

The Elder Scrolls™ V

SKYRIM



IMPORT, EXPORT ET COLLISIONS AVEC 3DS MAX ET NIFTOOLS

Un tutoriel réalisé par Oaristys pour la *Confrérie des Traducteurs*

SOMMAIRE

Prérequis.....	2
Import	4
Export	5
Conversion pour Skyrim	6
« Nettoyage ».....	8
Ajout d'une branche de collision	9
Astuce – Les vertex colors.....	10

PRÉREQUIS

3ds Max 2012

[Nifskope](#) 1.1.1

Voir le [tutoriel de Gerault](#) pour configurer le logiciel.

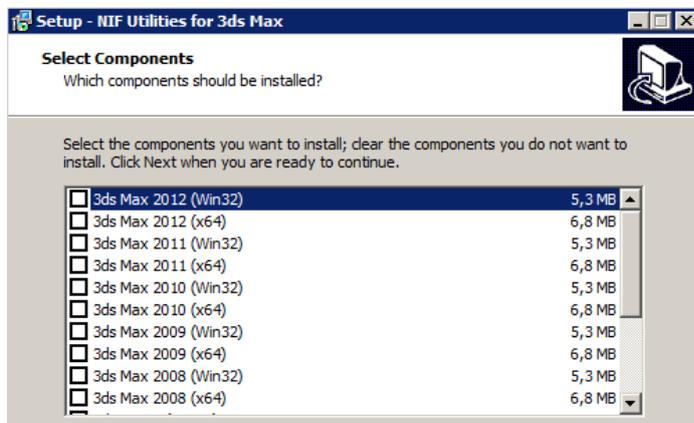
[Niftools](#) 3.7.3 (version non officielle sur Skyrim Nexus)

N.B. : La version 3.4 disponible sur le site officiel n'est pas à jour pour 3ds Max 2012.

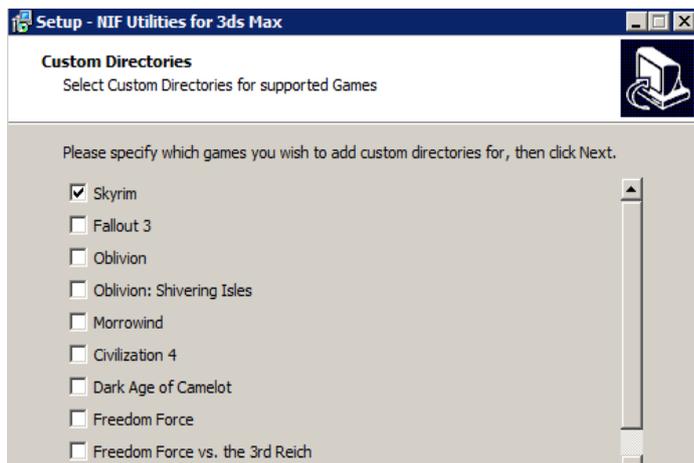
The screenshot shows the Niftools website interface. Under the 'MAIN FILES' section, there are two entries:

- max-plugins-373**: 8,867kb size, 10,605 downloads, 5,607 unique downloads. Below it is the text 'Nif Max Plugins 3.7.3 Installer' and a green button labeled 'DOWNLOAD WITH MANAGER' with the text '... or download manually' to its right. The date '9:55, 27 February 2012' and 'version 3.7.3' are also visible.
- max-plugins-373-zip**: 11,095kb size, 2,661 downloads, 1,636 unique downloads. Below it is the text 'Nif Max Plugins 3.7.3 Archive' and a green button labeled 'DOWNLOAD WITH MANAGER' with the text '... or download manually' to its right. The date '9:57, 27 February 2012' and 'version 3.7.3' are also visible.

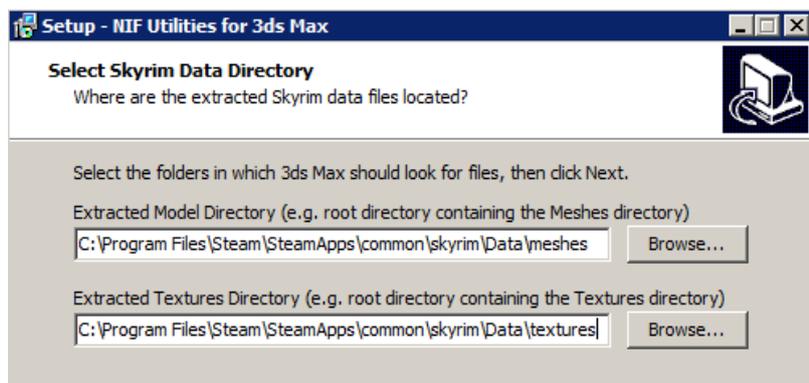
Lancez l'installateur et suivez les étapes. Choisissez votre version de 3ds Max :



Puis choisissez le jeu désiré :



Indiquez le chemin des dossiers meshes et textures de votre jeu (l'endroit où vous avez extrait les fichiers .bsa) :



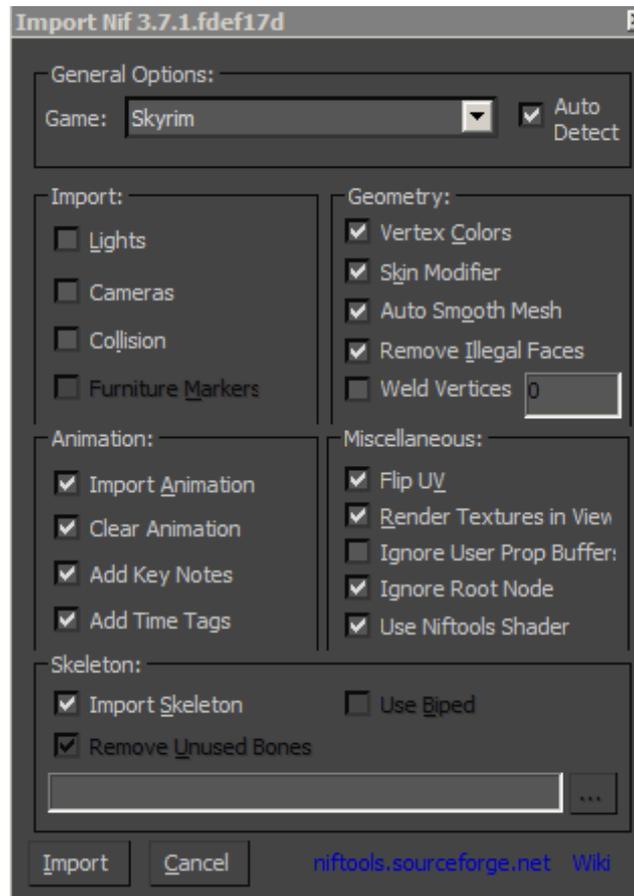
Et cliquez sur *Install*.

[NifUtils](#) pour 3ds Max

Extraire le contenu de l'archive dans le dossier de votre choix (ou sur le bureau pour un accès rapide).

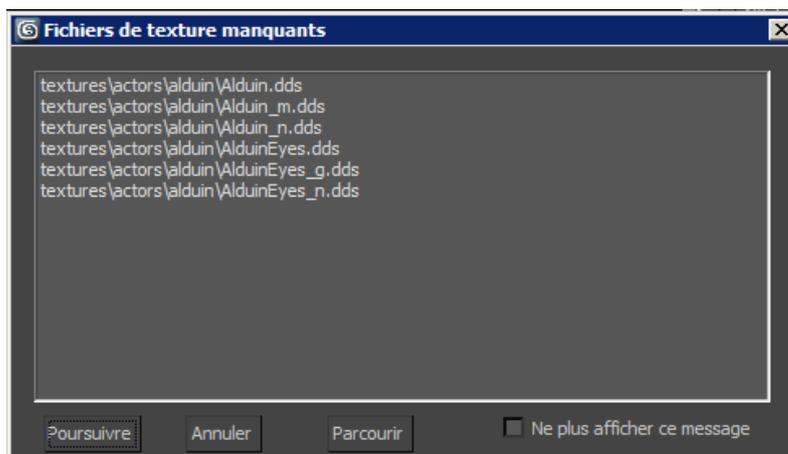
IMPORT

Une fois Niftools installé, si vous souhaitez importer dans 3ds Max des fichiers .nif de Skyrim, utilisez les paramètres d'importation suivants :



Cliquez sur *Import*.

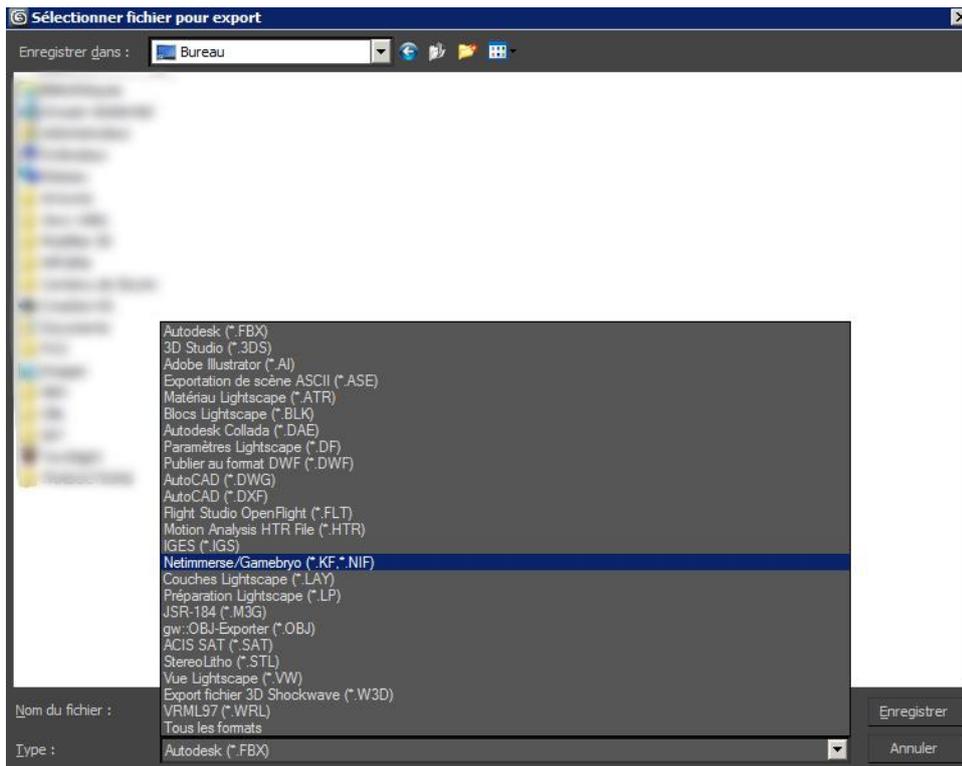
Si le logiciel ne trouve pas les textures au chemin indiqué lors de l'installation de Niftools, cette fenêtre peut s'afficher, indiquant les textures manquantes :



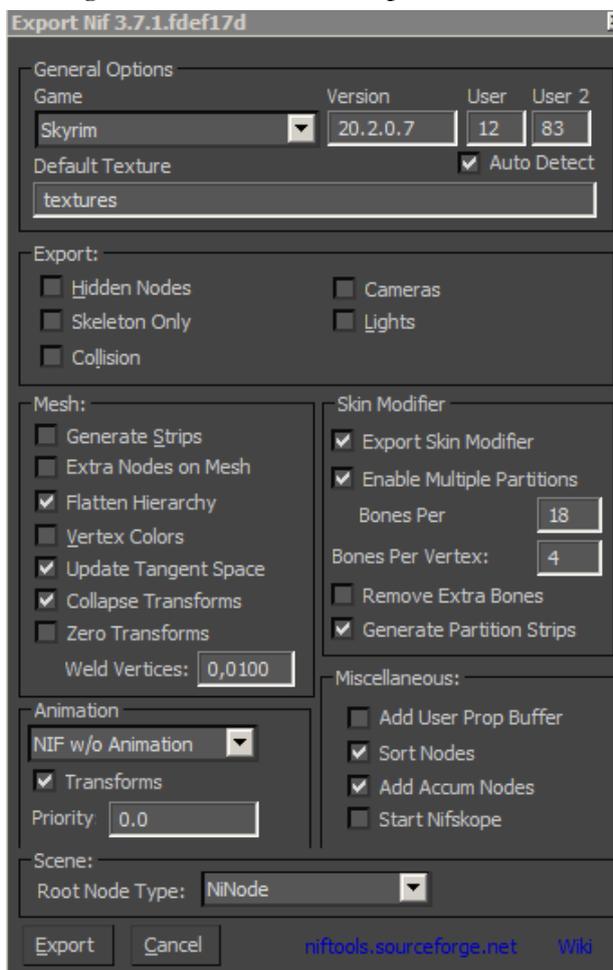
Vous pouvez alors cliquer sur *Parcourir* pour indiquer un autre dossier contenant les textures de votre jeu.

EXPORT

Cliquez sur Exporter, puis sélectionner le format désiré (.NIF) :



Configurez la fenêtre avec les paramètres ci-dessous :



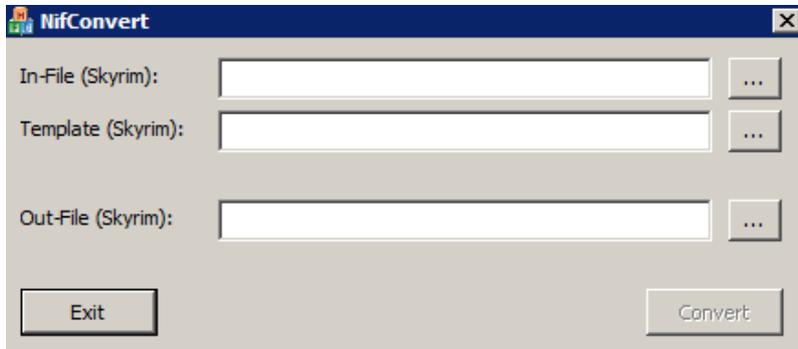
Cliquez sur *Export*.

Vous obtenez un fichier (nommons-le **Mesh.nif**) impropre à toute utilisation dans le Creation Kit. Il faut maintenant le modifier pour le rendre « jouable » grâce à NifUtils.

CONVERSION POUR SKYRIM

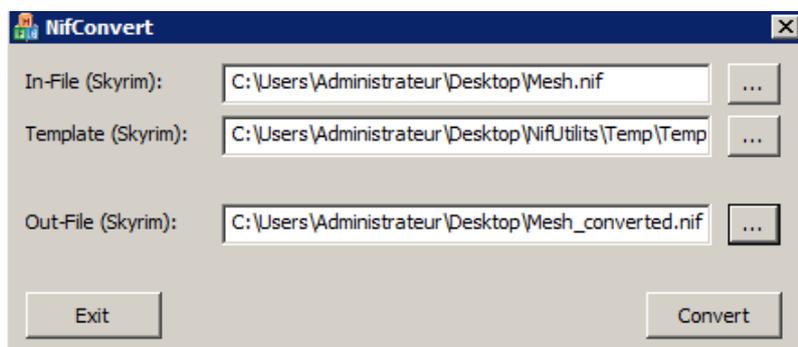
NifUtils comprend deux utilitaires : **NifConvert.exe** et **ChunkMerge.exe**. Dans un premier temps, nous allons utiliser NifConvert pour donner à notre mesh « brut » une forme se rapprochant de celle des meshes de Skyrim.

Lancez **NifConvert.exe**.



- **In-File** : Il s'agit du fichier Mesh.nif préalablement exporté depuis 3ds Max.
- **Template** : Il s'agit du fichier **Temp.nif** contenu dans le dossier NifUtils\Temp. C'est un modèle basé sur le mesh FarmHouse01.nif de Skyrim, que le logiciel va prendre comme référence pour convertir votre mesh.
- **Out-File** : C'est le fichier que vous obtiendrez après conversion.

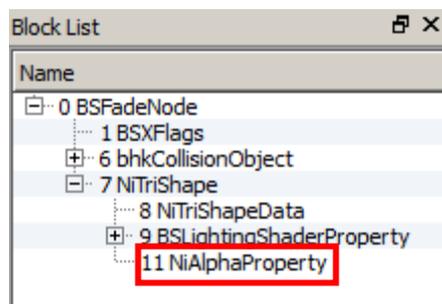
Un exemple de ce qu'on peut obtenir une fois les champs remplis :



Cliquez sur *Convert*. Le fichier Mesh_converted.nif sera créé dans le dossier indiqué (ici le bureau). Il a désormais la forme requise par le CK et Skyrim, mais il n'est pas encore tout à fait prêt à être utilisé.

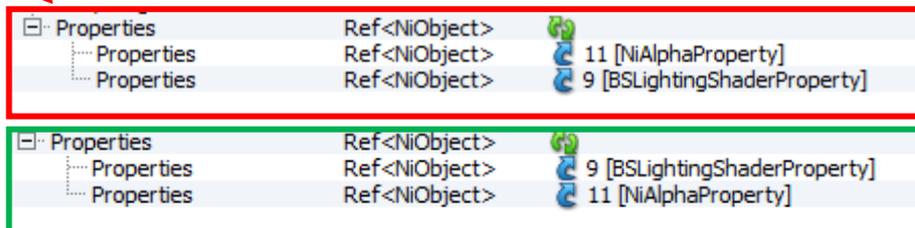
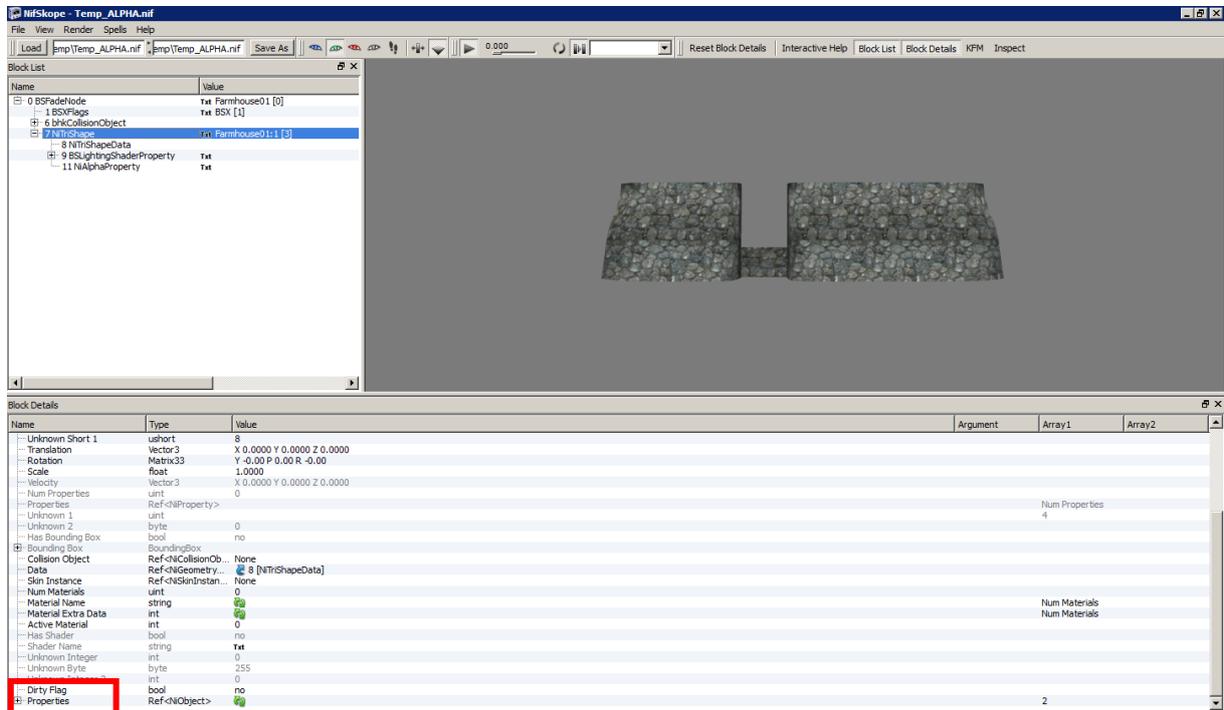
!!! ATTENTION !!!

Si votre Mesh.nif comprend des branches **NiAlphaProperty**, la procédure diffère légèrement. Cette branche, qui sert à gérer la transparence des textures, se présente de cette façon :



1. Assurez-vous, avant d'utiliser NifConvert, que les branches **NiAlphaProperty** de votre Mesh.nif sont bien **EN-DESSOUS** des **BSLightingShaderProperty**, comme sur l'exemple ci-dessus.

Si ce n'est pas le cas (l'ordre peut avoir été altéré par l'export de 3ds Max), sélectionnez votre NiTriShape, allez dans *Propriétés* (dans le *Block Details*) où vous verrez les deux branches ; double-cliquez sur les lignes et retapez les chiffres correspondants en les inversant.



NON

OUI

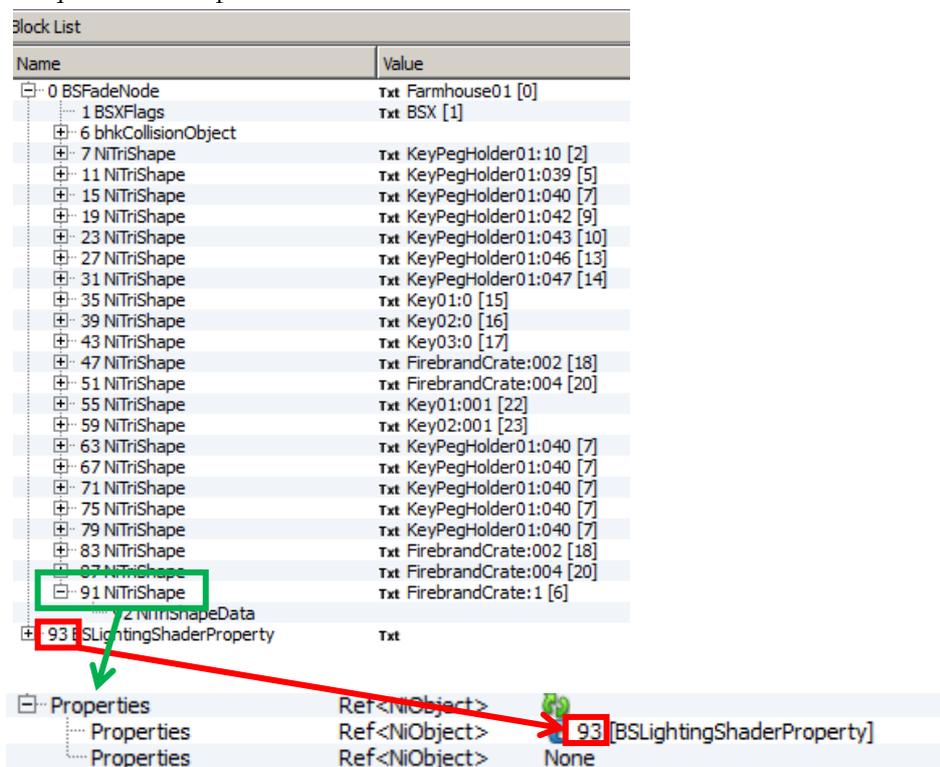
2. Suivez les instructions pour NifConvert, mais au lieu de Temp.nif, utilisez **Temp_ALPHA.nif** qui se trouve dans le même dossier.

« NETTOYAGE »

L'opération décrite ci-dessus avec NifConvert altère la branche BSLightingShaderProperty, car elle applique les données du BSLightingShaderProperty de Temp.nif à notre Mesh.nif. La plupart du temps, ce n'est pas très grave, mais certains objets ont des propriétés spéciales (comme la réflexion) qui donnent l'impression d'un matériau, d'une surface, etc. Un objet très brillant se retrouverait alors aussi mat que notre Temp.nif.

On pourrait avoir la tentation d'utiliser, au lieu de Temp.nif, le mesh vanilla original, pour que notre BSLightingShaderProperty soit directement identique à celui du mesh vanilla, mais c'est impossible (du moins sans préparation préalable du mesh) : le ChunkMerge ne fonctionne pas ensuite. De plus, votre mesh créé dans 3ds Max peut contenir plusieurs meshes vanilla liés entre eux et n'utilisant pas le même BSLightingShaderProperty.

La solution pour récupérer les valeurs des BSLightingShaderProperty vanilla est tout simplement de supprimer la branche dans notre Mesh_converted.nif, puis de copier-coller celle du mesh vanilla (en ouvrant plusieurs fenêtres de Nifskope) et enfin de la relier à notre mesh en inscrivant son numéro dans les *Propriétés* du NiTriShape correspondant. Il faut faire cela pour chaque NiTriShape.



Name	Value
0 BSFadeNode	Txt Farmhouse01 [0]
1 BSXFlags	Txt BSX [1]
6 bhkCollisionObject	
7 NiTriShape	Txt KeyPegHolder01:10 [2]
11 NiTriShape	Txt KeyPegHolder01:039 [5]
15 NiTriShape	Txt KeyPegHolder01:040 [7]
19 NiTriShape	Txt KeyPegHolder01:042 [9]
23 NiTriShape	Txt KeyPegHolder01:043 [10]
27 NiTriShape	Txt KeyPegHolder01:046 [13]
31 NiTriShape	Txt KeyPegHolder01:047 [14]
35 NiTriShape	Txt Key01:0 [15]
39 NiTriShape	Txt Key02:0 [16]
43 NiTriShape	Txt Key03:0 [17]
47 NiTriShape	Txt FirebrandCrate:002 [18]
51 NiTriShape	Txt FirebrandCrate:004 [20]
55 NiTriShape	Txt Key01:001 [22]
59 NiTriShape	Txt Key02:001 [23]
63 NiTriShape	Txt KeyPegHolder01:040 [7]
67 NiTriShape	Txt KeyPegHolder01:040 [7]
71 NiTriShape	Txt KeyPegHolder01:040 [7]
75 NiTriShape	Txt KeyPegHolder01:040 [7]
79 NiTriShape	Txt KeyPegHolder01:040 [7]
83 NiTriShape	Txt FirebrandCrate:002 [18]
87 NiTriShape	Txt FirebrandCrate:004 [20]
91 NiTriShape	Txt FirebrandCrate:1 [6]
92 NiTriShapeData	
93 BSLightingShaderProperty	Txt
Properties	Ref<NiObject>
Properties	Ref<NiObject> 93 [BSLightingShaderProperty]
Properties	Ref<NiObject> None

Cela peut se révéler extrêmement fastidieux si votre mesh comporte des dizaines de NiTriShape, mais nos connaissances s'arrêtent ici. Nous allons maintenant voir comment ajouter une branche de collision à notre mesh.

AJOUT D'UNE BRANCHE DE COLLISION

Nous partons du principe que le modèle de collision est identique au mesh, c'est-à-dire qu'il suit parfaitement ses contours. Nous prendrons donc pour modèle de collision notre Mesh.nif.

Lancez **ChunkMerge.exe**.

Select source Nif-File ✕

- Dans la première fenêtre qui s'ouvre, choisissez le modèle qui vient d'être créé avec NifConvert et auquel nous voulons ajouter une branche de collision, c'est-à-dire **Mesh_converted.nif**.

Select Nif-File for collision data ✕

- Dans la deuxième fenêtre qui s'ouvre, choisissez votre modèle de collision, ici **Mesh.nif**.

ChunkMerge crée alors un fichier du type xxx_mod (ici, **Mesh_converted_mod.nif**) à l'endroit où se situe Mesh_converted.nif : c'est le modèle avec une branche de collision fonctionnelle, prêt à être utilisé dans le CK.

Renommez-le à votre guise et copiez-le dans le dossier Data\meshes de votre installation de Skyrim pour l'ajouter à votre mod (voir section suivante).

ASTUCE – LES VERTEX COLORS

Ce qui suit est seulement le fruit de notre expérience personnelle ; il existe probablement une solution plus professionnelle, mais cette astuce de type « MacBidouille » peut servir.

Si les .nifs que vous importez dans 3ds Max contiennent des **vertex colors** (dans Nifskope, ligne « Has Vertex Colors : yes » pour un ou plusieurs NiTriShapeData) et que vous constatez visuellement un souci lors de l'export (couleur altérée), vous devez cocher la case « Vertex Colors » lors de l'exportation du mesh.

N.B. : Les vertex colors correspondent à la couleur associée à chaque sommet du mesh et cette information fait partie du NiTriShape.

Pendant, si votre objet créé contient des meshes qui possèdent des vertex colors **ET** d'autres qui n'en possèdent pas, il est possible que vous deviez faire une manipulation supplémentaire, à savoir deux exportations, l'une avec, l'autre sans cocher la case « Vertex Colors », et ensuite recomposer votre mesh dans Nifskope pour que seules les parties concernées possèdent des vertex colors.

Par exemple : votre mesh est composé d'une statue et d'un rubis.

- La statue ne possède pas de vertex colors, vous pouvez suivre la procédure habituelle.
- Le rubis possède des vertex colors, tous rouges, mais lors d'une exportation « normale », ils deviennent incolores et votre pierre se retrouve grise.

→ Exportez deux fois votre mesh, l'une en cochant la case « Vertex colors », l'autre sans. Ensuite, dans Nifskope, ouvrez le premier mesh et copiez la branche du rubis. Puis ouvrez le second mesh, supprimez le rubis gris et collez le rubis rouge. Votre statue est maintenant complète, mais seule le rubis possède des *vertex colors*.