Reference-Guide
NifUtilsSuite

Skyfox

December 1, 2013
Version 1.2.0

Contents

List of Figures

List of Tables

1 Preface

1.1 About NifUtilsSuite

Welcome and thank you for using to NifUtilsSuite.

NifUtilsSuite is a collection of tools about manipulating files in NetImmerse File Format (NIF). Especially they work with Skyrim version of NIF (20.2.0.7).

These tools are some 'little helper' for:

- changing NIF files of older formats (e.g. 4.0.0.2, 10.0.1.0) into 20.2.0.7
- adding Skyrim collision type to models (*bhkCompressedMeshShape*)
- extracting Chunk collision model information from *bhkCompressedMeshShape*
- preparing Skyrim armor for import into Blender

In addition, another small tool showing a rendered NIF using DirectX is implemented. It could visualize collision wireframes, too - even chunks from *bhkCompressedMeshShape*.

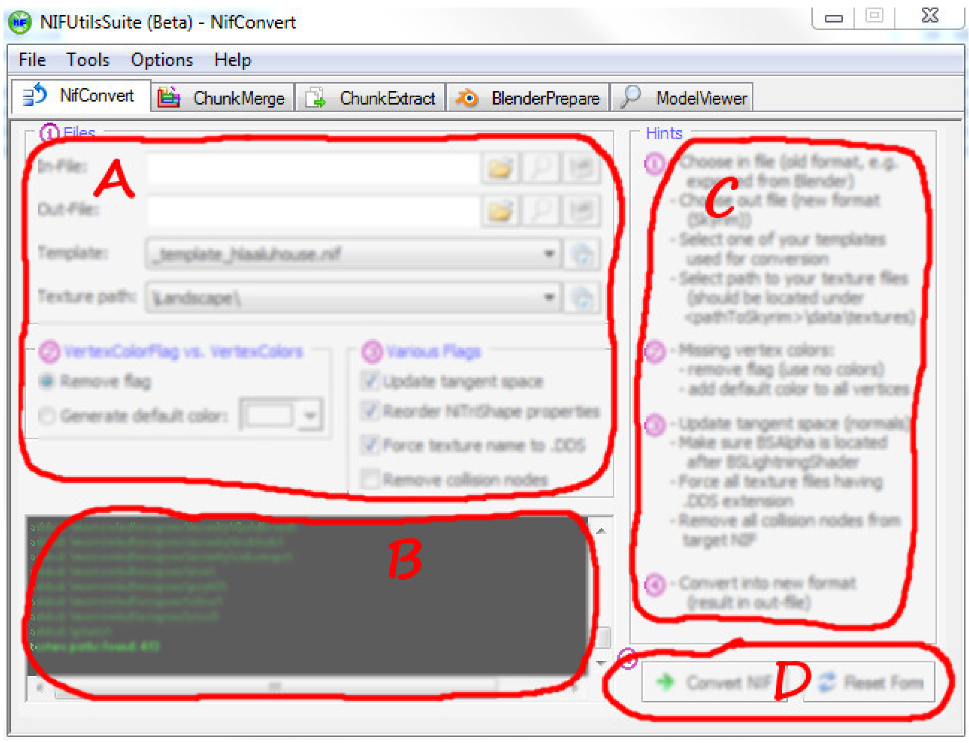
1.2 Thanks

I would like to they thanks to some special people helping me making NifUtilsSuite possible. These are (in no special order):

- **neomonkeus** – Never ending motivation and a lot of support. And sometimes the needed kick in the pants ☺
- **ttl269** – Inspirations, ideas and support for decoding that impossible *bhkCompressedMeshShape*
- **Tamira** – Inspirations, Blender/Fallout3 support and many hours of testing
- **jonwd7** – Bug reports and additions to NifSkope
- **theru** – Inspirations
- **Malo** – Inspiration to write BlenderPrepare
- and all the others at NifTools (`http://www.niftools.org`).

1.3 Common Layout

All tools mentioned in the section before are merged into one application framework: NifUtilsSuite.



Files

In-File:

Out-File:

Template: _template_Haaluhouse.nif

Texture path: Landscape\

VertexColorFlag vs. VertexColors

- [✓ Remove flag]
- Generate default color:

Various Flags

- [✓] Update tangent space
- [✓] Reorder NiTriShape properties
- [✓] Force texture name to .DDS
- [ ] Remove collision nodes

texture paths found: 4[illegible]

Hints

1. - Choose in file (old format, e.g. exported from Blender)
   - Choose out file (new format (Skyrim))
   - Select one of your templates used for conversion
   - Select path to your texture files (should be located under <pathToSkyrim>\data\textures)
2. - Missing vertex colors:
   - remove flag (use no colors)
   - add default color to all vertices
3. - Update tangent space (normals)
   - Make sure BSAlpha is located after BSLightningShader
   - Force all texture files having .DDS extension
   - Remove all collision nodes from target NIF
4. - Convert into new format (result in out-file)

Figure 1-1: Common layout of tool pages

NifUtilsSuite has one tab-page per tool. Except of ModelViewer each page share a similar layout as shown in Figure 1-1. A page is divided into four sections:

A. **Individual input data and flags for the tool**
   These fields are described at the tool chapters under the section *Form Fields*

B. **A small text field containing some log information**
   All log output of the corresponding tool is written here. These section has same functionality on each page.

C. **Some hints about what to do and in which order**
   This section contains some hints how to use the selected tool.

D. **Two buttons performing the tools action or resetting the form**
   And Action!

1.4 Common Icons

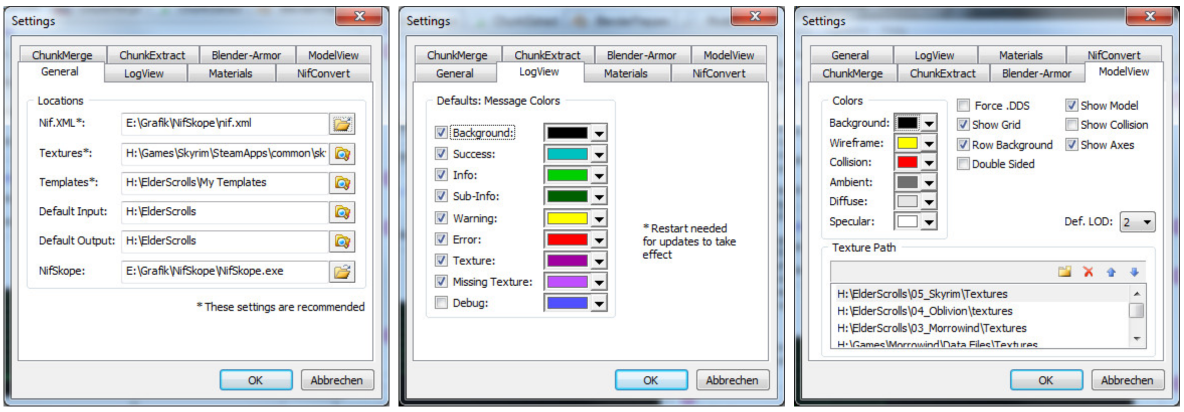
Next to a common layout for all tools there are some recurring icons within NifUtilsSuite used on more than one page. Their look and meaning are shown in Figure 1-2.

📂	**File-Dialog**	Open file dialog and choose a filename
📁	**Folder-Dialog**	Open folder dialog and choose a directory
⚙	**Open Settings**	Open the settings dialog (Chapter Settings)
🔍	**Open ModelViewer**	Switch to ModelViewer and display NIF
🖥	**Open NifSkope**	Open external app NifSkope and display NIF

Figure 1-2: Common used icons

1.5 First Startup

NitUtilsSuite detects a first time startup automatically. It will display the settings dialog as in Figure 1-3.



Locations
Nif.XML*: E:\Grafik\NifSkope\nif.xml
Textures*: H:\Games\Skyrim\SteamApps\common\sk
Templates*: H:\ElderScrolls\My Templates
Default Input: H:\ElderScrolls
Default Output: H:\ElderScrolls
NifSkope: E:\Grafik\NifSkope\NifSkope.exe
* These settings are recommended

Defaults: Message Colors
✓ Background: ⬛
✓ Success: 🟦
✓ Info: 🟩
✓ Sub-Info: 🟩
✓ Warning: 🟨
✓ Error: 🟥
✓ Texture: 🟪
✓ Missing Texture: 🟪
☐ Debug: 🟦
* Restart needed for updates to take effect

Colors
Background: ⬛
Wireframe: 🟨
Collision: 🟥
Ambient: ⬛
Diffuse: ⬜
Specular: ⬜
☐ Force .DDS
✓ Show Grid
✓ Row Background
☐ Double Sided
✓ Show Model
☐ Show Collision
✓ Show Axes
Def. LOD: 2
Texture Path
H:\ElderScrolls\05_Skyrim\Textures
H:\ElderScrolls\04_Oblivion\textures
H:\ElderScrolls\03_Morrowind\Textures
H:\Games\Morrowind\Data Files\Textures

Figure 1-3: Settings dialog

This is the first time setup. In order to getting NifUtilsSuite to full work, you have to fill out the forms. All fields marked with * are required at least. More on settings dialog is described in the corresponding Chapter Settings.

1.6 About Templates

Some words about template NIFs which are used on many places in NifUtilsSuite. Though their main purpose is to create a valid NIF tree usable for Skyrim (version 20.2.0.7), each tool takes other parts of them for input.

- **NifConvert:** NifConvert uses the template as main tree. This means the model to be converted is not done so within its own tree but each node is moved to the new tree provided by the template. In other words: Each node of the 'old style' tree is cut off of it and added to the 'new style' tree provided by the template. By this the main tree structure of *BSFadeNode*, *BSXFlags* and *NiTriShapes* is guaranteed. Also *BSLightingShaderProperty* and *NiAlphaProperty* of the first *NiTriShape* found within the template are used as templates in case of converting material or alpha properties.

- **ChunkMerge:** In case of ChunkMerge the template is used in a slightly other way. As ChunkMerge handles collision information, the *bhkCollisionObject* of the templates root *BSFadeNode* is used as template for all created collision data. This includes the whole sub-tree of *bhkCollisionObject*. That is *bhkRigidBody*, *bhkMoppBvT*reeShape, *bhkCompressedMeshShape* and *bhkCompressedMeshShape-Data*.
- **BlenderPrepare:** When preparing an exported armor NIF from Blender for use in Skyrim, Blender-Prepare needs a template, too. The first *BSLightingShaderProperty* is used when creating the desired properties of *NiTriShape*s.

Therefore, it makes sense to have different templates for each task. Even having different templates for NifConvert could make sense. One for converting static NIFs like houses and another for more or less static trees. It is up to you.

2 NifConvert

NifConvert is a tool which transforms a NIF of 'old' style into a Skyrim version.

2.1 Overview

NifConvert transforms an 'old style' NIF into one that's nearly readable by Skyrim. Following versions are supported:

- in: 4.0.0.2 (e.g. Morrowind)
- in: 10.0.1.0 (e.g. Oblivion)
- out: 20.2.0.7 (e.g. Skyrim)

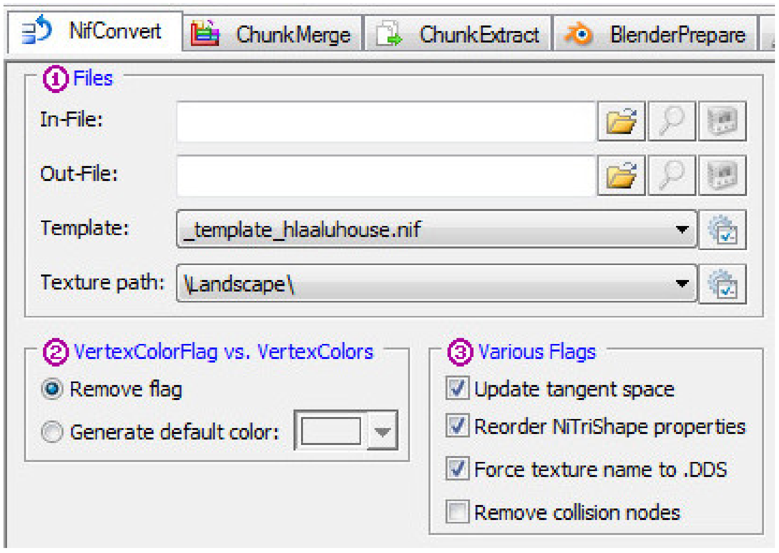
After converting through NifSkope the NIFs are mostly readable by Construction Kit (CK) or Skyrim. Some specials modification have to be added by using other tools e.g. NifSkope.

After loading the complete NIF as tree NifSkope parses the whole tree and analyses every node. All nodes are 'shiftet' to the new version which means, that known and compatible nodes like shapes, properties and collision data are repositioned to match Skyrim needs. All other nodes are removed from the tree.

Before removing nodes not supported by Skyrim some of them, in especially properties, are analysed and converted into their newer pendants. As an example, the *NiTexturingProperty* property node is converted into a *BSLightingShaderProperty* property and the texture name(s) are copied to the new node. Or, as another example, tangents and bitangents from a *NiBinaryExtraData* property are copied into their correspondending fields of *NiGeometryData* node.

2.2 Form Fields

The fields section of NifConvert page presents itself as shown in Figure 2-1.



In-File:

Out-File:

Template: _template_hlaaluhouse.nif

Texture path: \Landscape\

[✓ Remove flag]

Generate default color:

[✓ Update tangent space]

[✓ Reorder NiTriShape properties]

[✓ Force texture name to .DDS]

Remove collision nodes

Figure 2-1: NifConvert – Form fields

The fields and their meaning are described in the following table:

Files		
▭	**In-File**	Name of input file which should be converted for Skyrim.
▭	**Out-File**	Name of output file which is stored in Skyrim format.
▾	**Template**	Name of NIF-file used as template while conversion. Base tree structure and nodes are taken from template while model and collision nodes are copied from input file.
▾	**Texture path**	This path is used base base directory for all textures converted. Each texture file name is cut to be relative to this directory.
VertexColorFlag vs. VertexColors		
◉	**Remove flag**	Sometimes NIF shape nodes have vertex colors but the related flag isn't set or vice versa. This choice removes the flag in case of non-existing vertex colors.
◉	**Gen. def color**	This choice adds vertex colors to a NIF shape node having the corresponding flag set. All vertices get the same color as defined by 'default color'.
▭▾	**Default color**	Defines the color used when creating vertex colors. This field will be enabled only in case of »Generate default color« is selected.
Various Flags		
☑	**Upd. tang. space**	(Re-)generate tangents and bitangents in model shape nodes.
☑	**Reord. Ni. prop.**	Reorder properties of *NiTriShape* node to match correct order (e.g. *NiAlphaProperty*).
☑	**Force text. DDS**	Remove endings of all texture file names and append .DDS as file ending (e.g. `mytexture.tga` would become `mytexture.dds`).
☑	**Rem. coll. nodes**	Remove all collision nodes/sub-trees (*RootCollisionNode* and *bhkCollisionObject*) from resulting NIF before saving.

Table 2.1: NifConvert – Field descriptions

2.3 Supported Node-Types

All node types supported by NifConvert are listed below.

- **BSFadeNode**
- **NiNode**
- **NiTriShape**
- **NiTriStrips**
- **NiTriStripsData**
- **NiExtraData**
- **NiBinaryExtraData**
- **RootCollisionNode**
- **NiAlphaProperty**
- **bhkPackedNiTriStripsShape**
- **NiVertexColorProperty**
- **BSLightingShaderProperty**
- **NiTexturingProperty**
- **NiMaterialProperty**

3 ChunkMerge

ChunkMerge is a tool to add collision information to NIF files having version 20.2.0.7 (Skyrim).

3.1 Overview

With ChunkMerge you could add new style collision data to existing NIF files of version 20.2.0.7. This could be done by

- creating completely new collision model from mesh/shape model
- converting 'old-style' collision data into new one
- re-using existing collision data if compatible

In case of creating new or converting collision data the new type would be in Skyrims *bhkCompressedMesh-Shape* style. Re-using could only be done on collision node types supported by Skyrim. Unfortunately not all former types are supported in Skyrim any more.

Next to creating/converting collision data, ChunkMerge could set the type of material, too. By doing this collision data of a wooden stair acts like wood (arrows could penetrate) and a stone wall like stone (arrows bounce off). You could choose between one material for all collision data or a single material per model shape.

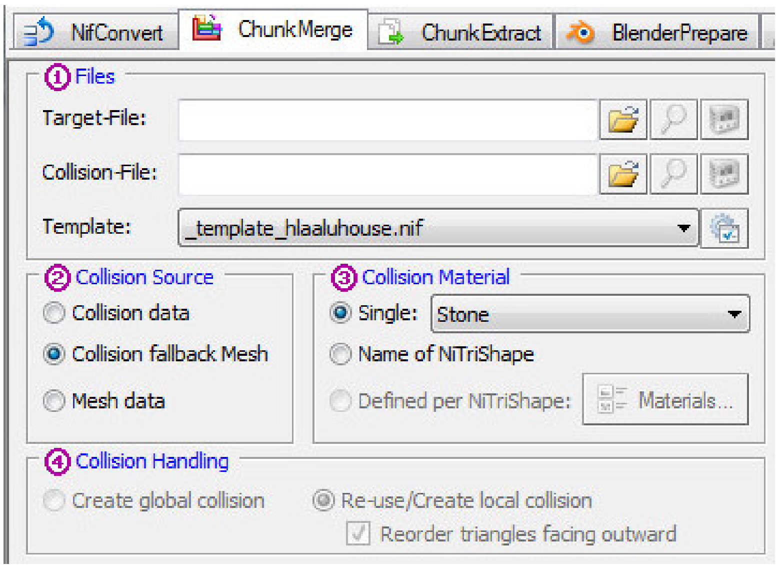
The latter is a bit tricky because there is no way to assign materials to shape in ChunkMerge yet. This is planned but not implemented yet. But there is a workaround for this by using the name of *NiTriShape*. Entering material definition text as defined in `nif.xml` as name for a shape triggers ChunkMerge to decode this name to a material and assign this to the corresponding collision model. As the number of definitions may grow, these are read from `nif.xml` at startup of NifUtilsSuite. Here are some examples:

- **SKY_HAV_MAT_WOOD:** Material Wood
- **SKY_HAV_MAT_SNOW:** Material Snow
- **SKY_HAV_MAT_MATERIAL_STONE_AS_STAIRS:** Material Stone As Stairs
- **SKY_HAV_MAT_GLASS:** Material Glass

To get all benefits of this feature, you have to point NifUtilsSuite to the file `nif.xml` at the first startup. This could be changed in the settings dialog at any time.

3.2 Form Fields

The fields section of ChunkMerge page presents itself as shown in Figure 3-1.


NifConvert | ChunkMerge | ChunkExtract | BlenderPrepare

① Files

Target-File:

Collision-File:

Template: _template_hlaaluhouse.nif

② Collision Source

○ Collision data
[✓ ◉ Collision fallback Mesh]
○ Mesh data

③ Collision Material

[✓ ◉ Single: Stone]
○ Name of NiTriShape
○ Defined per NiTriShape: Materials...

④ Collision Handling

○ Create global collision
[✓ ◉ Re-use/Create local collision]
[✓] Reorder triangles facing outward

Figure 3-1: ChunkMerge – Form fields

The fields and their meaning are described in the following table:

Files

▭	**Target-File**	Name of output file to which collision data should be added to.
▭	**Collision-File**	Name of the file containing the collision data to be added to »Target-File«. There is a special handling in case of »Collision-File« is same as »Target-File«. See »Collision Handling« for more information on this.
▾	**Template**	Name of NIF-file used as template while conversion. Collision nodes like *bhkCollisionObject*, *bhkMoppBvTreeShape* or *bhkCompressedMeshShape* are adapted from this template file and added to the output NIF.

Collision Source

◉	**Collision data**	Collision data would be generated from collision data information of »Collision-File«. If no collision data is stored within this file, no collision data is written to »Target-File«.
◉	**Coll. fb. Mesh**	Collision data would be generated from collision data information of »Collision-File«. If no collision data is stored within this file, collision data is generated from mesh data of »Collision-File«. This produces very 'big' collision information mostly, as mesh data usually has much more polygons than collision data.
◉	**Mesh data**	Collision data would be generated from mesh data of »Collision-File« with the disadvantage of 'big' collision information size.

Collision Material

●	**Single**	All collision objects (e.g. chunks in *bhkCompressedMeshShapeData*) get the same material as defined in the nearby selection box. There will be one single material for all newly created collision objects.
▾	**Material**	The material all collision objects get in case of choosing »Single« as material source.
●	**Name of NiTri.**	The material is defined by the naming of matching NiTriShape nodes. This works only in case of choosing »Mesh data« as »Collision Source«. The name of the NiTriShape node has to match the material name definition in `nif.xml` (e.g. SKY_HAV_MAT_STONE).
●	**Def. per NiTri**	This option is not available yet.
Collision Handling		
●	**Cr. glob. coll.**	This is the default option and produces a single collision node (*bhkCollisionObject*) for the whole mesh model. This is located as sub-tree directly in the root node.
●	**ReUse loc. coll.**	In case of »Collision-File« is the same as »Target-File« a 'search-and-replace' of collision data could be processed. This is, parsing the whole NIF tree for collision nodes and modify them matching Skyrims needs (e.g. scaling). In some cases (e.h. *bhkNiTriStrips*) modifying has no effect because the collision nodes are not supported by Skyrim anymore. Such collision data is converted into *bhkCompressedMeshShape*.
✓	**Reord. triangles**	When handling TriStrips every odd triangle has wrong winding so collision data would be generated in a wrong way. Reordering the triangles winding produces correct collision data.

Table 3.1: ChunkMerge – Field descriptions

3.3 Supported Node-Types

All collision type nodes supported by ChunkMerge are listed below.

- **BSFadeNode**
- **RootCollisionNode**
- **NiNode**
- **NiTriShape**
- **NiTriStrips**
- **bhkCollisionObject**
- **bhkRigidBody**
- **bhkMoppBvTreeShape**
- **bhkCompressedMeshShape**
- **bhkCompressedMeshShapeData**
- **bhkCMSDChunk**
- **bhkCMSDMaterial**
- **bhkPackedNiTriStripsShape**
- **hkPackedNiTriStripsData**
- **bhkTransformShape**
- **bhkListShape**
- **bhkBoxShape**
- **bhkCapsuleShape**
- **bhkSphereShape**
- **bhkConvexVerticesShape**

4 ChunkExtract

ChunkExtract exports collision information included in *bhkCMSDChunk*s as *NiTriShape*s in a NIF file.

4.1 Overview

Sometimes it is only necessary to make slight adjustments to an existing collision model. The fact, that this model is extremly compressed in *bhkCMSDChunk* nodes of *bhkCompressedMeshShapeData* nodes for NIF version 20.2.0.7 (e.g. Skyrim), makes this work nearly impossible.

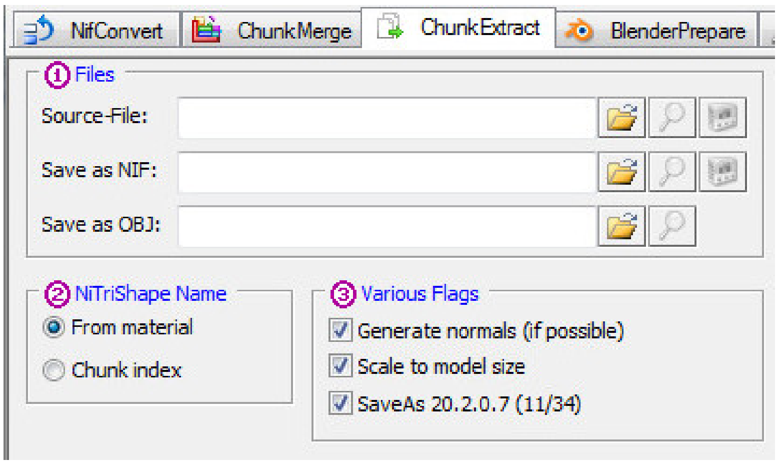
ChunkExtract helps in this case. This tool extracts the collision data stored in *bhkCompressedMeshShape-Data* as simple *NiTriShape*s and saves them as NIF tree. Saving the tree as version 20.2.0.7 with user version 11/34 (FallOut3 format) the NIF could be imported in Blender without another manual modification.

After being modified and exported from Blender, the resulting NIF file could be used as »Collision-File« in ChunkMerge. A simple round-trip-modification is born.

Besides the possibility of saving as NIF the collision information could be saved using the OBJ-format, too. But remember, this is not as precise as saving in NIF-format.

4.2 Form Fields

The fields section of ChunkExtract page presents itself as shown in Figure 4-1.



[✓ From material] Chunk index

[✓] Generate normals (if possible)
[✓] Scale to model size
[✓] SaveAs 20.2.0.7 (11/34)

Figure 4-1: ChunkExtract – Form fields

The fields and their meaning are described in the following table:

Files

	Source-File	Name of input file the collision should be extracted from.
	Save as NIF	Name of NIF-file exported NiTriShapes should be written to. If this field is empty, no NIF data would be saved.

▭	**Save as OBJ**	Name of OBJ-file exported geometries should be written to. If this field is empty, no OBJ data would be saved.
NiTriShape Name		
◉	**From material**	NiTriShape nodes are named by the related material definition (e.g. SKY_HAV_MAT_STONE).
◉	**Chunk index**	NiTriShape nodes are named by index number of related *bhkCMSDChunk* (e.g. Chunk #1).
Various Flags		
☑	**Gen. normals**	Try generating face normals for exported *NiTriShape*s/gemometries. This might not always be possible.
☑	**Scale to model**	Collision geometry in *bhkCompressedMeshShapeData* is scaled by factor 1/1000. Choosing this option will scale the collision geometry to match the model size, so multiply by 1000.
☑	**Save as (11/34)**	Export the NIF with version 20.2.0.7 with user version 11/34. This version could be imported as FallOut3 NIF in Blender directly.

Table 4.1: ChunkExtract – Field descriptions

4.3 Supported Node-Types

All collision type nodes supported by ChunkExtract are listed below.

- **bhkCollisionObject**
- **bhkRigidBody**
- **bhkMoppBvTreeShape**
- **bhkCompressedMeshShape**
- **bhkCompressedMeshShapeData**
- **bhkCMSDChunk**

5 BlenderPrepare

BlenderPrepare could be used to modify an armor NIF file with version 20.2.0.7 in a way to be importable in Blender and vice versa.

5.1 Overview

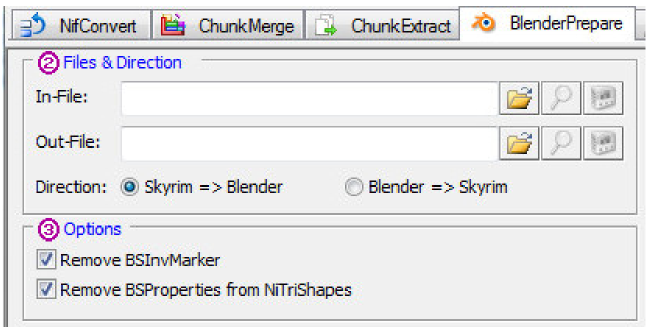
Skyrim armor NIFs could not be imported into Blender directly as of yet unsupported format.

To make things easier on one side BlenderPrepare cuts down Skyrim version armor NIFs in such a way to nearly match Oblivion/FallOut3 which could be imported into Blender. In detail it removes all *BSSIn-vMarker* objects and all *BSProperties* attached to a *NiTriShape*.

On the other side BlenderPrepare could be used to get armor exported from Blender nearly workable in Skyrim. This is done by exchanging body part definitions as given within a mapping table. In addition a *BSLightingShaderProperty* is added to each NiTriShape, too.

5.2 Form Fields

The fields section of BlenderPrepare page presents itself as shown in Figures 5-1 and 5-2.



In-File:

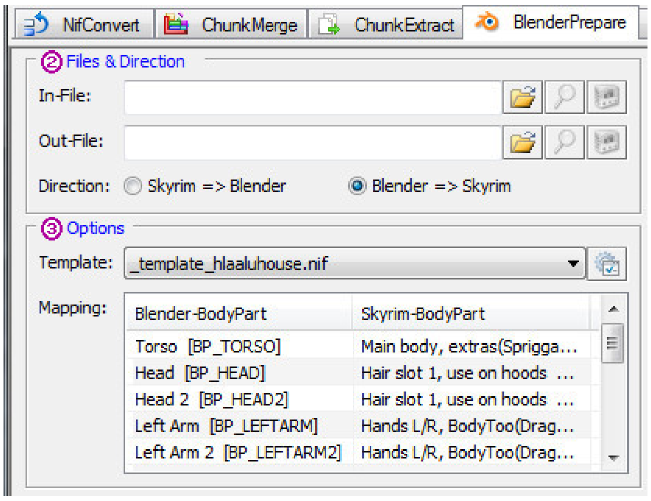
Out-File:

Direction: [✓ Skyrim => Blender] Blender => Skyrim

[✓] Remove BSInvMarker

[✓] Remove BSProperties from NiTriShapes

Figure 5-1: BlenderPrepare – Form fields 'to Blender'



In-File:

Out-File:

Direction: Skyrim => Blender [✓ Blender => Skyrim]

Template: _template_hlaaluhouse.nif

Mapping:

Blender-BodyPart	Skyrim-BodyPart
Torso [BP_TORSO]	Main body, extras(Sprigga...
Head [BP_HEAD]	Hair slot 1, use on hoods ...
Head 2 [BP_HEAD2]	Hair slot 1, use on hoods ...
Left Arm [BP_LEFTARM]	Hands L/R, BodyToo(Drag...
Left Arm 2 [BP_LEFTARM2]	Hands L/R, BodyToo(Drag...

Figure 5-2: BlenderPrepare – Form fields 'from Blender'

The fields and their meaning are described in the following table:

Files		
▭	**In-File**	Name of input file which armor should be prepared for Blender.
▭	**Out-File**	Name of output file with armor prepared for Blender.
Options - To Blender		
☑	**Rem. BSInvMarker**	Remove all *BSInvMarker* from NIF tree before saving.
☑	**Rem. BSproperties**	Remove all *BSProperty*s and derived objects from *NiTriShapes* of NIF tree before saving.
Options - From Blender		
▾	**Template**	Name of NIF-file used as template while conversion from Blender to NIF version 20.2.0.7.
▦	**Mapping**	Table showing the mappings between Blender exported body parts (mostly Oblivion or FallOut3 types) and Skyrim body parts. These could be changed at the BlenderPrepare page of Settings dialog (Chapter BlenderPrepare).

Table 5.1: BlenderPrepare – Field descriptions

5.3 Supported Node-Types

All collision type nodes supported by BlenderPrepare are listed below.

- **NiNode**
- **NiTriShape**
- **NiAVObject**
- **NiExtraData**
- **BSInvMarker**
- **BSLightingShaderProperty**
- **BSDismemberSkinInstance**
- **BSShaderPPLightingProperty**

6 ModelViewer

ModelViewer is a simple visualization tool for displaying NIF models including their collision data.

6.1 Overview

Originally ModelViewer was written to display collision data stored in *bhkCMSDChunks* of *bhkCompressedMeshShapeData* nodes. No other tool was able to do so - not even NifSkope - at that time. Thanks to jonwd7 one of the upcomming versions of NifSkope can display this collision data, too, I think.

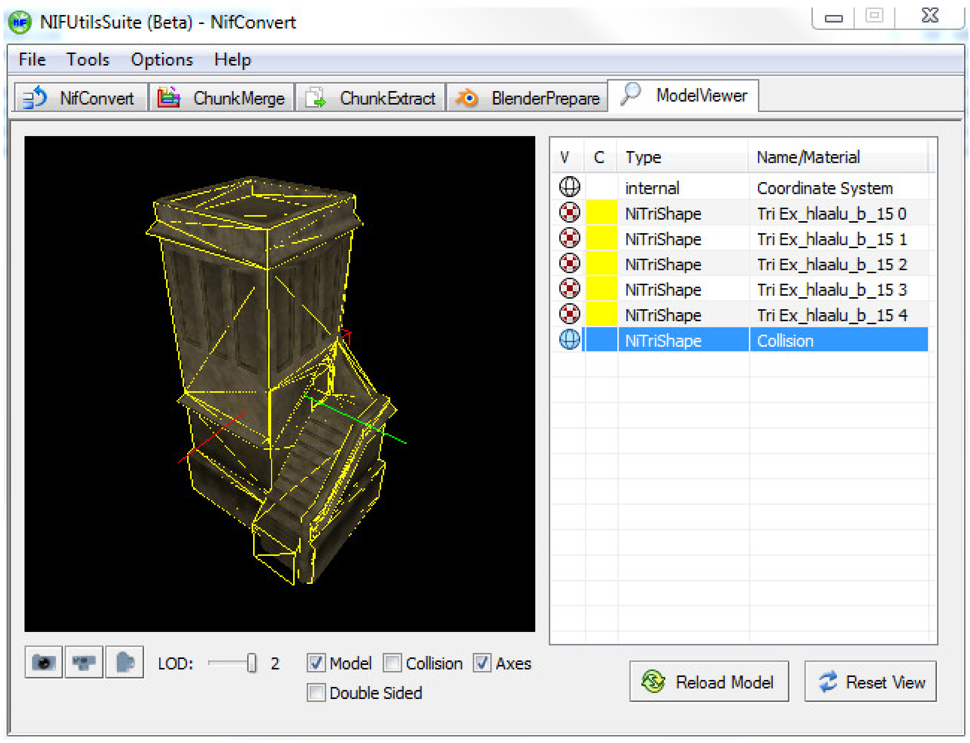
But what is displaying Skyrim like collision data when not showing the model itself? So ModelViewer grew in functionality, first showing the model itself, than displaying more and more collision data from other nodes e.g. *bhkBoxShape* or *bhkPackedNiTriShape*.

Some special display settings could be done using right mouse button on the object list table on the right side. This includes switching between wireframe and solid rendering or changing wireframe color per node. More about this could be found in Table 6.2.

6.2 Form Fields

The layout of this page differs from other tools layout. One central part of this page builds the rendering window found on the left side. A list of all known and found nodes of a NIF is listed on the table to the right. The control fields are located under these sections.

The ModelViewer page presents itself as shown in Figures 6-1.



V	C	Type	Name/Material
⊕		internal	Coordinate System
⊗	🟨	NiTriShape	Tri Ex_hlaalu_b_15 0
⊗	🟨	NiTriShape	Tri Ex_hlaalu_b_15 1
⊗	🟨	NiTriShape	Tri Ex_hlaalu_b_15 2
⊗	🟨	NiTriShape	Tri Ex_hlaalu_b_15 3
⊗	🟨	NiTriShape	Tri Ex_hlaalu_b_15 4
⊕		NiTriShape	Collision

LOD: 2 [✓] Model [ ] Collision [✓] Axes [ ] Double Sided

Figure 6-1: ModelViewer – Form fields

The fields and their meaning are described in the following table:

Widgets

	Front View	Reset display to view from front.
	Top View	Reset display to view from top.
	Side View	Reset display to view from side.
	LOD	Some NIF models have a 'Level-Of-Detail' depending from the in game view distance. There are three known levels at current time. This slider could be used to adjust this level within ModelViewer.
☑	**Model**	Switch displaying model shapes on/off.
☑	**Collision**	Switch displaying collision shapes on/off.
☑	**Axes**	Switch displaying coordinate system on/off.
☑	**Double Sided**	Switch displaying shapes in double side mode on/off.

Table 6.1: ModelViewer – Field descriptions

The node listing table on the right side has some extended functionality:

Object Table

	Full Render	Render object in full rendering mode including textures.
●	**Solid**	Render object in solid mode.
⊕	**Wireframe**	Render object in wireframe mode only show edges as lines.
	Invisible	Do not render object at all.
🖱	**Right Click**	Right mouse click on table opens its context menu.

Table 6.2: ModelViewer – Object table functions

Graphics display could be controlled by:

Render Window

🖱	**Left Click**	Mouse movement rotates displayed model.
🖱	**Right Click**	Mouse movement zoomes displayed model.
🖱	**Middle Click**	Mouse movement moves center of displayed model.
🖱	**Wheel**	Mouse movement zoomes displayed model.

Table 6.3: ModelViewer – Render window functions

6.3 Supported Node-Types

All collision type nodes supported by ModelViewer are listed below.

- **NiNode**
- **BSFadeNode**
- **NiBillboardNode**
- **NiTriShape**
- **NiTriStrips**
- **BSLODTriShape**
- **NiTriBasedGeom**
- **NiTriShapeData**
- **NiTriStripsData**
- **NiAlphaProperty**
- **NiTexturingProperty**
- **NiMaterialProperty**
- **NiSourceTexture**
- **BSLightingShaderProperty**
- **BSShaderTextureSet**
- **RootCollisionNode**
- **bhkCollisionObject**
- **bhkCompressedMeshShape**
- **bhkPackedNiTriStripsShape**
- **bhkNiTriStripsShape**
- **bhkConvexVerticesShape**
- **bhkBoxShape**
- **bhkSphereShape**
- **bhkCapsuleShape**
- **bhkRigidBodyT**
- **bhkMoppBvTreeShape**
- **bhkCompressedMeshShapeData**
- **hkPackedNiTriStripsData**
- **bhkConvexTransformShape**
- **bhkListShape**
- **bhkTransformShape**

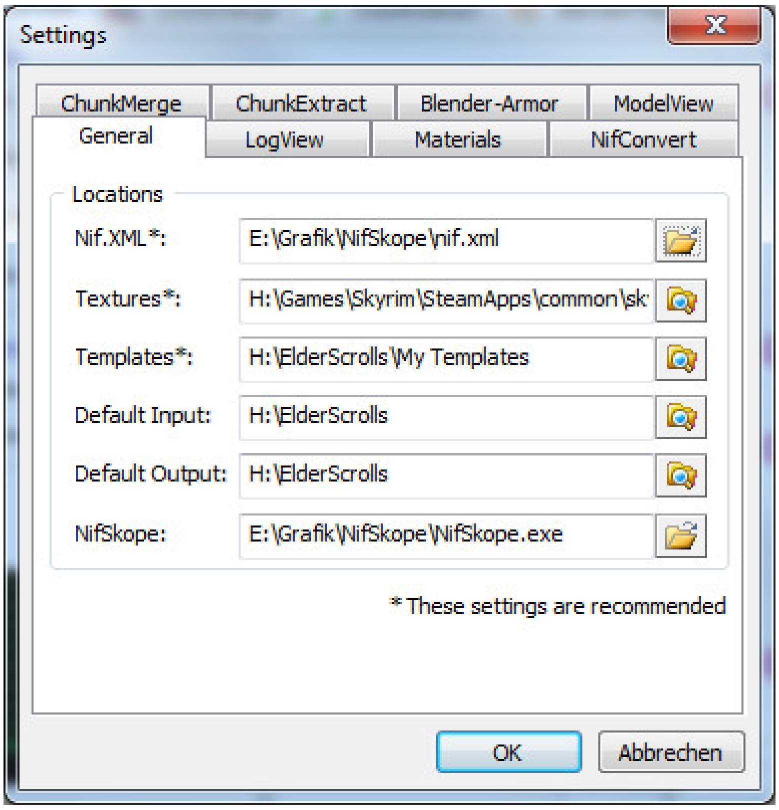
7 Settings

This chapter is about the Settings Dialog.

All settings done here are used as defaults only. Most of them could be overwritten within the related tool.

7.1 General

Settings common for all tools included in NifUtilsSuite are set here - paths to files at least.



Locations

Nif.XML*: E:\Grafik\NifSkope\nif.xml

Textures*: H:\Games\Skyrim\SteamApps\common\sk

Templates*: H:\ElderScrolls\My Templates

Default Input: H:\ElderScrolls

Default Output: H:\ElderScrolls

NifSkope: E:\Grafik\NifSkope\NifSkope.exe

* These settings are recommended

Figure 7-1: Settings – General

Locations

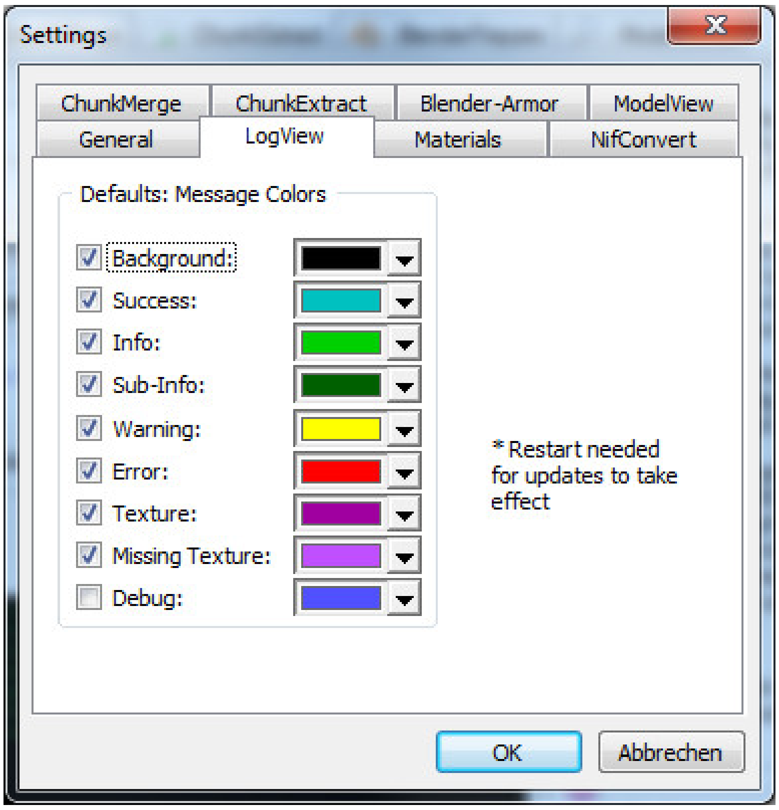
Nif.XML	Path to actual `nif.xml` file. The file is used for identification of materials, layers and body parts. This setting is mandatory if NifUtislSuite should work correct.
Textures	Path to root directory of texture files. It is used as base directory for texture file names in NifConvert to create relative paths. It is mandatory, too.
Templates	Path to a directory holding some (at least one) NIF file of version 20.2.0.7 user version 12/83 (Skyrim). This file is used as template for many operations to create a correct NIF tree. This field is mandatory, too.

▭	**Def. Input**	Path to default input directory where all your NIF files are stored you want to modify with NifUtilsSuite.
▭	**Def. Output**	Path to default output directory. All modified files would be saved here.
▭	**NifSkope**	Path to NifSkope executeable. If set, all models could be opened in NifSkope directly by pressing a simple button (1-2)on the related tools page.

Table 7.1: Settings – General page

7.2 LogView

The level of logging information and colors for different log levels are defined here.



Defaults: Message Colors

✓	Background:	⬛
✓	Success:	🟦
✓	Info:	🟩
✓	Sub-Info:	🟩
✓	Warning:	🟨
✓	Error:	🟥
✓	Texture:	🟪
✓	Missing Texture:	🟪
	Debug:	🟦

* Restart needed for updates to take effect

Figure 7-2: Settings – LogView

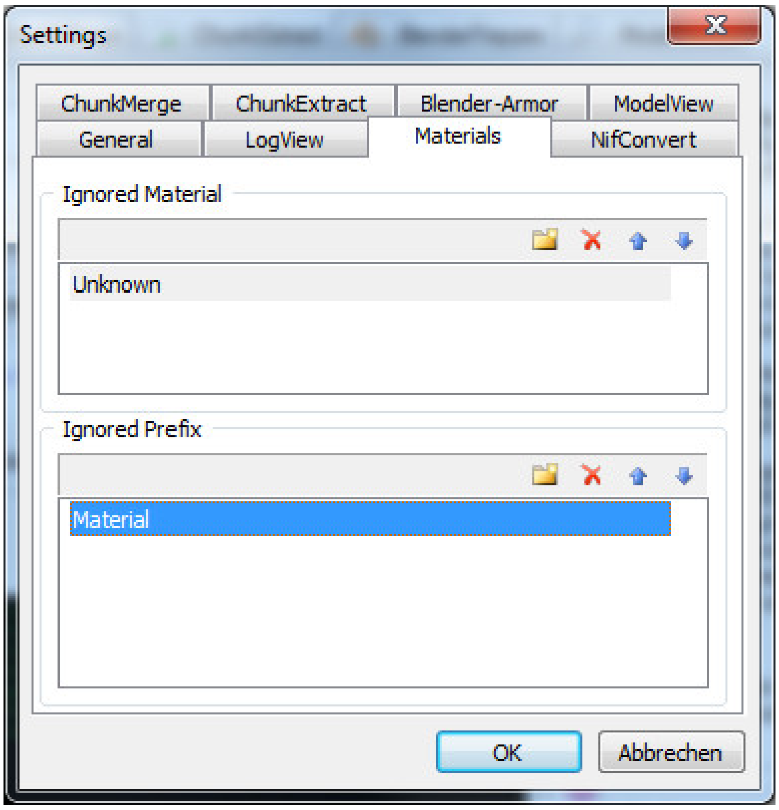
Message Colors

☑	*<diverse>*	Using the checkbox fields enables or disables the corresponding log information.
▭▾	*<diverse>*	Sets the color for corresponding log level.

Table 7.2: Settings – LogView page

7.3 Material

On this page you could define which material should be ignored when reading from `nif.xml`. Prefixes which should not be used for material name display could be set, too.



Ignored Material

Unknown

Ignored Prefix

Material

Figure 7-3: Settings – Material

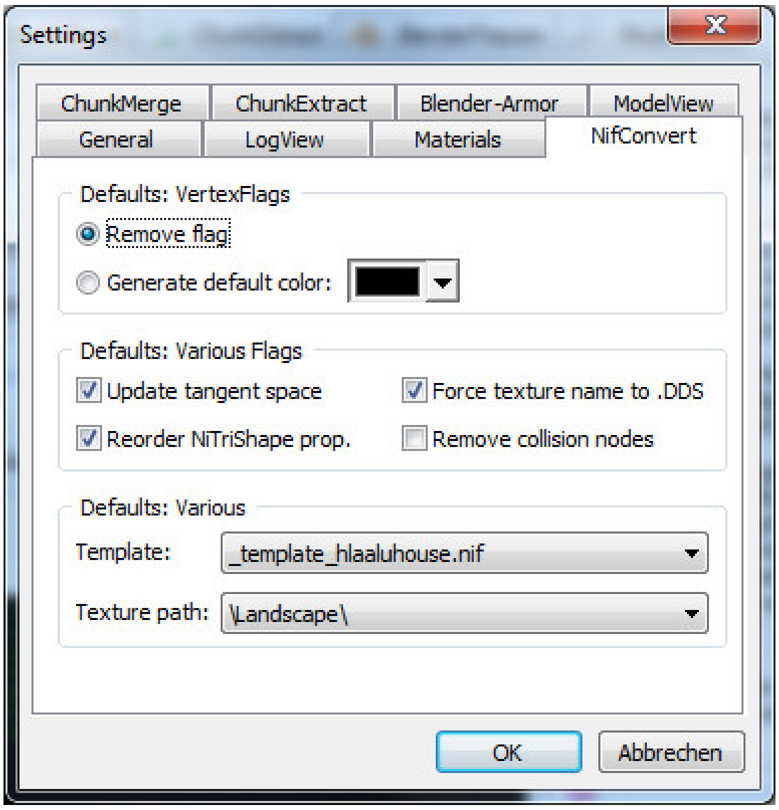
Ignored Material/Prefix

	Ign. Material	A list of strings which a material will not be read from `nif.xml` at startup when its name sounds like the string. For example: An entry of 'Unknown' would lead to ignore matrials named 'Unknown_Water' or 'Some_Unknown'. NifUtislSuite will not know these materials.
	Ign. Prefix	A list of strings which will be cut from material name while reading from `nif.xml` at startup. For example: An entry of 'material' would lead to matrial name of 'Water' from 'MaterialWater'.

Table 7.3: Settings – Material page

7.4 NifConvert

Defaults for NifConvert are set here.



Defaults: VertexFlags

- [✓] Remove flag
- [ ] Generate default color: ■

Defaults: Various Flags

- [✓] Update tangent space
- [✓] Force texture name to .DDS
- [✓] Reorder NiTriShape prop.
- [ ] Remove collision nodes

Defaults: Various

Template: _template_hlaaluhouse.nif

Texture path: \Landscape\

Figure 7-4: Settings – NifConvert

VertexFlags

◉	**Remove flag**	Sometimes NIF shape nodes have vertex colors but the related flag isn't set or vice versa. This choice removes the flag in case of non-existing vertex colors.
◉	**Gen. def color**	This choice adds vertex colors to a NIF shape node having the corresponding flag set. All vertices get the same color as defined by 'default color'.
▭	**Default color**	Defines the color used when creating vertex colors. This field will be enabled only in case of »Generate default color« is selected.

Various Flags

☑	**Upd. tang. space**	(Re-)generate tangents and bitangents in model shape nodes.
☑	**Reord. Ni. prop.**	Reorder properties of *NiTriShape* node to match correct order (e.g. *NiAlphaProperty*).
☑	**Force text. DDS**	Remove endings of all texture file names and append .DDS as file ending (e.g. `mytexture.tga` would become `mytexture.dds`).
☑	**Rem. coll. nodes**	Remove all collision nodes/sub-trees (*RootCollisionNode* and *bhkCollisionObject*) from resulting NIF before saving.

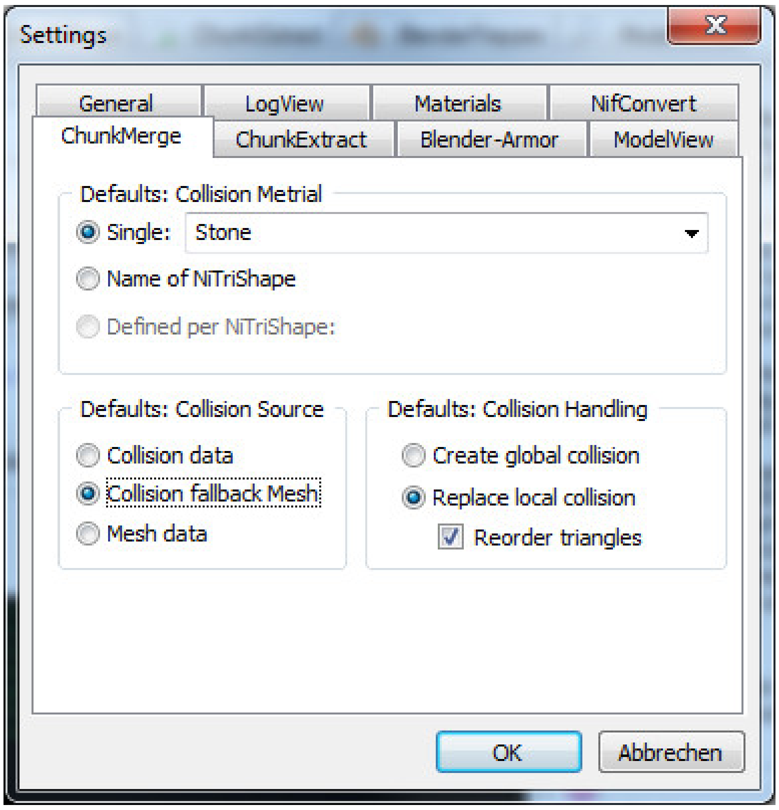
Various

▾	**Template**	Name of NIF-file used as template while conversion. Base tree structure and nodes are taken from template while model and collision nodes are copied from input file.
▾	**Texture path**	This path is used base base directory for all textures converted. Each texture file name is cut to be relative to this directory.

Table 7.4: Settings – NifConvert page

7.5 ChunkMerge

Defaults for ChunkMerge are set here.



Settings

Tabs: General | LogView | Materials | NifConvert | ChunkMerge | ChunkExtract | Blender-Armor | ModelView

Defaults: Collision Metrial
- [✓ Single:] Stone
- Name of NiTriShape
- Defined per NiTriShape:

Defaults: Collision Source
- Collision data
- [✓ Collision fallback Mesh]
- Mesh data

Defaults: Collision Handling
- Create global collision
- [✓ Replace local collision]
- [✓ Reorder triangles]

Figure 7-5: Settings – ChunkMerge

Collision Material

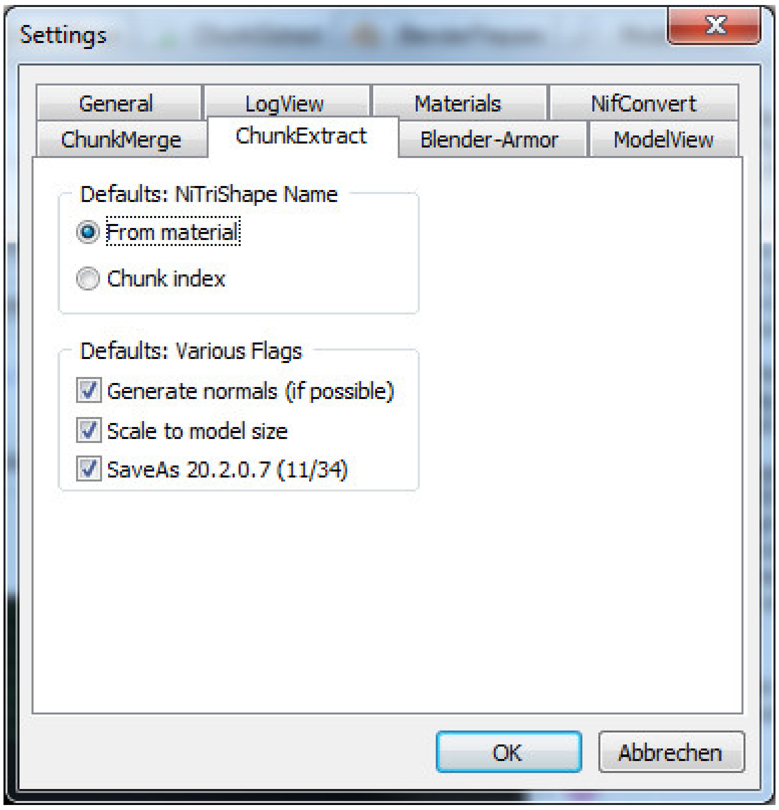
●	**Single**	All collision objects (e.g. chunks in *bhkCompressedMeshShapeData*) get the same material as defined in the nearby selection box. There will be one single material for all newly created collision objects.
▾	**Material**	The material all collision objects get in case of choosing »Single« as material source.

◉	**Name of NiTri.**	The material is defined by the naming of matching NiTriShape nodes. This works only in case of choosing »Mesh data« as »Collision Source«. The name of the NiTriShape node has to match the material name definition in `nif.xml` (e.g. SKY_HAV_MAT_STONE).
◉	**Def. per NiTri**	This option is not available yet.
Collision Source		
◉	**Collision data**	Collision data would be generated from collision data information of »Collision-File«. If no collision data is stored within this file, no collision data is written to »Target-File«.
◉	**Coll. fb. Mesh**	Collision data would be generated from collision data information of »Collision-File«. If no collision data is stored within this file, collision data is generated from mesh data of »Collision-File«. This produces very 'big' collision information mostly, as mesh data usually has much more polygons than collision data.
◉	**Mesh data**	Collision data would be generated from mesh data of »Collision-File« with the disadvantage of 'big' collision information size.
Collision Handling		
◉	**Cr. glob. coll.**	This is the default option and produces a single collision node (*bhkCollisionObject*) for the whole mesh model. This is located as sub-tree directly in the root node.
◉	**ReUse loc. coll.**	In case of »Collision-File« is the same as »Target-File« a 'search-and-replace' of collision data could be processed. This is, parsing the whole NIF tree for collision nodes and modify them matching Skyrims needs (e.g. scaling). In some cases (e.h. *bhkNiTriStrips*) modifying has no effect because the collision nodes are not supported by Skyrim anymore. Such collision data is converted into *bhkCompressedMeshShape*.
☑	**Reord. triangles**	When handling TriStrips every odd triangle has wrong winding so collision data would be generated in a wrong way. Reordering the triangles winding produces correct collision data.

Table 7.5: Settings – ChunkMerge page

7.6 ChunkExtract

Defaults for ChunkExtract are set here.


Settings

Defaults: NiTriShape Name
[✓ From material]
Chunk index

Defaults: Various Flags
[✓] Generate normals (if possible)
[✓] Scale to model size
[✓] SaveAs 20.2.0.7 (11/34)

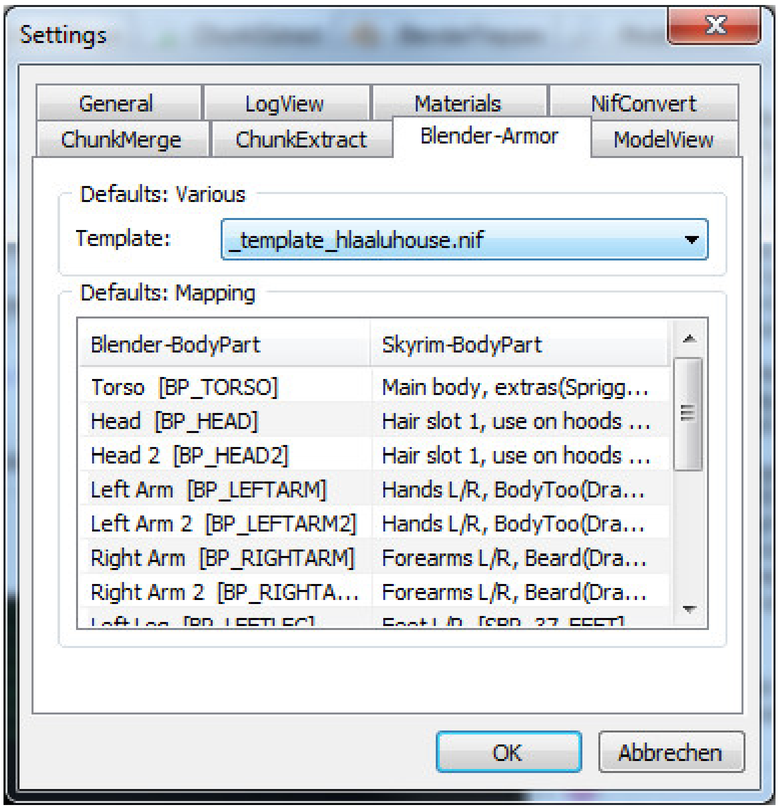
Figure 7-6: Settings – ChunkExtract

NiTriShape Name		
●	**From material**	NiTriShape nodes are named by the related material definition (e.g. SKY_HAV_MAT_STONE).
●	**Chunk index**	NiTriShape nodes are named by index number of related *bhkCMSD-Chunk* (e.g. Chunk #1).
Various Flags		
✓	**Gen. normals**	Try generating face normals for exported *NiTriShapes*/gemometries. This might not always be possible.
✓	**Scale to model**	Collision geometry in *bhkCompressedMeshShapeData* is scaled by factor 1/1000. Choosing this option will scale the collision geometry to match the model size, so multiply by 1000.
✓	**Save as (11/34)**	Export the NIF with version 20.2.0.7 with user version 11/34. This version could be imported as FallOut3 NIF in Blender directly.

Table 7.6: Settings – ChunkExtract page

7.7 BlenderPrepare

Defaults for BlenderPrepare are set here.



Defaults: Various

Template: _template_hlaaluhouse.nif

Defaults: Mapping

Blender-BodyPart	Skyrim-BodyPart
Torso [BP_TORSO]	Main body, extras(Sprigg...
Head [BP_HEAD]	Hair slot 1, use on hoods ...
Head 2 [BP_HEAD2]	Hair slot 1, use on hoods ...
Left Arm [BP_LEFTARM]	Hands L/R, BodyToo(Dra...
Left Arm 2 [BP_LEFTARM2]	Hands L/R, BodyToo(Dra...
Right Arm [BP_RIGHTARM]	Forearms L/R, Beard(Dra...
Right Arm 2 [BP_RIGHTA...	Forearms L/R, Beard(Dra...
Left Leg [BP_LEFTLEG]	Foot L/R [SBP_37_FEET]

Figure 7-7: Settings – BlenderPrepare

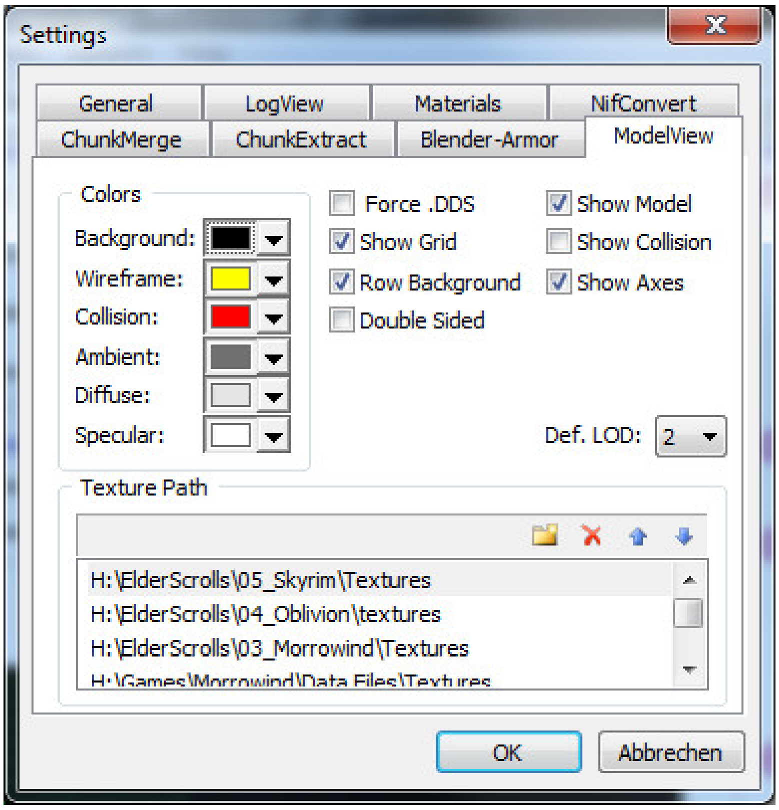
Various

	Template	Name of NIF-file used as template while conversion from Blender to NIF version 20.2.0.7.
	Mapping	Table showing the mappings between Blender exported body parts (mostly Oblivion or FallOut3 types) and Skyrim body parts.

Table 7.7: Settings – BlenderPrepare page

7.8 ModelViewer

Defaults for ModelViewer are set here.



Settings

General | LogView | Materials | NifConvert
ChunkMerge | ChunkExtract | Blender-Armor | ModelView

Colors
- Background: ⬛
- Wireframe: 🟨
- Collision: 🟥
- Ambient: ▩
- Diffuse: ▢
- Specular: ▢

- [ ] Force .DDS
- [x] Show Grid
- [x] Row Background
- [ ] Double Sided
- [x] Show Model
- [ ] Show Collision
- [x] Show Axes

Def. LOD: 2

Texture Path
- H:\ElderScrolls\05_Skyrim\Textures
- H:\ElderScrolls\04_Oblivion\textures
- H:\ElderScrolls\03_Morrowind\Textures
- H:\Games\Morrowind\Data Files\Textures

OK | Abbrechen

Figure 7-8: Settings – ModelViewer

Colors

▢	*<diverse>*	Sets the color for corresponding light source.

Texture Path

▤	**Paths**	Base path for textures. ModelViewer uses this paths as base path when searching for textures. First matching path with existing texture file is used.

Widgets

☑	**Force .DDS**	Force all endings of texture names to .DDS before checking for existing texture file using base path list.
☑	**Show Grid**	Show grid layout in object list table.
☑	**Row Backgnd**	Alternate background of each other row of object list table.
☑	**Model**	Switch displaying model shapes on/off.
☑	**Collision**	Switch displaying collision shapes on/off.
☑	**Axes**	Switch displaying coordinate system on/off.
☑	**Double Sided**	Switch displaying shapes in double side mode on/off.

▾	**Def. LOD**	Some NIF models have a 'Level-Of-Detail' depending from the in game view distance. There are three known levels at current time. This slider could be used to adjust this level within ModelViewer.

Table 7.8: Settings – ModelViewer page