The purpose of this tutorial is to have a some information how to use nifscope written down. While for sceneviewer (aka nifviewer), the from Firaxis provided tool, already lots of tutorials exists, the same can´t be said for nifscope.

So here i try to give you an impression how to work with a tool, with may thanks to the niftools team for providing and creating it, A tool which is my eyes better than nifviewer and also allows you more things to do. It seems that i use the outdated version 1.0.8, but there shouldn´t be big differences the handling.

What you need for this tutorial:

installed nifscope
unpacked your Civ IV art and
nifviewer[1].

The links for this programs can be found in the appendix.

The tutorial will start with the basic in how to use the program and will cover all fields i know. Hopeful if somebody knows more about it, he will add the information. If later questions arise in thread i will cover them there, and perhaps if my time allows it and it´s worth, update this text.

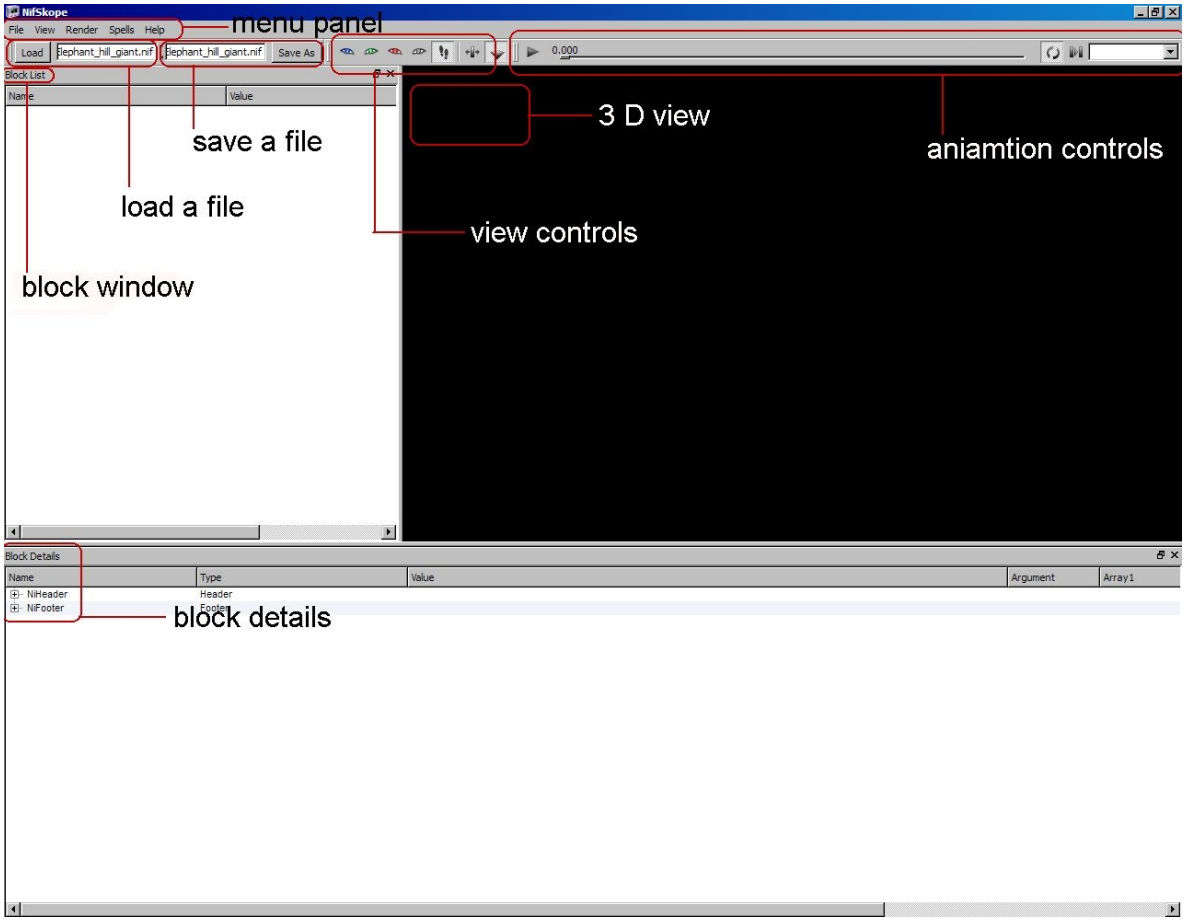
Ciao
The_Coyote

Outline:

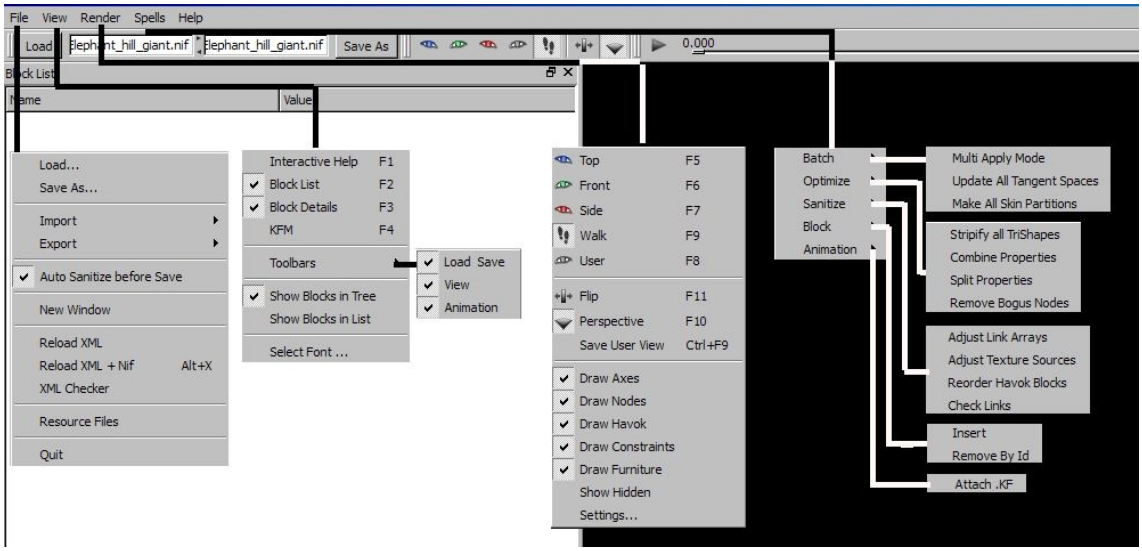
1 Yes, also nifviewer, you will later see why it´s good to have both

I. Basics
I.1. the interface and set up

So let start with the basics. After you installed all needed programs open nifscope. My looks this way



Most likely your fresh install will look a bit different. Because i use my structure and your aren´t used to use the program it´s a good exercise to change yours to fit mine. For this have a closer look at the menu panel.



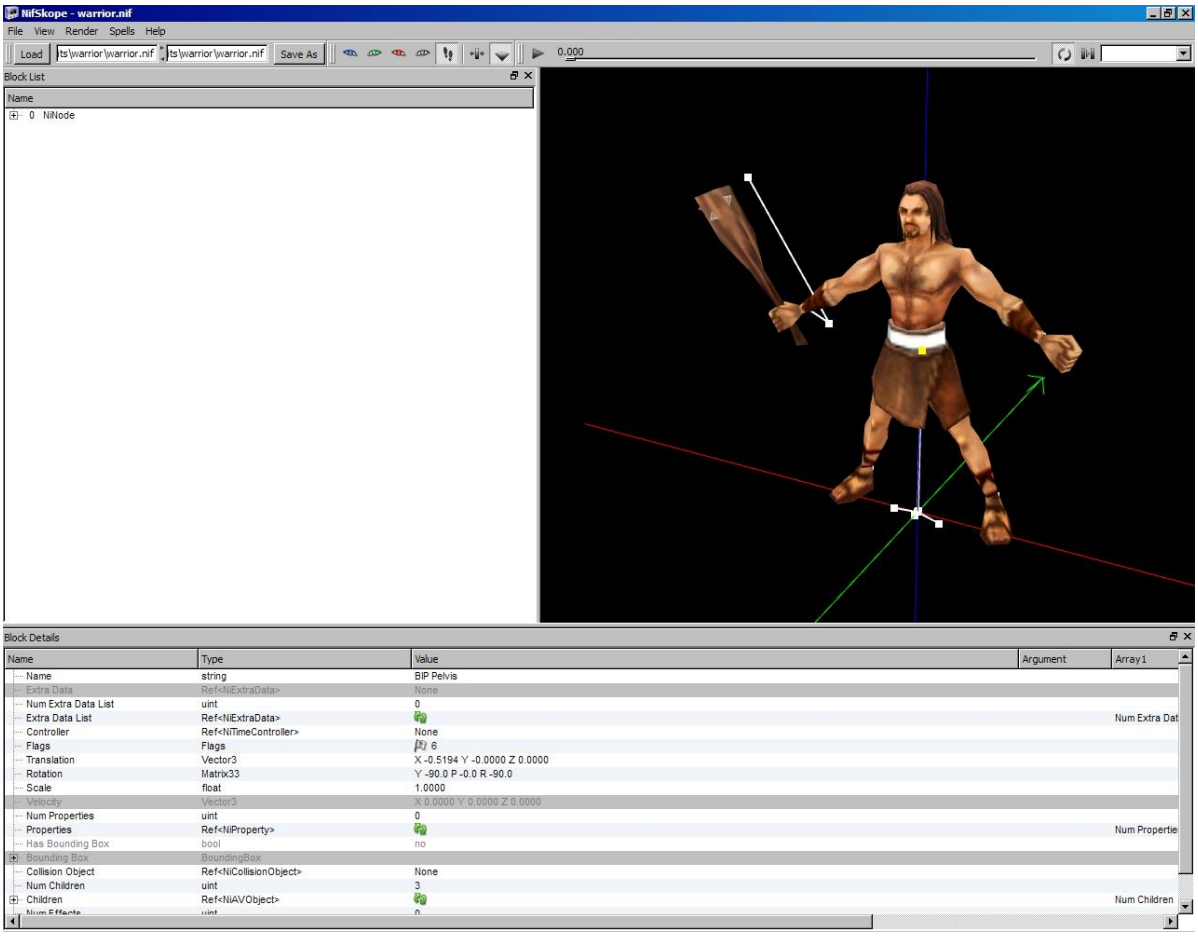
At first you can see which option i have activated, important are Block List and Block Details, both activate or deactivated the corresponding window. The difference between Show Blocks in Tree or in List is illustrated later[2]. I prefer the Tree setting. The settings under Render are needed later for nifs, they control which is shown. Most other options are not needed for the moment.

To rearrange your windows, go with the mouse on the border of two windows. Your are right if the cursor changes. Than press the LMB[3] and move the border line. If you click on the field name, eg Block List, and press and hold the LMB you can drag the window and attach it where you want, or at least you want and the program lets you. As next we want to play with models for Civ IV, so lets open a nif.

I.2. the structure of a nif

Before starting with the real work, some basics about nifs. The nif files includes all information of the model, this means the model (mesh) itself, the additional information for this mesh, the armature (skeleton) and helper, likes nodes for postions where later the effects will be played. Also there are information which will not interest you at the beginning.

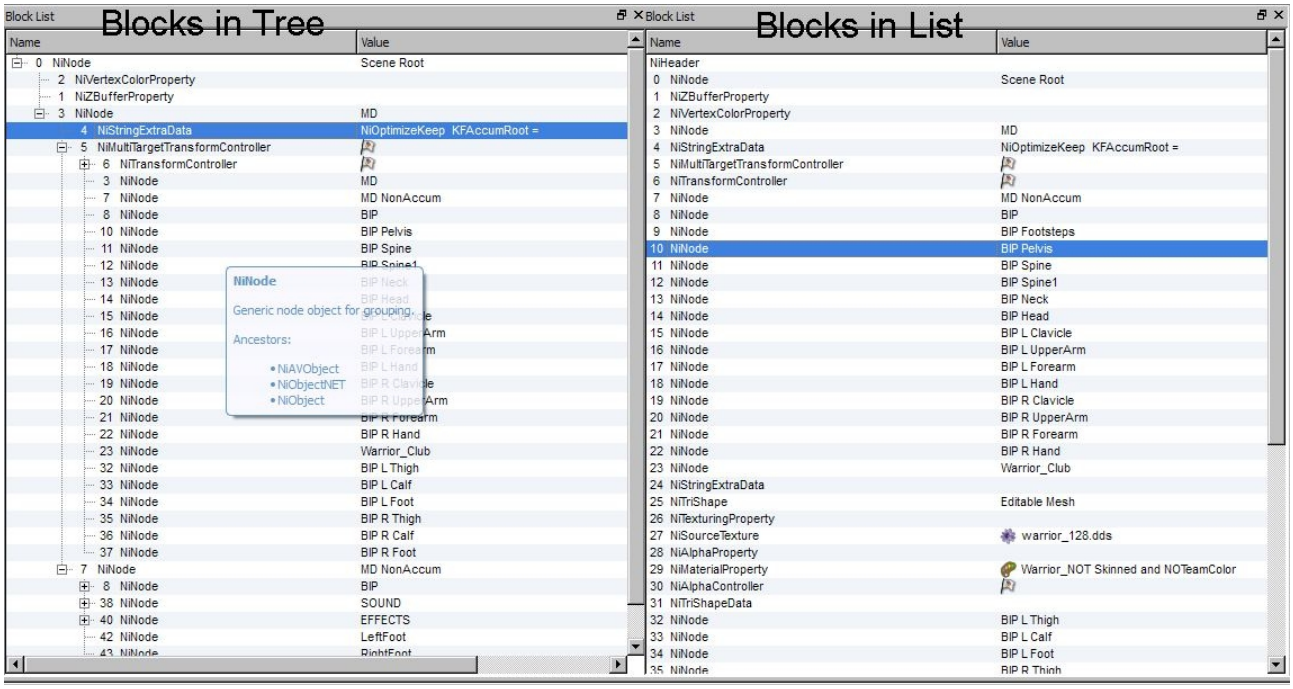
So how to open a nif. First if nifscope is the standard tool for opening nif files, double click a nif and finished. If not, go to File – Open and select the file you want. For this example open the warrior.nif[4]



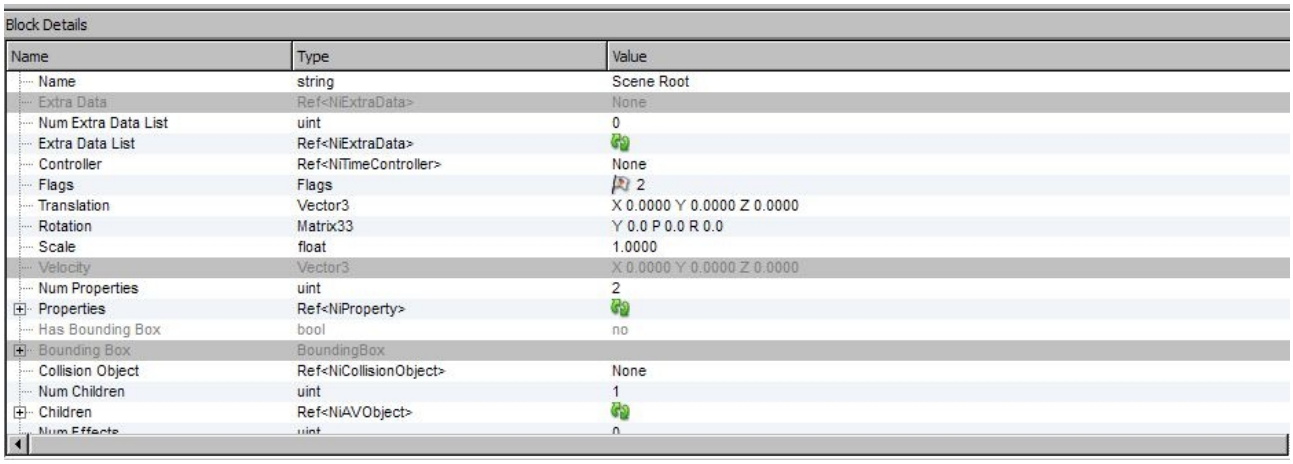
2 For this we need a nif
3 Left mouse button
4 We use a standard nif here, because custom models will look sometimes different

Perhaps what you will see will be similar to this or not. Do you have a black 3 D view? If yes be sure that your are in the walk mode, the feet icon in the view controls. Than go with the mouse in the 3 D view and use your mouse wheel to zoom in and out[5]. With pressed LMB you can rotate, with pressed mouse wheel (3. mouse button) you can move. Now scroll out until you see the warrior model and than rotate around him one time.

The lines in the 3 D view are the axes, green is the x axis, red the y axis and blue the z axis. Now we go back to the option Block in Tree or List. Here is a pic where you can compare both. Like said i prefer the Tree structure.



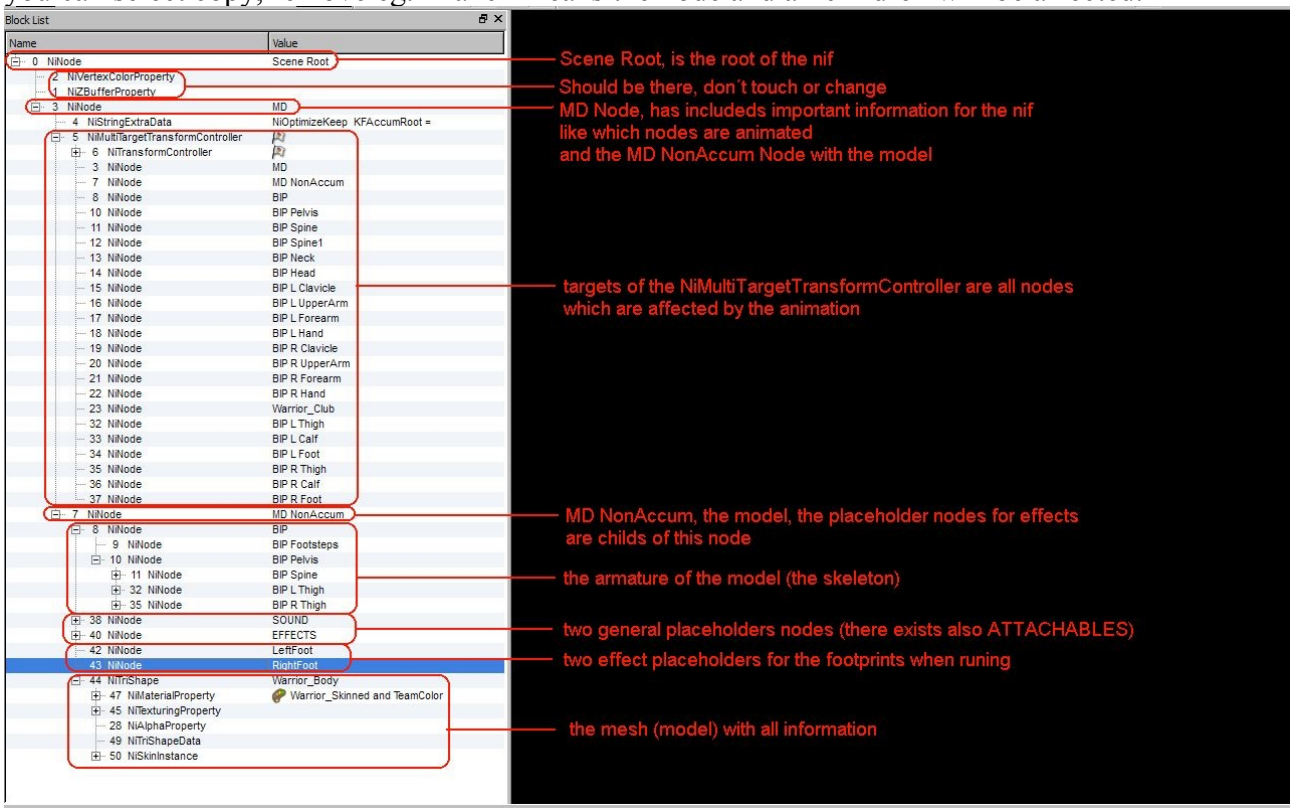
Now to the other two windows. In Block List you see the structure of all nodes in the nif, who is child from who, the number of the node and the name. In Block Details you see the information for the selected node, as example Scene Root



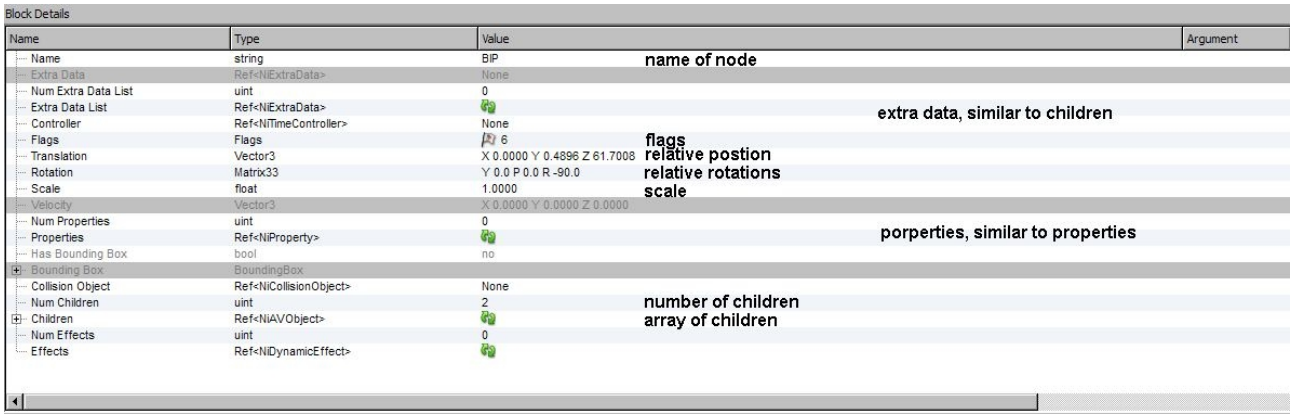
But lets start with the general outline of a nif. So have a closer look at the block list. First with the small plus or minus you can open or close a tree, The lines show you which is the parent of the

5 You can also use scroll up and down

node. So with this informations lets look at the warrior.nif again. With RMB on a node → block you can select copy, remove eg. Branch means the node and all children will be affected.



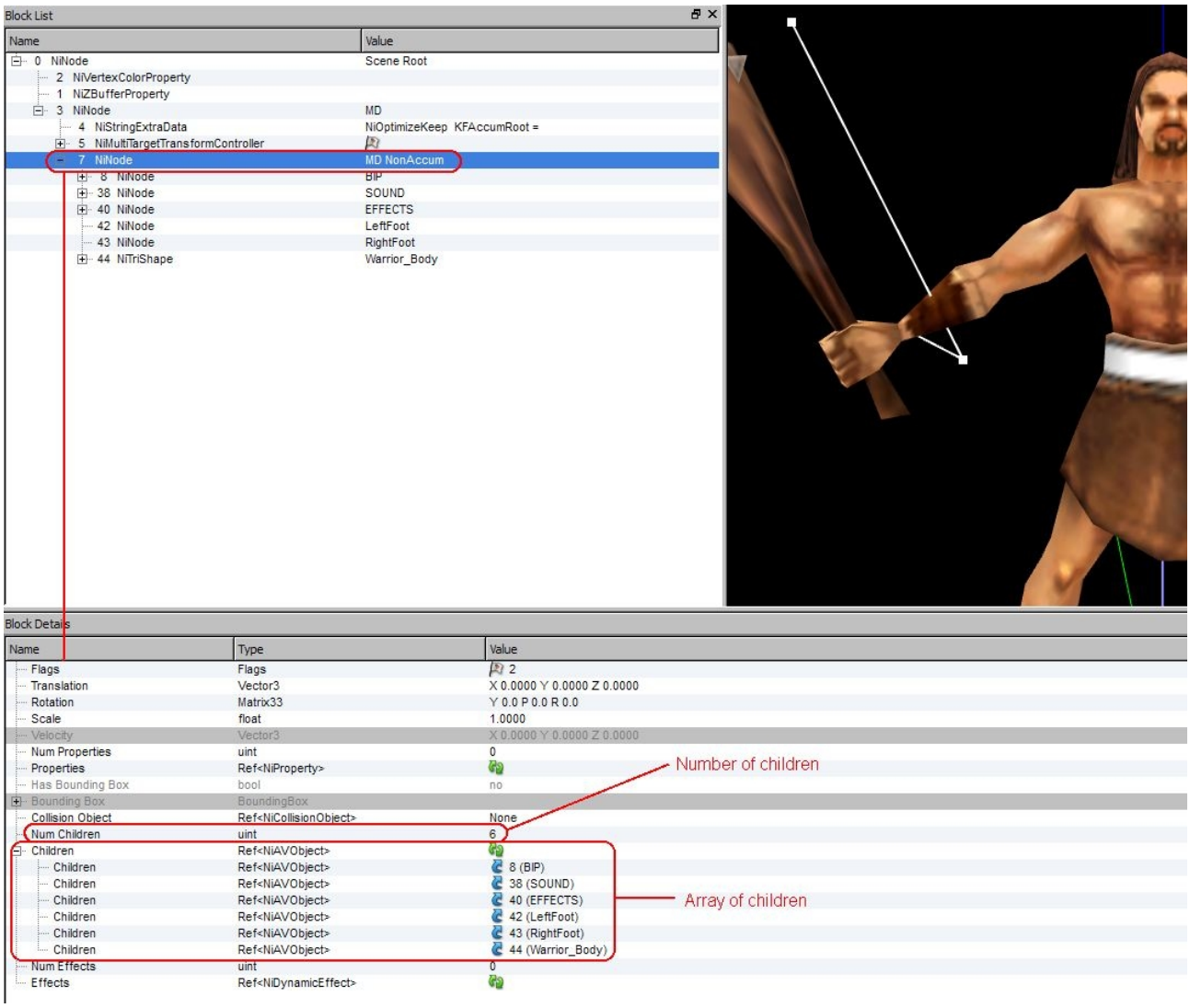
This is the general structure of a nif. Different nifs will look different, sometime there are more than one mesh in the nif, eg the rider and the horse at cavalry units, or the meshes are child of an additional node. But the basic outline will be the same. Now this doesn´t include information about the node, for this look at the block details.



You see the name of the node and the values, all non grayed values can be changed, simply double click the value. Try it with scale[6], see how the warrior gets bigger and reduce the scale again. If you click on the flag in front of the flag value and new window opens.

6 Name, positions, rotations and scale at general nodes can also be changed by RMB the node in the Block List, Transform and Edit. Other kind of nodes allows you others values to change

Now the position, it´s a relative value to the parent node, also considering the rotation. The same is for rotation. If you want to change the rotation, you have to options to enter, to change between the simply click on the field Euler or Axis. I prefer entering the values in the Euler mode, it simply an angle. As exercise try to change some values and look what happens in the 3 D view. After this restore the original values. I´m not completely sure what each flag values means, but they are important, there are different values for effect nodes, armature nodes, armature end nodes and models. Best look at a model to see the values.

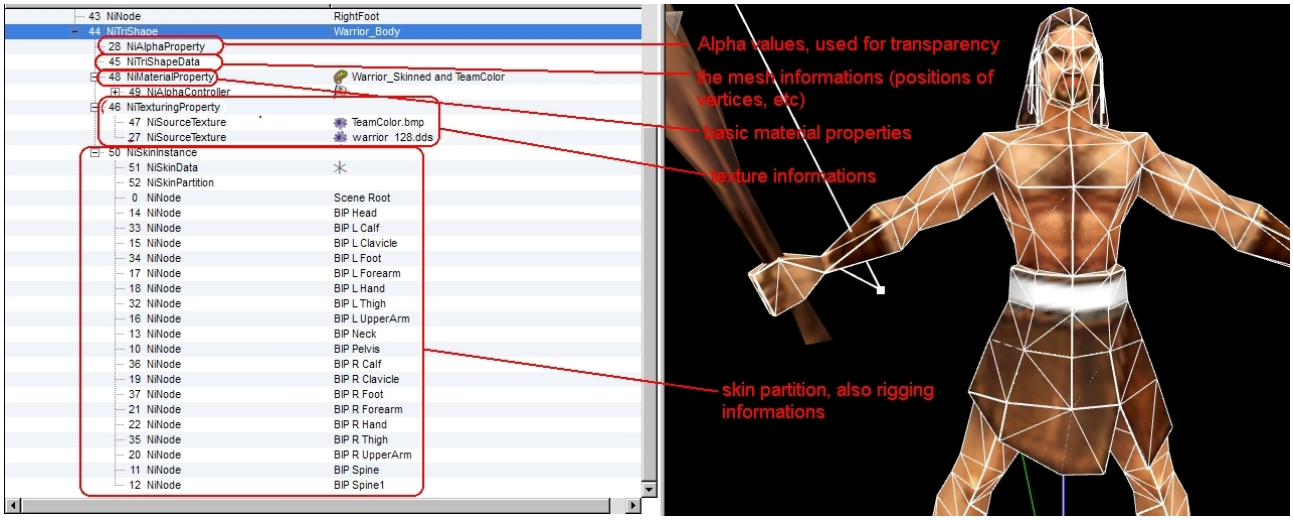


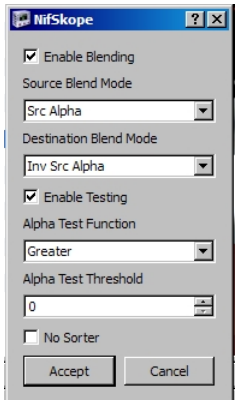
Next a basic and later often used method. How to add, change or remove children[7], first expand the "MD NonAccum" node. You can see that this node has six children. In the block details you can see a value called num children with the number six, the values says how many children this node can have[8], if you expand the children array you see a list with all possible children positions. Not every possible children link must actual have a child, but here it´s the case. The blue arrow is a link to the children, in general every time you see it and click on it you will select the node. If you click on the name you can change the child, if the field is empty, no child is actual assigned. Now try to remove the mesh, node 44 – warrior body, clicking on the name and delete the entry and press return.[9] Now you will instead of node name a none and the warrior will be outside the tree. Now

7 it´s the same for properties, Extra Targets and other similar arrays
8 Not has
9 Sometimes – not here - the num block return means new line in the field, the "main" return will accept the changes,

add the warrior again by again activate the field and enter 44, the node number of the warrior_body, which must be used in most cases. Be aware that sometimes the order of the children is important, the most important case I know is transparency. Now we want to add (or remove) a children position. To do this simply raise (lower) the number after num children. To see your changes you must update the children array. Simply click on the arrow circle or press RMB on children and select array → update.

Now we will look at the mesh itself. It´s a node called NiTriShape or NiTriStrips. The later should perform faster but will normally raise your polycount. So if the there is an huge difference use the first. You can easily change it by RMB on the NiTriShape (NiTriStrips) → mesh → stripify (triangulate). But now have a look at the warrior.





The alpha property is needed when you want ot use transparency, because the alpha layer of the textures decides if a part has transparency a mesh can only have transparency or teamcolour. This non shader model has teamcolour applied, easy to see because there is a texture called TeamColor.bmp. To edit the alpha property, press RMB on the node in the tree view or LMB on the small flag in front of the flag value in the block details. The left values worked for me in most cases, sometimes with a different Alpha Test Threshold.

The next interesting object is the NiTriShapeData (NiTriStripsData). In most cases you will do nothing there, but you can do things there. The one I used so far is to option to change the positions of vertices.[10] To to this expand the vertices array in the block details. Now you see all vertices of the mesh and their positions. If you select a vertices it is displayed yellow in the 3 D view. Enter the new position and finished. But I would recommend it only for small changes.

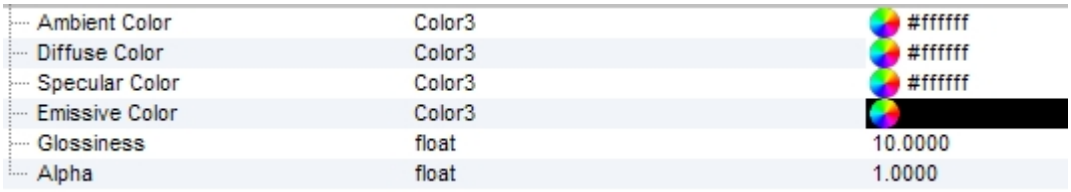
Next in the row are the material properties, here you can find the basic informations about the material color, darkness and gloss. If a model is darker than the texture or has a different colour, the reason could be found here.

This are the interesting values, also this are the values you see. If your unit is too glossy in game

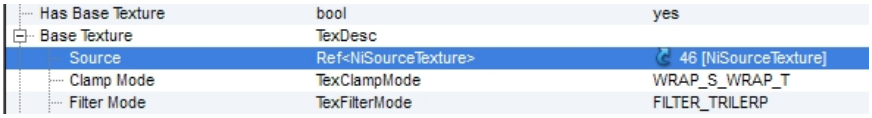
Esc will cancel the changes

10 To fix Bone Transplant errors or remove parts of a model, eg a sword

and has no shader, set the specular color to full black. With ambient and diffuse you can control the darkness and with the later also the colour of the texture.



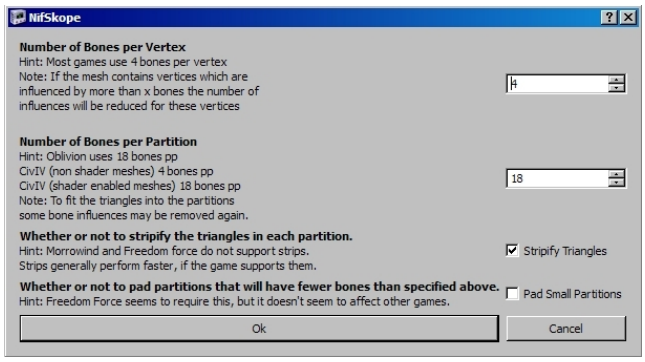
Now after the basic material properties are known lets looks at the texture properties. Here you can change the texture, add new textures or remove textures. If you want to know which slot is needed for which kind of model look at Refar great tutorial.



How you can add a new NiSourceTexture, the fasted way is to RMB on a existing node, select block → copy, RMB in the tree view outside the tree, block → paste. Now you have a copy of the NiSourceTexture, simply change the texture name, double click on the name after File Name and enter your texture name. To add this new NiSourceTexture to a slot, first activate the slot with the has X texture, expand the slot, after this mimic the values of a similar texture[11] in the X texture. Under source enter the number of the new NiSourceTexture. Finished, you have added a new texture to the model. Also if you press RMB on the NiTriShape and select texture → edit UV you can change the UV of the model, the used texture slot is the base texture and with the teamcolour texture in this slot it will not work. Dutchking showed this in one of his tutorials.

The last object marked above is the NiSkinInstance, if such a object exits you can add a shader to the model, if not it´s is not possible. Also it shows which are the nodes the mesh is rigged too, a very useful information if you copy meshes from one nif to another. Rigged means that certain parts of the model are affected by the movement of other nodes in the nif, eg which vertices should be moved if our warrior moves his BIP Head armature node.

A last important but in the warrior.nif not used node is the NiStencilProperty. Here you can select that a face is drawn two side. To add this property select the NiTriShape, RMB → Node → Attach Property → select NiStencilProperty from the list. The same way you can add other properties. Now select the NiStencilProperty and in block details change the draw mode to draw both.



11 The damage texture has different values, so if you want to apply a damage texture later, mimic the values of the damage texture you want to use

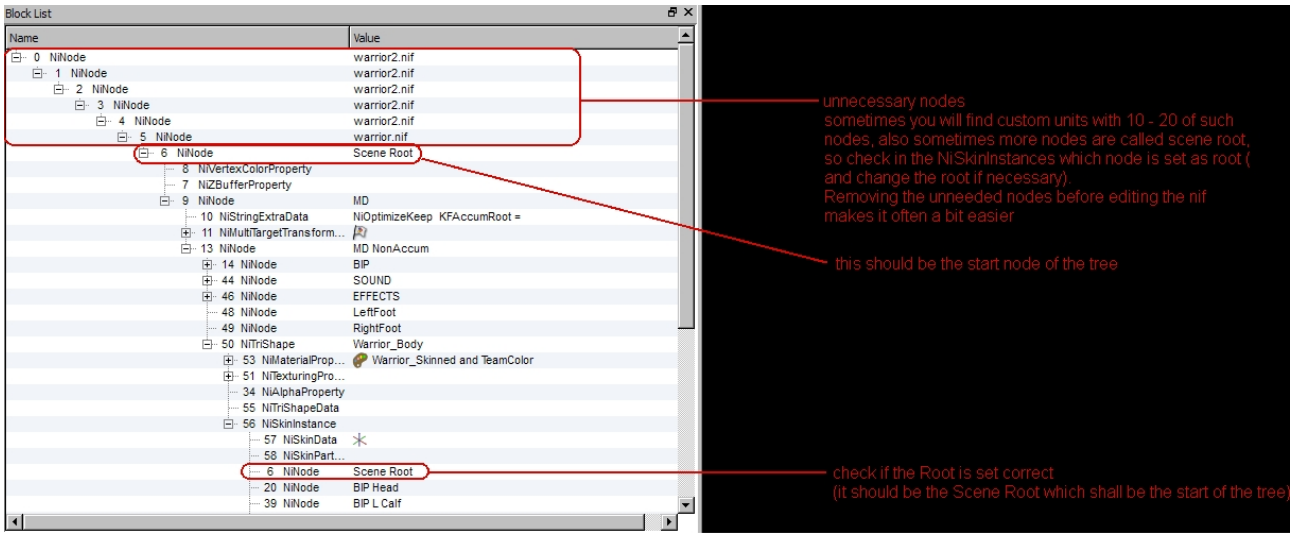
Like said, if a model has a NiSkinInstance you can add a shader. Shader models will perform better and will work on most PC today. Also if you read Refar texture tutorial you see a lot of things are only possible if you add a shader. But how to make a model a shader model. For this again select the NiTriShape, RMB → mesh → make skin partition.

Now use the option above and the model has a skin partition. After this go to the block details of the NiTriShape and scroll down till you see this fields.

Set has shader to yes, enter the name of the shader you want to use, be carefully, the exact writing is important, therefore if prefer copying the name from a model i know it uses this shader, and set the unknown integer to -1. Finally it should look like this.

Shader model have the suffix _fx, at least if it´s a Firaxis units. The nifs with freeze[12] are used for the frozen animation option in civ IV.

Example: When a unit is edited in nifviewer[13] and saved, every time the a new is added add the beginning of the tree. If you want follow this example but doesn´t have a custom unit with this "problem", make a copy of the warrior.nif, open this file in nifviewer and save it, open again the saved file and save it again. Do this about five time and before somebody asks, there is no deeper educational sense so far. Now open the file in nifscope. It should look like this.



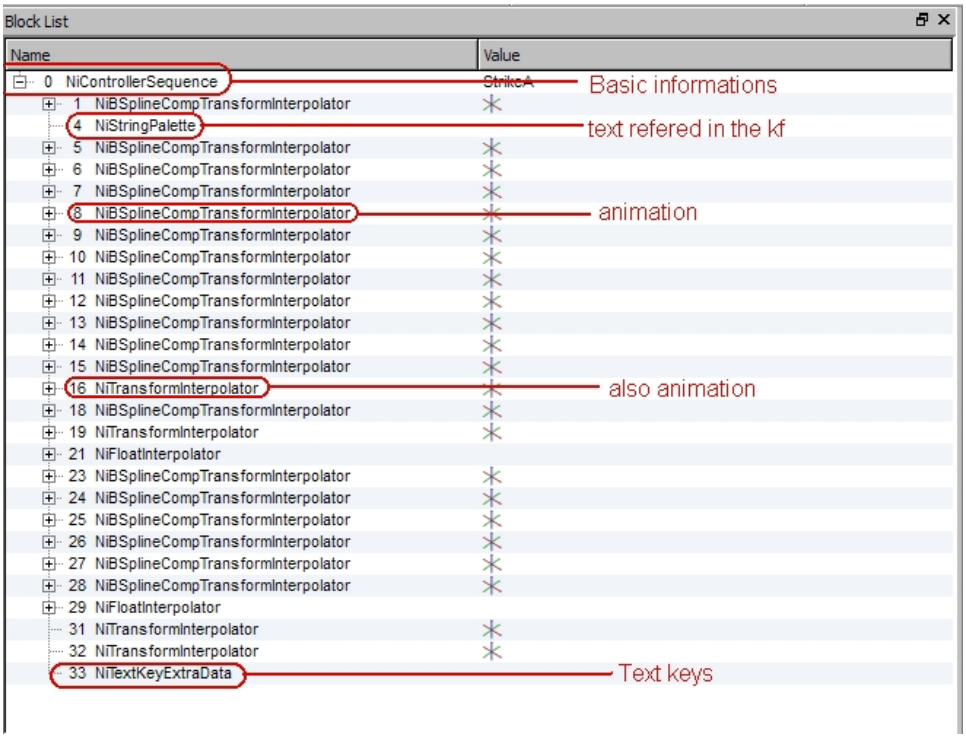
All fields you must check before editing are marked. Now if all is set correct, select the parent node of Scene Root, in this case it´s the warrior.nif node. Remove the child from the node and update the children array, now the Scene Root is outside the old tree. As last step before saving the nif, delete al unneeded nodes, the fastet way is to select node number 0, RMB → Block → Remove Branch (shortcut: ctrl + del). Now Scene Root should be the node with the number 0.

12 Most custom units will not have such files

13 There is a great tutorial for this program, more or less the reason why this piece of text exists

I.3. the structure of a kf

Kf files are the animations files for civ IV units, sometimes the reason why changes you did in nif files don´t work can be found there, Especially if you changed the position of armature nodes or scaled them. And remember idle is also an animation. For every animation sequence there is a own kf file. Look at the animation flowcharts Firaxis provided for more details. Also the are some files which are no covered here, there are special animations files for the damage stats and damage texture, also for applying or remove parts on units. For this chapter open warrior_md_strikea.kf.



First you will see that there is no model displayed in the 3 D view, which is correct. In this file are the information what the game shall do with certain bones and properties, not a model. But nevertheless the files affects the model. In the picture above you can see some nodes, the NiControllerSequence is more or less the master node of the file, here are the information what kind of animation or which node affects which bone of the kf.

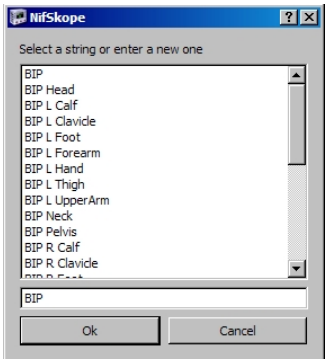
Name	Type	Value
Name	string	StrikeA
Text Keys Name	string	
Text Keys	Ref<NiTextKeyExtraData>	None
Num Controlled Blocks	uint	25
Unknown Int 1	uint	1
Controlled Blocks	ControllerLink	
Weight	float	1.0000
Text Keys	Ref<NiTextKeyExtraData>	33 [NiTextKeyExtraData]
Cycle Type	CycleType	CYCLE_CLAMP
Unknown Int 0	uint	0
Frequency	float	1.0000
Start Time	float	0.0000
Stop Time	float	0.7000
Unknown Float 2	float	0.0000
Unknown Byte	byte	0
Manager	Ptr<NiControllerManager>	None
Target Name	string	MD
String Palette	Ref<NiStringPalette>	4 [NiStringPalette]

Most important is the quick link to the text keys, the cycle type, because if you look at the

flowcharts you see that an animation can have clamp or loop and the start and stop time. The rest is also important, but more or less only if you make your own animations in Blender. The part which you perhaps will is controlled blocks, here are all block listed which are affected by the kf, adding a new block works like adding a new children slot.

Controlled Blocks	ControllerLink	
Controlled Blocks	ControllerLink	BIP
Target Name	string	
Controller	Ref<NiTimeController>	None
Interpolator	Ref<NiInterpolator>	1 [NiBSplineCompTransformInterpolator]
Controller	Ref<NiTimeController>	None
Unknown Link 2	Ref<NiObject>	None
Unknown Short 0	ushort	0
Priority?	byte	0
Priority?	byte	0
String Palette	Ref<NiStringPalette>	4 [NiStringPalette]
Node Name	string	
Node Name	string	
Node Name Offset	StringOffset	Txt BIP
Property Type	string	
Property Type	string	
Property Type Offset	StringOffset	Txt <empty>
Controller Type	string	
Controller Type	string	
Controller Type Offset	StringOffset	Txt NiTransformController
Variable 1	string	
Variable 1	string	
Variable Offset 1	StringOffset	Txt <empty>
Variable 2	string	
Variable 2	string	
Variable Offset 2	StringOffset	Txt <empty>

This is a detail look on the informations you can see there, the interpolator is a link to the node where you can found the transform information. String palette is the number of the string palette. All entries after this are marked as txt, which means you must enter plain text there. To edit the values, RMB → Edit String offset.



The left pop up will open. In the upper part you see all text strings which are already in the string palette. If the text you want is not there, enter in the line below the list the text you want and press ok. The text will be accepted and also added to the string palette.

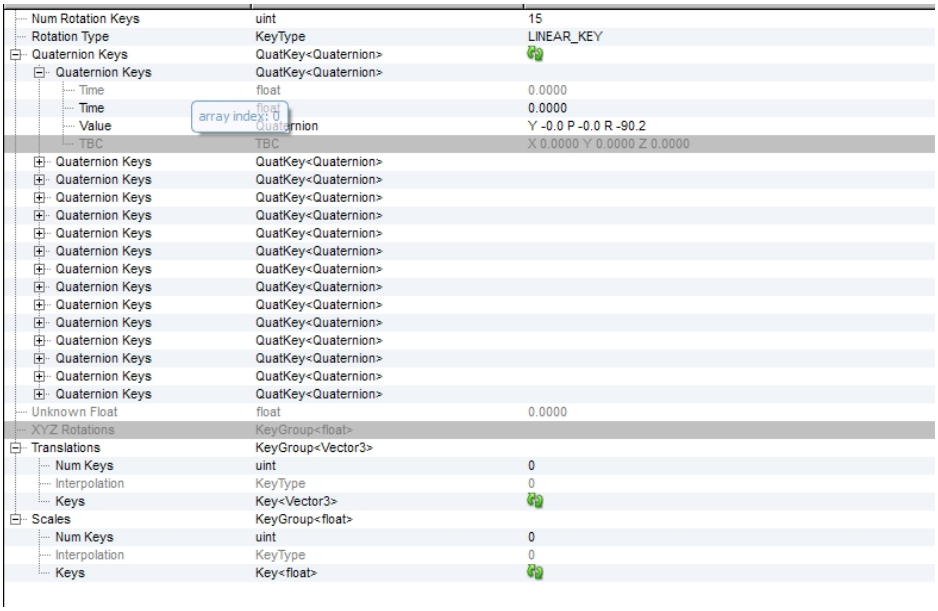
The interesting values are Node Name Offset, here is the name of the node in the nif which shall be affected by the interpolator. The name must be exact the same, it is also case sensitive. So one option for you if you want the remove the animation of an node, change the node name offset in the kf here. The Controller Type Offset must be set, here it´s NiTransformController because it is animations information for an armature node. Different types of effects will have different controllers, but when you start playing here you should already have a bit experience.[14]

The next two marked nodes are two animations nodes. At first the bad news, the first type, NiBSplineCompTransformInterpolator, is very unfriendly to change. You will only have reduced success when you want to change something there. The second, NiTransformInterpolator, is more user friendly to edit.

Translation	Vector3	X 17.4980 Y 0.0000 Z 0.0000
Rotation	Quaternion	Y -0.0 P -0.0 R -7.0
Scale	float	<float_min>
Unknown Bytes	byte	
Data	Ref<NiTransformData>	17 [NiTransformData]

14 Or in other words, it´s not part of this tutorial

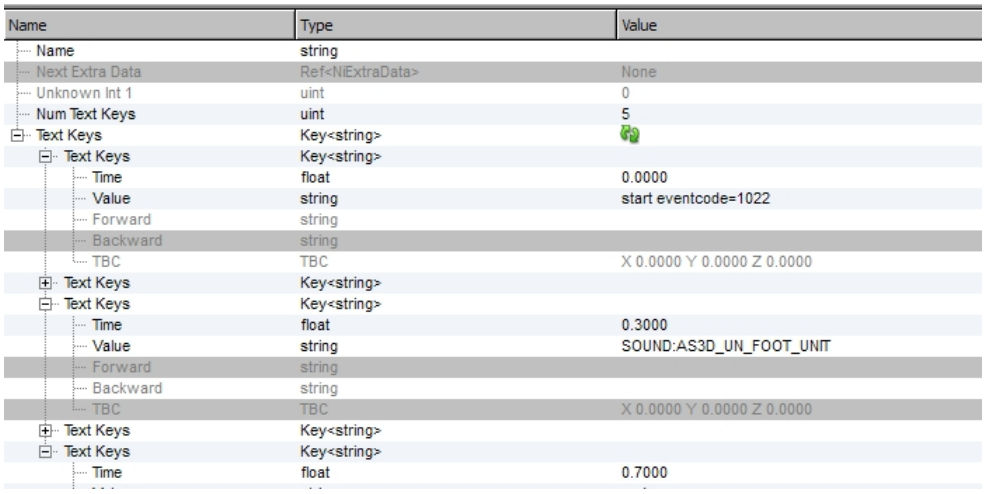
Under translation you see and can change the start position of the node in the animation, rotations the rotations and scale the scale. Data says if there is more animation information beside of this for this node available.



There are three parts, the rotation keys, the translation keys and the scale keys. This node only has rotation keys, which means all value changed above in the NiTransformInterpolator beside the rotation will be constant. If you expand all quaternion keys, you see the new rotations value and the time when this position shall be reached.

If there are informations here, they will be the ones which will be used in game. So if you change the scale in the nif, but the scale is also set in the kf, you will not see your change in game, if you set new values NiTransformInterpolator, but there are values in the NiTransformData for this values, you will also not see your change in game. As last hint, if you change the MD or MD NonAccum nodes, remember every rigged model is affected twice, the model itself is a child of this nodes and the armature also. Something which you will only see in game, if you test an animation in nifscope you don´t see the error.

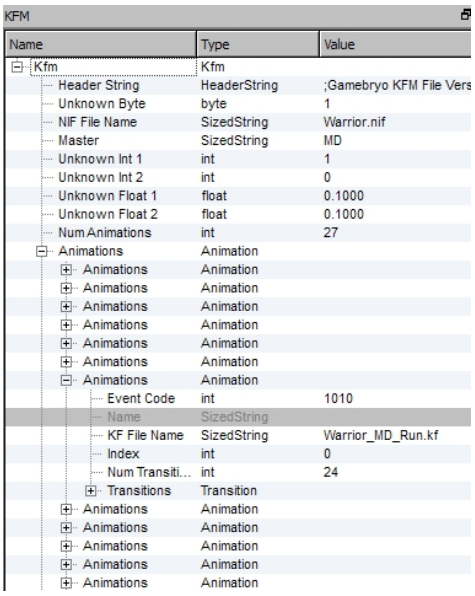
The last marked node in the kf was the NiTextKeyExtraData. Here are special information for the animation, eg which and where an effect shall be displayed or played.



The text keys look similar to the quaternion keys before. Adding a new also works the same, raise the number and refresh. Every animation must have (?[15]) at least two keys. One at time 0 with start eventcode = intern number of animation and one at the end time with end. The text keys between controls which effects are played. Above you see how a sound effect is activated. SOUND is the node where the effect is played, if you look in the nif you will also see a node SOUND. The text behind this is the intern code of the the effect. You can look at the sound xmls for a complete list.

Visual effects have a different structure: EFFECT:name_of_node:intern_effect_name. Also here you accept your changes with the main enter, the num block enter brings you in a new line, so there can be more than one effect to a time. I would recommend to copy the entry from an unit with the effect you want and change the node name. Sometimes it will be necessary to adjust the scale and / or rotation of the node in the nif.

I.4. the kfm file



The last missing file is is the kfm file. It´s a link file between the nif and the kf files. It´s says which structure shall be used, also has a link to a master nif and which animation file is used. You can open a kfm file in nifscope but if you want or must edit the file you must use the kfm editor. But you can use nifscope to check some values. Open the warrior.kfm. At first you will see it looks like you have opened the nif, you see the model and the tree. If you don´t see the unit when open the kfm you know that the master nif for the kfm is missing. To check some values activate the kfm window. [16]

Nif file name is the name of the master nif, under animation you see all animations listed with intern code and file name. Here you can check if the code are correct and or the file name are corrects. But like said before, for editing the kfm use the kfm editor.

I.4. nifscope , nifviewer comparison

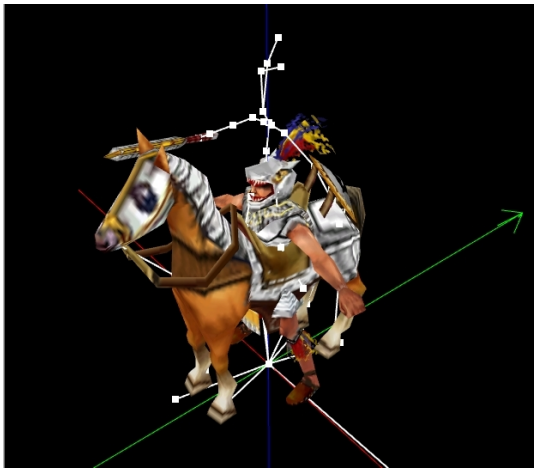
At first both programs are good, more a less it´s a question of personal taste which program you prefer. I still use both, but mainly nifscope. In certain cases nifscope will not display the unit correct, eg with a glow texture, darkness or if its a unit with two scene roots because of nifviewer bone transplant, but nifviewer will. In nifviewer you have faster an overview about the complete polycount of your unit, also if you apply a shader you can select the name from the list.

On the other hand, nifviewer crashes much more often than nifscope, and certain options in nif editing can only be done in nifscope.

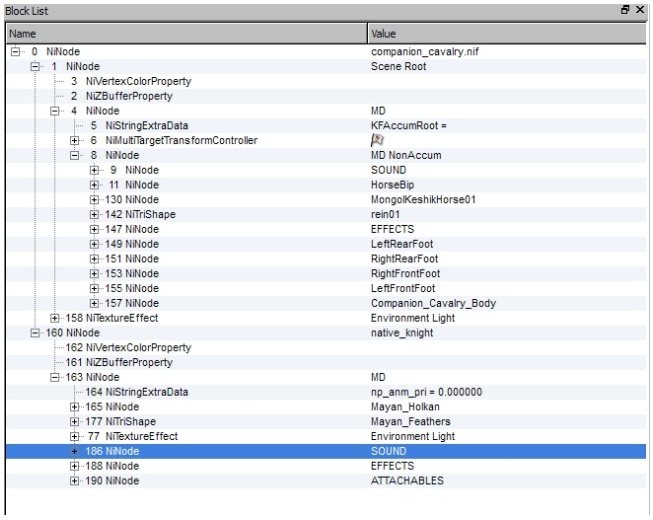
Now two special cases as examples of the mentioned differences, at first a pic from a custom unit[17] in nifscope on the left and in nifviewer on the right. You see, that in nifviewer the unit looks ok while in nifscope the rider is in the horse while the armature is above the horse. Now you can

15 Not checked so far
16 Look above or press F4
17 Native knight, unfortunately I don´t know the author

imagine it will be a bit hard to find the correct position eg of the weapons in nifscope.

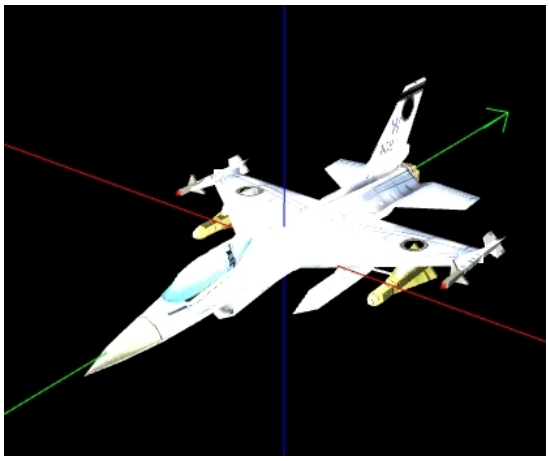


The unit was created using bone transplant in nifviewer, so if you have such a case if you open a custom nif, look at the node tree. You can see that the nif has a scene root, in this part of the tree is armature of the unit. Than the is a node called native_knight, in this part of the tree is the model. So the model is not in the same part of the tree how the armature. Also if you look at the armature you will see that certain armature nodes, the ones from the rider, have a node as child with the same or a similar name. This is a result of the bone transplant.



While the unit itself works fine in game there can be problems when you try to open such a file in Blender. In chapter two will be a quick explanation how you can fix this.

The second pic shows the limits of the 3 D view. Open the jetfighter_fx.nif, the unit will look ok. Now copy a file called 'environment_fx_greymetal.dds' from the shared folder in the folder with the jetfighter_fx.nif and open the file again. Now you should see something like this.



Now the model looks too bright, the texture is no longer really visible. Bad if you want to check the result of your reskin. The „problem“ is the file you copied in the folder, it is the glow texture of the unit, and this is the way nifscope displays the glow texture. The picture one the right shows the same file in nifviewer, looking much better.

The solution is easy, if you want to check you texture in nifscope and the unit has a glow texture, remove as long as you work the texture from the folder, or check your texture work in nifviewer. I do the later, one of the situation i still use nifviewer, because the unit is displayed more like it will look in game, because nifviewer uses a similar grafik engine.

The positive side of the way nifscope displays the model is that if you add a skin partition to an unit to add a shader, the nif will not crash. Trying to open such a nif in nifviewer before setting the shader can have as result a crash.

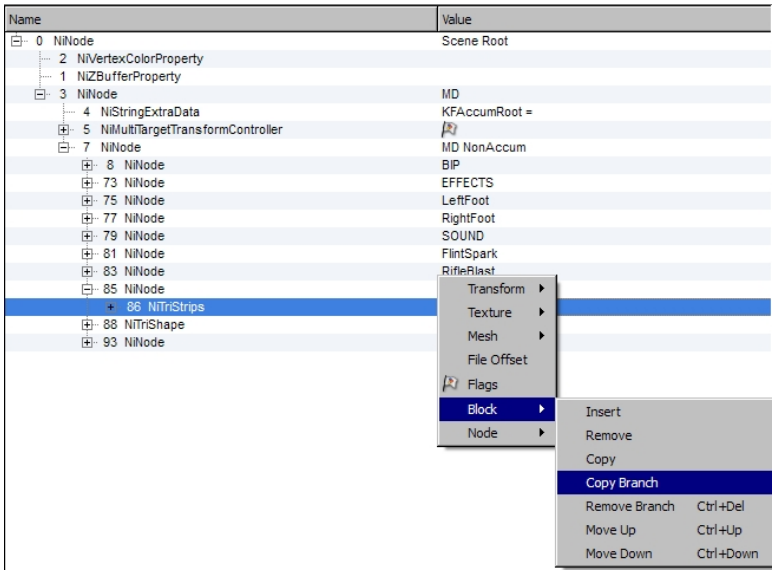
A last difference, nifscope supports non uniform scale, RMB on NiTriShape -> transform -> scale vertices, only for meshes, so if you want to scale an effect non uniform you have to use nifviewer.

II. Easy Jobs

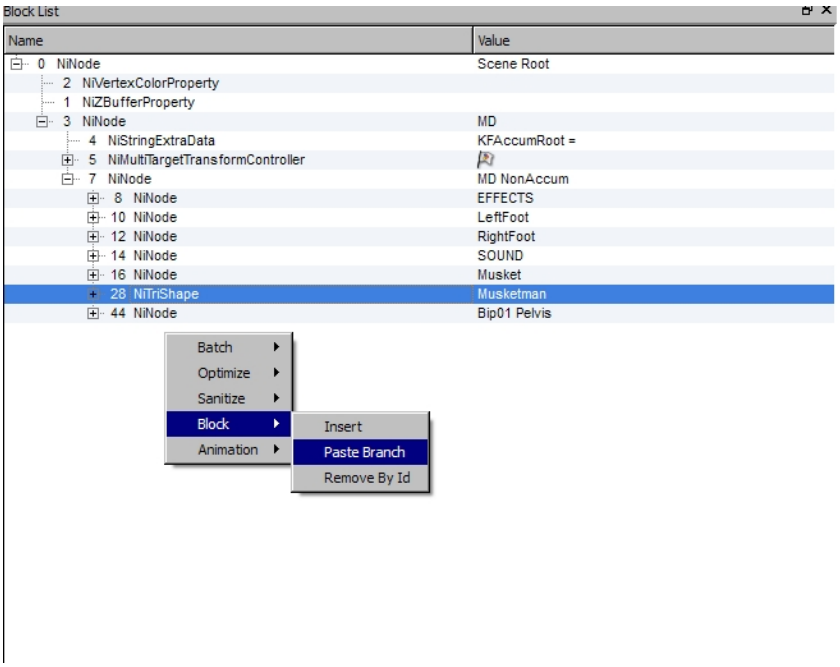
Now will will beginn with some real nifwork. I will try to use examples for the things i want to explain. Beginning with the simple exchange of parts like helmets or weapons, going from exchanging the model mesh in easy to more complicated cases.

II.1. Copy a model

Our target is to change the weapon of the musketman to the one of the civil war scenario militia. Open both nifs with nifscope. As next step select the milita weapon in the 3 D view. Now the mesh node is also selected.



RMB on the NiTriStrips, the actual weapon mesh, block and copy branch[18]. After this activate the musketman nif



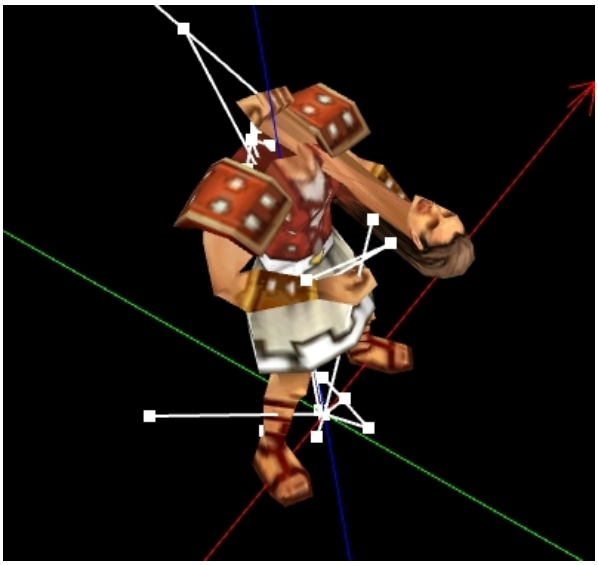
18 Remember, copy branch copy the node and all children, properties, eg of this node

RMB in the white space below the tree, block and paste branch. Now the weapon is in the nif. The final steps are, make the copied weapon a child of the musket node, the node where already the weapon is, and adjust the position[19]. Rename the mesh to the name of the standard musketman weapon (Editable Poly@#5 to Editable Mesh) and remove the old musket from the nif and save your result. As next step look at the idle animation[20] and see how our modern irregular holds his weapon on the bayonet. It should be clear that you also have to copy the weapon texture in the unit folder if you want ot use the model in game.

The same method can be used to exchange other separate parts of units hats, like swords or shields, but also for other things, like adding shoulder armour to a mesh or exchange the head. Search the node which fits bests and add the mesh there, or in the case of the head[21], reduce the size of the head node to 0.01 and copy the separate head mesh to the head node and change the scale to 100. This is also the reason when you copy a head added this way from one unit to another with standard head scale, you will see a giant head.

Now after copying a part we want to use an existing model with another animation, in our cases the swordsman model with the crossbowman animation. Open both nifs, select the swordsman NiTriShape and copy branch[22]. Go to the crossbowman and paste the branch. Also exchange the crossbowman mesh with the copied swordsman mesh, save and test with an animation. All seems to work like it should.

In this case it´s true, but life it´s not that easy. The reason why it worked, both nifs have (almost) exact the same armature as base, when you copy a rigged mesh from one nif to another nifscope tries to assigns the rigged parts to nodes with the exact same name. If a node doesn´t exist the copying will fail. To see what i mean, rename[23] before you start copying the swordsman mesh in the swordsman nif the left hand to head and the head to left hand. Now copy and look at the result.



It worked, but the result looks a bit strange. As next try to rename the head to head2 and try again to copy. Now you will see there will be an error.

19 You will see rotation is a bit tricky, but a good solution could be [Euler]: 7,0,-180 and as z-value 23

20 Save before looking at an animation, because open an animation will add objects to the nif we don´t want (NiTransformController) in a nif, because they should be in the kf and deleting them afterwards by hand will be pain

21 Which must be a separate object

22 The model could be white, it´s because of the missing textures

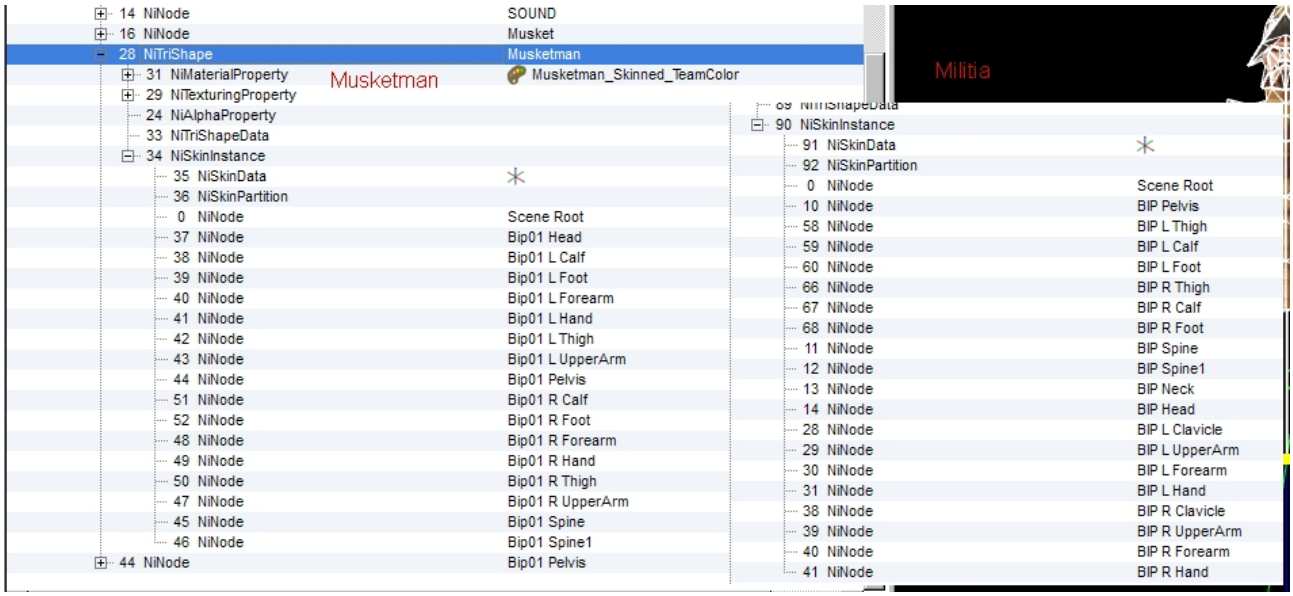
23 Can be very easy done in the NiSkinInstance

II.2. Bone transplant

You will see there exists a lot of different armature, some with neck, some without, some called BIP, some called Bip01, some have clavicle and other will have even more individual nodes.

If the general structure is the same, all nodes exists in both nif but in one nif they are called BIP and in the other Bip01, rename the nodes of your base, not the target, nif before you copy the mesh. With luck it will work. Also there can be problems even if both have the same structure, try to make the scout the gunner of the canon unit. The copy will work but the result looks a bit different from the original scout.

A solution for this is bone transplant, which means you copy all or parts of the nodes a mesh is rigged to in the target nif before you copy the mesh. The reason is, the animation controls the nodes in the nif, so changing their values direct will not work, but if you make the node the model is rigged to a child of the animated node, you can adjust slightly the scale and the position of the node while they is still affected from the animation. The steps are the same like before. To give you a small example open again the militia and the musketman.

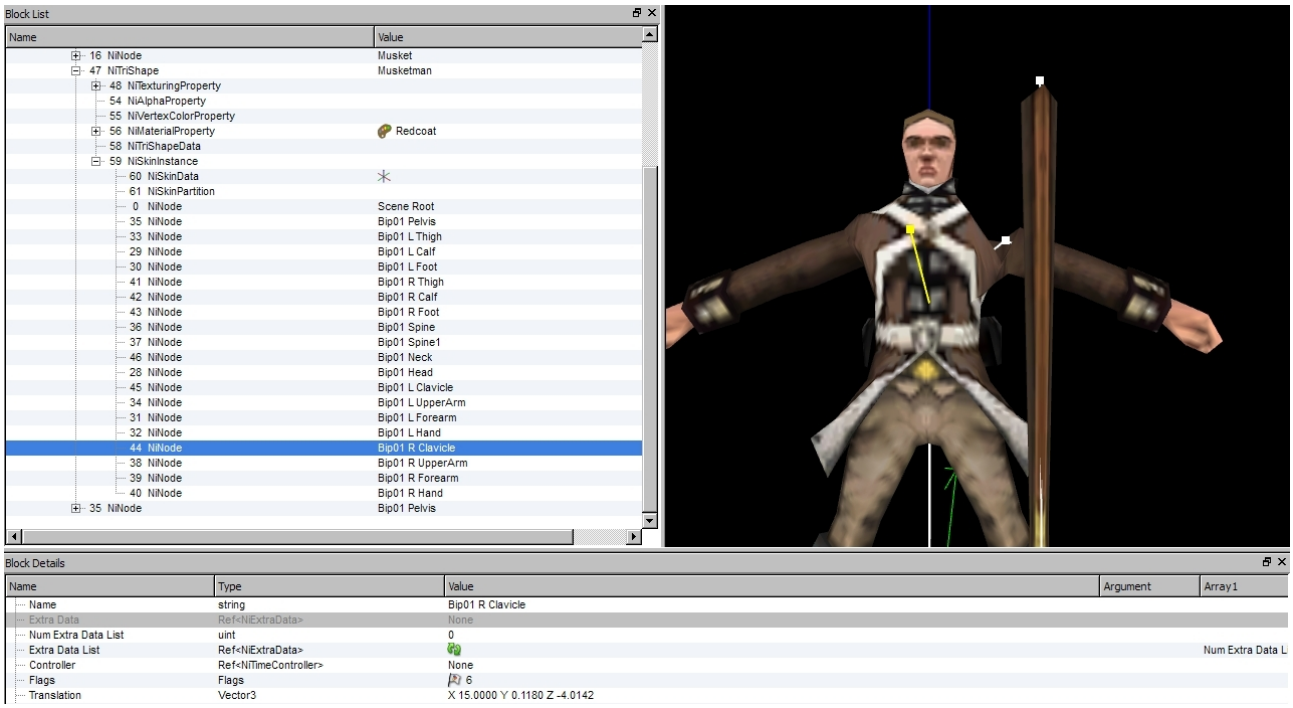


If you compare the armature structure you see that the militia armature starts with BIP while the musketman armature starts with Bip01. Also the militia has bones that doesn´t exist in the musketman armature, neck and both clavicle. So first exchange the BIP in all nodes in the militia nif with Bip01. After this is done we must copy the three nodes. First look which is the parent of the node, so a good guessing would be to make them children of spline1 in the musketman nif. Copy, only copy and not copy branch, the three nodes, remove their children[24] and make them children of spline1. Now copy the mesh and exchange the models. It looks almost good beside the clavicle parts of our model.

But before we fix this copy the hat from one nif to another. The reason for the clavicle problem is the position of the clavicle nodes, the two we have copied. Compare the position in both nifs and you see the difference, the reason is that the nodes have a relative position to their parent node. And in our case it´s not the same. In the militia nif the parent node of the clavicle was the neck, in the

24 Because they had children in the original nif the get also children in the new nif, nifscope choose the one with the same number

musketman nif we used spline1. But our error can be fixed easily, set the x value of both nodes to 15.



A full bone transplant means that you copy every node to the matching node in the other nif, set the position and rotation of this node to 0,0,0, add the missing nodes, copy the mesh, and now adjust the position and rotation of every node until the model looks ok. If you want to see such a result download my civil war cannons. Be carefully with the names, if you want to make a bone transplant of a unit to a nif whose armature has the same names, change the names before you copy. Remember nifscope uses the names and two equal named bones can cause trouble.

Please remember that the above tips are for vanilla units or units with a similar structure. There are a lot of custom units out that are already a result of bone transplant[25] or have a new head. So if you work with such a unit, have a closer look at the structure of file before you start. To which bones the model is rigged, which are the parent nodes of this nodes, are the nodes missing or special in one of the nifs. Are there parts of the models which are a child of a bone and not rigged one. If you know this, start to copy all needed bones, look that you find similar parents and / or adjust the position. Now copy the mesh itself and afterwards all parts you need.

II.3. model fixes for Blender import

Like already said at the end of chapter one there can be problems with customs units and the import in Blender, also i know at least one vanilla unit that makes trouble. To limit the possible error reason, delete all unneeded parts in the nif, like weapons, shields, etc, before you try to import a problematic file. If you are lucky after this you can import a file which before caused trouble.

But lets start with custom units which were created using bone transplanted and nifviewer. This files can have two scene roots and at least this surely causes import errors. The solution is quite easy. Open the nif and check to which bones the model is rigged and that this names differ from the animation base armature. If not, change the name of the bones the model is rigged to (eg from BIP

25 I have seen units where both bones, the original and the transplanted had the same name

to Bip01). Now save and open the nif a second time. Copy the mesh from one of the nifs to the other, make it a child of the primary scene root, it´s the one with the armature, and delete the entire second scene root branch. I know this could be also done without open the nif a second time, but this way it´s saver.

The second problem is a vanilla unit which you can´t import in Blender. Try to import the maceman and there will be an error.[26] The solution is a bit tricky. First select the mesh and export the mesh in nifscope as .obj, select the parent node of the mesh and import the exported mesh. The import misses important informations, like the NiSkinInstance, so the model is no longer rigged. But because it´s the same model you can copy the NiSkinInstance and also the NiTexturingProperty from the not exported mesh. After this the model should look correct again. As last steps change the name of imported mesh to the one of the original mesh and delete the second. Save and now the file should be importable again, as long no other part of the nif needs exact the same threatment.

26 At least with the newer scripts

III. Advanced methods

With the knowledge of chapter one and two you can now do more than only exchange meshes in nif. More or less with bone transplant and rescaling you can deform a unit until you like the result, you can adjust the position of vertices, the uv layout and much more. A bit of the more will be part of the this chapter.

III.1. Changing animations

Before writing down what is possible, perhaps i should say that if the interpolator in the following parts is a NiBSplineCompTransformInterpolator, i´m not sure if it works the way like described. So you must test it.

In chapter one the structure of an animation file was explained. Now we want to really use this knowledge to extend our possibilities in using existing animations. The first easy trick is to change the position of a node, think you modelled a ship but you turret layout is not covered by the existing animations. Now you could make a new animation on your own or as one possible solution, change the position of the turret nodes in the nif until you turrets have the correct position. Write down the numbers and save the file. Now play the idle animation and you will see the turrets are again on their old places.

So open the idle kf and search the NiTransformInterpolator which controls the turrets. Change the position in the NiTransformInterpolator and if there exists a NiTransformData with translation keys also there. If there is a real animation – means the value are not constant over the time - try to mimic the old differences with your start value as base. After saving the file and again checking the nif with idle animation all seems correct. Almost, you now only have to change the position in all kf files with data for the turret nodes.

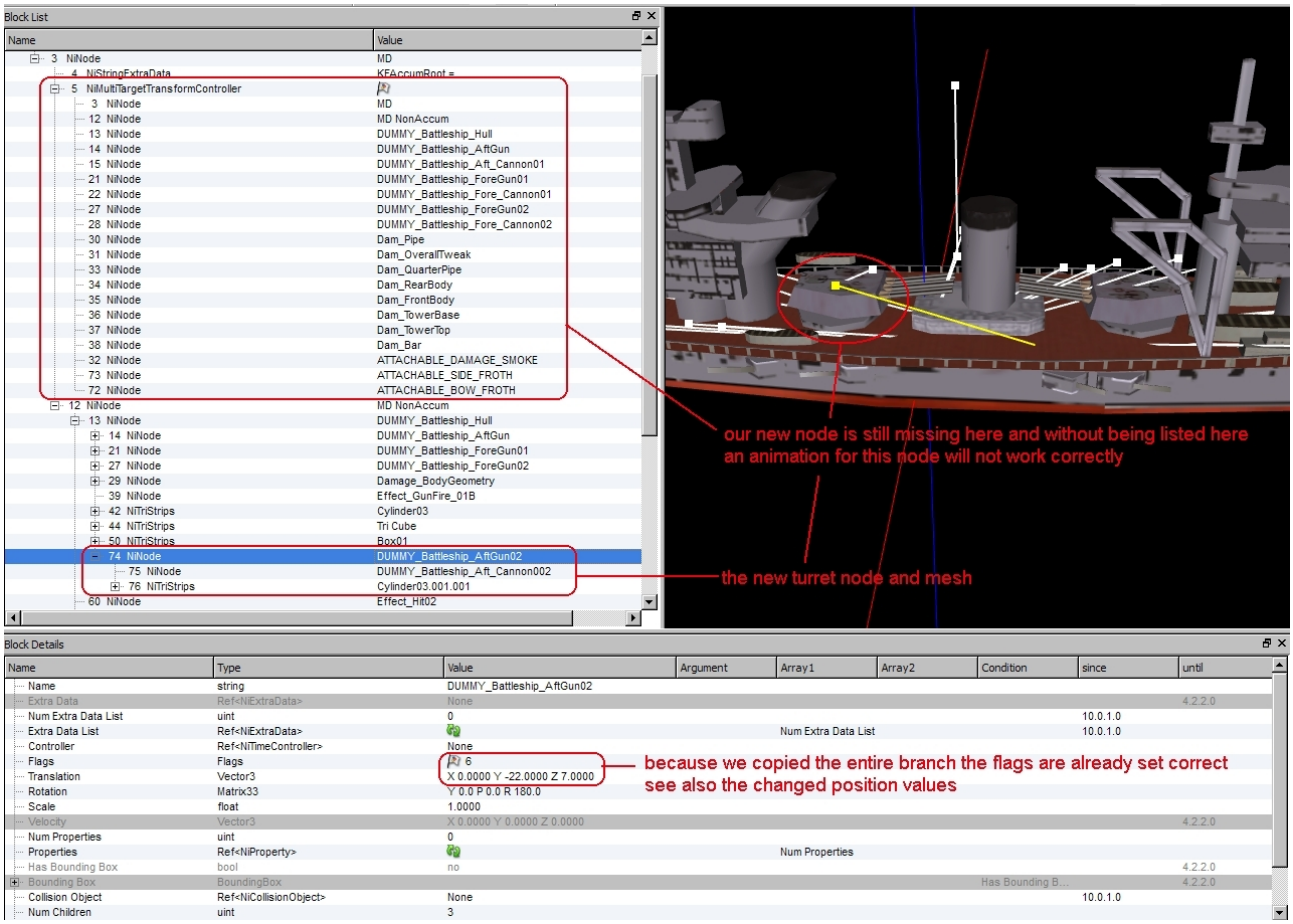
No we want something more interesting, we want to add a new animated turret bone to our model. Download General Matt´s Oktyabrskaya Revolutsiya (OR)[27] and copy all battleship kf files, together with the nif and the kfm, in this folder. Now open the Oktyabrskaya_Revolutsiya_BB.nif in nifscope. You will see that the structure could need some rework to make our future steps easier. Remove the unneeded nodes in front of the scene root and remove the animations from the kf. For this compare the original BB MD node with the MD from the OR, you will see the differences. After this save the file and check the strike animation. You will see that the two middle turrets are not working. Our target is to fix this.

But first look at the model itself, you see that the model has more than one parts, each turret also the hull has its own mesh. Also the model parts are direct assigned as children to the armature nodes. The reason is that the model is not rigged to an animation – you will see that there is no NiSkinInstance. The model works because each children of an animated node is also affected from the animation. Now one turret[28] can be fixed using the steps above, make the mesh a child of the turret without mesh and change the position of the turret in the nif and the kf files. But for the fourth turret there is now node left, so we need another solution.

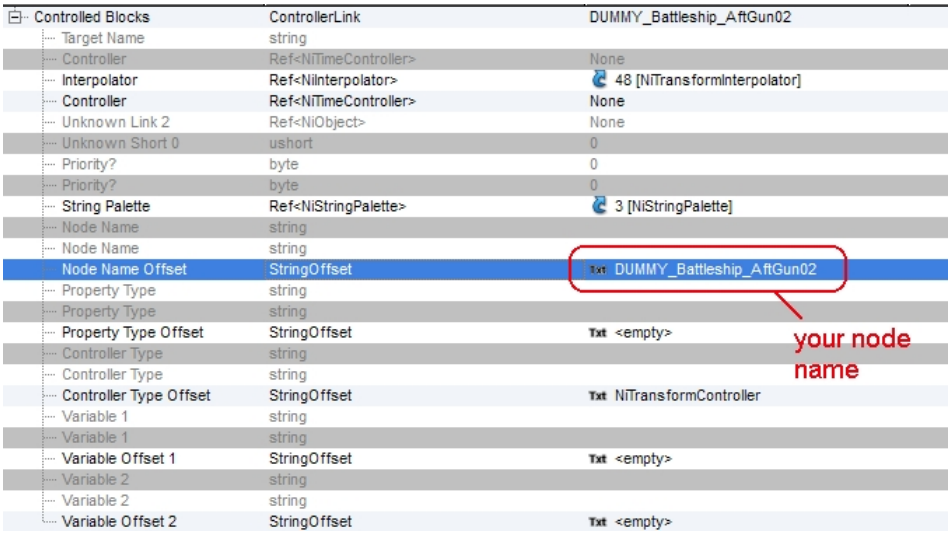
We need a new animated bone. Make a branch copy of the aft turret node and make this copy a child of the DUMMY_Battleship_Hull node. Now rename the copied nodes to differ them from the original aft turret, eg add the suffix 2, and after this change the position of the node. Use for this the

27 http://forums.civfanatics.com/downloads.php?do=file&id=8648
28 cylinder03

values of the turret we want to get animated. Because we copied the entire branch there is already a turret mesh a child of the new turret node, reset the position of this mesh and remove the now unneeded one from the file. Now save the file.



The result so far should look similar to the pic above, also the next task is mentioned there. We must add our new node to the NiMultiTransformController under extra targets.[29] Now theoretical our turret can be animated, but there are no informations for this node in the kf files so far. So lets open as first the strikeA and add the informations there.



29 Only the nodes, not the mesh

At first search for the NiTransformInterpolator of the aft turret and add a branch copy in the kf of this interpolator. After this change the values in this copy to the numbers of our new turret. Now we must add the information that the game should use this interpolator for our new turret. Select the NiControllerSequence and add a new controlled block by raising the number by one. Now under controlled blocks the will be an empty block. Enter all needed informations until the results looks like shown above. Save the file and lets test the new strike animation in nifscope.

If all was done correct now all turrets should rotate. The next problem is, we only have three firing effects in the nif so we must also add a new effect for the fourth turret. Make a copy of one of the nodes and change the position of the copy until its fits the firing position of the fourth turret. And because your are doing this you can also alter the position of the third effect to match the third tower. Before again open the strikeA rename the effect node for the fourth turret, so that the name is unique. In the strikeA go directly to the text keys.

Value	string	start eventcode=1022
Forward	string	
Backward	string	
TBC	TBC	X 0.0000 Y 0.0000 Z 0.0000
Text Keys	Key<string>	
Time	float	0.3000
Value	string	Effect:Effect_GunFire_01A:EFFECT_WEAPON_BATTLESHIP01
Forward	string	Effect:Effect_GunFire_02B:EFFECT_WEAPON_BATTLESHIP02
Backward	string	Effect:Effect_GunFire_03A:EFFECT_WEAPON_BATTLESHIP01
TBC	TBC	Effect:Effect_GunFire_04A:EFFECT_WEAPON_BATTLESHIP02
Text Keys	Key<string>	
Time	float	0.3333
Value	string	SOUND:AS3D_UN_DESTROYER_FIRE,AS3D_UN_SIX_SHOT_FIRE
Forward	string	
Backward	string	
TBC	TBC	X 0.0000 Y 0.0000 Z 0.0000
Text Keys	Key<string>	
Time	float	0.3333
Value	string	Effect:Effect_GunFire_01B:EFFECT_WEAPON_BATTLESHIP02 Effect:Effect_GunFire_02A:EFFEC
Forward	string	
Backward	string	
TBC	TBC	X 0.0000 Y 0.0000 Z 0.0000
Text Keys	Key<string>	
Time	float	0.7000
Value	string	end
Forward	string	
Backward	string	
TBC	TBC	X 0.0000 Y 0.0000 Z 0.0000

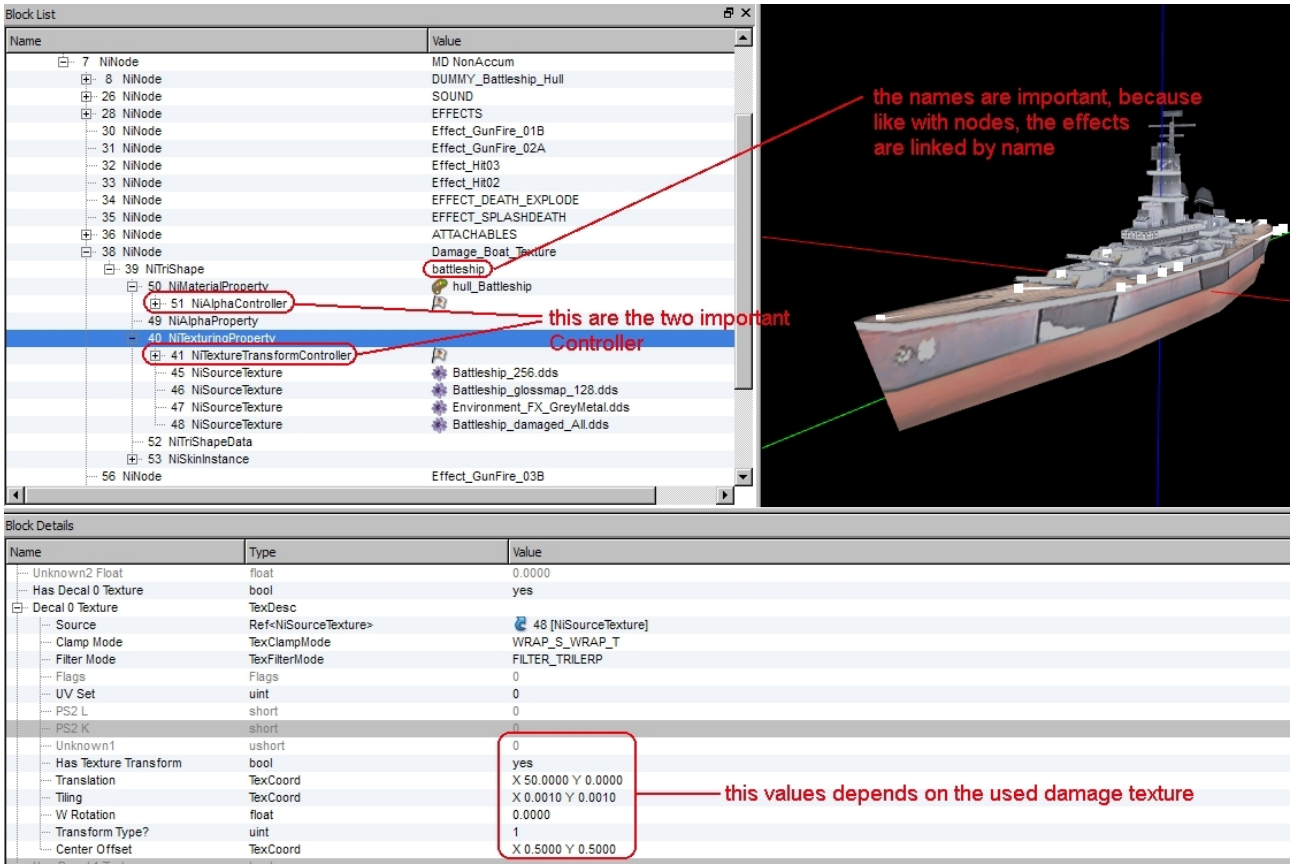
I marked the added entry, a similar entry should be added to the text key for time 0.3333. Now the ship has four firing turrets. Now you only have to had the new turret to all other kf files like described above.

And now you perhaps will hate me, while it´s absolut necessary to add animation information for new added bones, it´s not necessary to change the position in all kf files. There is also a quicker, but a bit dirtier, way to change the position of an animated node. The reason is that like said before every bone has only a relativ position to the parent node and this is what we use. We add a new node between the parent and the node. RMB on the parent node and Node -> Attach Node -> NiNode, set the flag of this new node to six, rename if you want and make the node you want to move a child of this new node. Now change the position of the link node until your node has the right position and rotation. I think one example of this technik are the centaurs in FFH, the upper part of the rider was made a child of the front spine of the horse.

III.1. Damage texture

The last part of the tutorial will cover how to add a damage texture to a rigged mesh. The damage texture parts of the animation have endings like damtexture1. This files manipulated certain controller attached to the mesh in the nif. To show you the important ones open the battleship_fx.nif

in nifscope.



It´s best to copy the needed values and nodes from a unit which already uses the damage texture. Than you must add the kf files to your animation if there aren´t part so far. It´s also possible to add a damage texture to more than one mesh in the nif, open the kf files you want to use and make a copy of all controllers in this kfs and link this controller to the mesh you want.

IV. Appendix

Software:

Blender:	http://www.blender.org/
Gimp:	http://www.gimp.org/
Paint.net:	http://www.getpaint.net/index.html
NifScripts:	http://sourceforge.net/project/showfiles.php?group_id=149157&package_id=166219
NifScope:	http://sourceforge.net/project/showfiles.php?group_id=149157&package_id=170735
Niflib:	http://sourceforge.net/project/showfiles.php?group_id=149157&package_id=166113
PyFFI:	http://sourceforge.net/project/platformdownload.php?group_id=199269
Python:	http://www.python.org/

Tutorials:

Export from Blender: http://forums.civfanatics.com/showthread.php?t=167335
Texturing: http://www.colacola.se/howto_texttut.htm
Assign vertices to bones: http://forums.civfanatics.com/showthread.php?t=252568
some basics and assign vertices: http://forums.civfanatics.com/showthread.php?t=267233
Unit making: http://forums.civfanatics.com/showthread.php?t=279548

Nifviewer, Nifscope: http://forums.civfanatics.com/showthread.php?t=263814
UV edit in nifscope: http://forums.civfanatics.com/showthread.php?t=258966
Nifscope: http://forums.civfanatics.com/showthread.php?t=183742
Nifviewer, Part I: http://forums.civfanatics.com/showthread.php?t=163585
Nifviewer, Part II: http://forums.civfanatics.com/showthread.php?t=165689
Nifviewer, Part III: http://forums.civfanatics.com/showthread.php?t=176106
Nifviewer, shader, teamcolour: http://forums.civfanatics.com/showthread.php?t=245363

Animations: http://forums.civfanatics.com/showthread.php?t=271331
Blender animation: http://wiki.blender.org/index.php/Manual/PartI/Your_First_Animation_in_30_plus_30_Minutes_Part_II
Blender animations, text keys: http://niftools.sourceforge.net/wiki/Blender/Oblivion_Character_Animation
Blender, animation: http://forums.civfanatics.com/showthread.php?t=296286