes. One of these checkbox is labelled “Static Pose”. If this box is checked, or no keyframes have been added to the RTM keyframe list, the ***current pose*** of the armature is exported as static pose, regardless of keyframes and/or setting of the timeline cursor. Basically, what you see is what you get. Otherwise, the keyframes from the RTM keyframe list are exported to the RTM file.

The second check box is “Clip Frames”. It is checked by default, and will prevent frames outside the start/end frame area to be exported; basically, it will prevent any frames with a time index below zero or above 1 to be exported into the file.

Clicking “Save” will write the RTM file. There is no need to import it into O2, it should be ready to go.

Copyright and Credits

The Arma Toolbox is written by myself, Alwarren. I can be reached as Hans-Joerg@friedenhq.org, or as Alwarren on the BI forums. All code is written by myself, but I have to credit Leopotam, the author of the original Blender import/export scripts, since without his effort I would have never gotten into any of this and nothing of it would have ever seen the light of day.

The latest version of this package can always be found on our website, http://friedenhq.org.

Standard Disclaimers apply. Copyright does not extend to anything that this software outputs, so you retain the full rights of your own creations.

If you have any requests, bug reports, or ideas, don't hesitate to contact us.

Thank you to Maczer for testing the RTM exporter and his feedback.

Asking for Donations

We originally didn't want to do this, but decided to give it a try. The Web site now has a “Donate” button. We spent a lot of time on making these utilities, addons, and missions for Arma 2 and 3. We would like to think about making a small donation to encourage further work and acknowledge what we already did.