



# Sniper's Paradise

## *Who Loves You Baby!!!*

### Sniper's Paradise!

---

#### Unreal Tournament Skin Maker

##### Introduction

Unreal Tournament (UT) skins are made from a maximum of eight textures. The UT models use a maximum of five, and the fifth texture is the face that shows in the message window of other players when you talk to them (Talk Texture). The other textures are used to cover the 3D model of the player. Not all the textures are mandatory, each model has its own format and needs. This program allows you to create the UT files needed to use an skin in the game; it gets some textures and names and it creates two files (INT and UTX) that you have to put in the correct UT subdirectories (or let the program put them automatically).

##### How to create a skin

You will need some graphics program to create the textures. The program supports a lot of formats, but better if you make them in BMP format, and better if in 24 bits. They don't need to be the correct size nor they need to have the same height than width, you can make them bigger and the program will scale them down with a good resampling filter (if not flagged as non-resamplable). They don't have to be named as the final texture objects. You can use the texture template files to help you draw the textures, there should be one file for each section (except for the talk texture) named modelWireN.bmp, being model the model prefix and N the section index. I recommend using a program that support layers (Photoshop, Paintshop Pro) and use the wire texture as a template in a different layer. The program has a rudimentary texture editor to make small changes and view the change immediatly in the preview (only for 24bit textures). See "Texture Editor" below.

Note: If the model has masked areas, your textures will have to be bmps in 8 bit format with the masked color being the first in the palette. In this case, the textures will not be scaled, so make them the correct size.

Note2: The program by default fixes the size of the textures to 256x256. This can be changed by modifying the proper skme file (see below its format). The internal resampling does not modify the texture file, but the resampled texture will go to the UTX file. Do not modify the texture size of the talk texture (it should be 64x64).

If you want to create a skin for a new model (not supported by a fresh install of the program), you will have to install the UTSkinMaker files supplied by the model creator in the CustomModels subdirectory or compile yourself the model to create the needed files. After that, the program will recognize the new model. It will also work if you put them directly in the program directory, but for clarity and version change issues it is preferred to use the CustomModels subdirectory.

Select the New Project option in the menu. Select the model you want to use for the skin (in which you have based your texture files), for example "Male Soldier". Then you will have to select the name for the project, the program will ask you to save it. Please keep the starting prefix because UT finds the files this way (if you change the prefix UT will not load your skin). For example: soldier\_test (extension is .skm). You can add the word "skins" after the prefix, but i found that it is not needed (f.e. soldierskins\_test). Keep in mind that the file is not actually saved at this time.

Now you can add Skins. A skin is the set of textures. Right-click in the soldier\_test element to show the New Skin option and select it. You are prompted to introduce a description and a four-letter code. The description is the text that will appear in the Skin list in UT's character selection dialog. The four-letter code should be unique along all the skins of that model. For some models you should only add one skin (Nali and NaliCow from the BonusPack) because these models do not support more than one set of team textures (but you can add several faces in different skins if you want to because the skin name and description is not used).

After creating the skin you can see that inside it are several elements describing a section of the skin, corresponding to the texture files you already have. In the case of the Male Soldier



model, they are: Chest, Legs, Arms, Head, Talk Texture. Inside these elements you have to insert the textures. If you right-click on any of these elements you will see the Add Texture option. When you add a texture you will be prompted to set the bitmap filename for the texture (one of your already created files) and a texture name. This texture name specifies what will be this texture used for. Depending on the section, the initial options can include: "<Default>" for a default texture (not a team or a face), "<XXX Team>" for the four available color teams, and anything you want to write down (only if the section allows faces). If you write any other text the texture will be created with that description and a code will be generated from that text. This is used to create the specific faces for your skins. Once a description is created it will appear on the list so you can use it for other textures (for example, after creating the Head section you can reuse the description for the Talk Texture section). You can preview the textures clicking on them on the tree. If you edit a texture description it will change that description in all textures with the same old text.

There is an option to Import Textures. This option allows you to use an already filename-formatted texture and create the skin and texture directly. For example, if you already have a sldr4Rambo.bmp file you can import it and create a sldr skin with a Rambo texture in the Head section. This only works for the format name of the UT standard models. It's a fast option to migrate an already created skin to UTSkinMaker.

In the preview panel you can select the skin, team and face as in UT's character selection dialog. You can change the view with the following controls:

Moving the mouse while left-clicking: rotates the model.

Moving the mouse while right-clicking: pans in XY plane.

Moving the mouse while right-clicking and pressing Ctrl key: pans in ZY plane.

Moving the mouse Wheel or the mouse while left+right-clicking: zooms.

You can save the view to a preset using the presets list popup menu. After editing the presets you will have to save them to make them permanent for the current model. You can change the model default position, but this is something you should only do just after compiling a model or if you don't see it completely (although you can also use the view controls in this latter case).

If you think a texture needs a change, you can change the bitmap file and the textures will refresh in a 3 second interval. You can also edit them using the supplied texture editor, the changes will be automatically shown in the preview.

When you have added all the relevant textures and skins you only need to Compile Project to UTX/INT to generate the two files that must be under the UT directory. The UTX file in the Textures subdirectory and the INT file in the System subdirectory. If you select the Compile Project to UTX/INT and Install in UT then the files will be copied for you in the correct subdirectories (overwriting any file with the same name).

You can configure project options in the Project Options tab. You can set if you want to create all the mipmaps or stick with only one. Using only one mipmap reduces the size of the UTX file but it could affect the speed of UT (I have seen no difference in speed or quality with my equipment, but I have a fast computer). An UTX file with only one mipmap is about 3/4 the size of the full mipmaps equivalent. You can also set an option to reduce the number of palettes created in the package. This can also reduce file size a bit, but the quality loss is noticeable. Any masked texture will have its own palette regardless of this settings. The preview panel will show the aspect using the palettes generated with this option so you can see the result. Another option is one that sets the package as a client optional package so that when connecting to a server with this skin it is not necessarily downloaded (however, I'm not sure how this works). The skins for BonusPack models have the flag set because a client does not have to have the BonusPack.

## Texture Editor

The texture editor allows you to retouch the textures while you see them in the preview panel. It is not comparable to a full-blown graphics application, but it allows you to make some changes. The supported tools are:

Brush: paints a brush of the specified size and intensity in the selected color.

Tint: same as brush but changes only the hue component. Good to make team textures for example.

Lighten: same but increases the texture intensity.

Darken: same but decreases the texture intensity.>

Picker: click on the texture and the selected color will change to the one the pixel you clicked have.

Save: save the texture (when saving the project it will ask you for any unsaved texture).

Revert: reverts the texture to the saved one.

Undo: undoes the last change.

Zoom: ditto, 1X,2X,4X.

Color: select the color to work with.

Brush Size: select the brush size (1-64)

Brush Intensity: select the brush intensity (1-100). 1=nearly transparent, 100= full solid.

Wire Overlay: this control allows you see the Wireframe texture so that you can paint within the limits of the polygons.

Any change to the texture (with the tools when you release the mouse button) will be automatically updated in the preview panel. This can take some seconds because it reloads all the textures.

How to configure the program to work with a model

You can configure the program to allow creating skins for any custom model. The program will read any file with an skme extension (SKinmaker Model Extension) which finds in the program directory or the CustomModels subdirectory (this one preferred). These files contain some information about the model needed by the program to allow creating skins for them.

You have to compile the model (Direct UT LodMesh or Unreal 3D, 3DStudio, MilkShape3D or Painter3D files) with the Model Compiler menu option to generate the needed files. This creates at least three files, one is the skme file filled with most of the data (though it always needs some editing, see below), an skmm extension file (the 3D model) and some bitmaps with their filename ended in "Wire" and a number (the wireframe templates for the textures). Just deploy all the files with your model and anyone using the same version of UT Skin Maker could use them to create skins and preview them (by copying the files in the CustomModels subdirectory). Since this version has support for Direct UT LodMesh compiling, anyone can compile any model installed in UT.

Using Direct UT LodMesh:

This option allows you to compile a model from a Mesh or LodMesh object inside an UT package file. This is the preferred way of compiling a model, since it works with any model installed in UT (and some other unreal engine games).

Select the game from the list. You can specify if you want to use INT files (each game has a predefined setting for this, but you can always uncheck the option if you want to read directly a package file; it's preferred to left it as the game selection sets it). Select the package file from the game System subdirectory (should be the default directory in the dialog box). Then select one of the models in the list (the first is selected by default). Then select the animation sequence and the frame number (of that sequence), by default the "Breath1" sequence is selected if it exists, or select an absolute frame number (frame 52 seems to be a good one). Then press the " Compile" button. The files are created in your CustomModels subdirectory and you only have to edit manually the skme file to set/fix some values (see SKME file below).

Using model files:

Remember that the preferred method of compiling is using Direct UT LodMesh compiling.

There are several ways to create a model, but with all of them you have to create Unreal 3D files:

Since the Model Compiler can import from the model\_D.3D and model\_A.3D files, you can always import a model, and this is the preferred format (except for direct lodmesh compiling of course). But you can also import from the original formats (3DStudio, MilkShape3D, Painter3D). The Painter3D support is the worst because it does only record one material with all the model mappings, so the format is very restricted to be the same as the standard UT models. This format is mainly supported because the standard models had been distributed in this format.

First you select the model file. Then you have to select the mesh/frame number (if there is more than one). Select one that has the model in a relaxed position (one of the breathing frames for example). Then press the Get Mesh Data button, and a list of materials will show in the list. Reorder the materials if needed, putting the weapon material at the end; do not move any other material. This list will match with the texture sections. Finally, press the Compile button, and all the files will be created (and overwritten). Press the Finish button to

exit this window.

In case you import from the Unreal 3D files the materials are read with names SKINn (being n a number starting with 0), since that format does not record material names. The compiler will suppose that material 0 ("SKIN0") is the "face" material and it will put it the first in the list.

In case you import from a 3DStudio file, the materials could be named as for use with 3DS2UNR or Digital Extremes Exporter and the program will recognize both material names schemas. For 3DS2UNR, the "face" material is "SKIN", and it supports the suffixes: ".WEAP" and ".MASK". For DE Exporter, the material names are SKIN0 to SKIN9 (the program supposes SKIN0 is the "face" material) and the suffixes are ".WEAPON" and ".MASK". The other possible flags (translucent, twosided, modulate, flat, unlit, environment mapped, no smooth) are not used right now.

In case you are importing from a MilkShape3D file, there is no material name convention, so the program uses the same as for the others. That is, the "SKIN" or "SKIN0" material is the "face" and it reads the flags as with 3DS2UNR (the same for name suffixes).

SKME file:

This file will need some editing after compiling. The parts that need editing are the positional parameters, that can be modified inside the preview panel, and the Formats and Sections values. The Formats will probably be ok for most models, but the Sections have to be edited to put the correct descriptions and Face/Team/Talktexture/etc flags.

This is an example of an skme file:

; The text after the "Model-" is irrelevant.

[Model-BonusPack-SkaarjTrooper]

; The Name value is used as the description for the model. Try to use the menu name for the model.

Name=BonusPack - Skaarj Trooper

; The Prefix value is used as the prefix for the skin files.

Prefix=tskm

; The 3DFile value specify the name of the file containing the model data.

3DFile=tskm.skmm>

; The LodMesh value specify the name of the mesh/lodmesh object in UT (only exists if compiled from UT).

LodMesh=package.object

; You will need to edit the following values to adjust the size and position of the model in the preview panel.

ScaleX=1.0

ScaleY=1.0

ScaleZ=1.0

OffsetX=0.0

OffsetY=0.0

OffsetZ=0.0

Pitch=0.0

Roll=0.0

Turn=0.0

; The following format strings are used to generate the texture names, here are shown the defaults.

; For example, the Nali and NaliCow models have different values here.

; This value depends exclusively on the SetMultiSkin unrealscript function. If the model has overridden this function the values must change.

; You can use the following variables:

; %SKIN% = The skin four letter code

; %SECTION% = The section index

; %TEAM% = Theteam index (0-3)

; %TEAMNAME% = The team name (Red/Blue/Green/Gold)

; %FACE% = The face name

FormatDefault=%SKIN%%SECTION%

FormatTeam=%SKIN%%SECTION%T\_%TEAM%

FormatFace=%SKIN%%SECTION%%FACE%

FormatTalk=%SKIN%%SECTION%%FACE%

; TalkTexture specifies the index of the texture used as the "talk texture" in in-game messages.

TalkTexture=5

; Here comes the sections that make an skin. For each section you have to write its name followed by a comma-separated list of flags.

; These flags can be:

; Team (the section allows team textures)

; Face (the section allows faces)

; OneMipMap (the textures in this section will have only one mipmap)

; LODSetWorld (the section creates textures with a LODSet value of "World")

; Masked (the section has masked polygons/textures)

; CannotResample (the textures will not be resampled when loaded)

; TextureIsWxH (the textures in this part are WxH pixels and will be resampled if needed; substitute W and H with the values)

; NoINT (the section does not generate a line in the INT file)

; TalkTexture (the section is the "Talk Texture")

; Current UT textures (except for the Talk Texture) are 256x256 in size.

; OneMipMap and LODSetWorld flags are used for Talk Texture sections. The texture size in this section is always 64x64.

; The index of the section corresponds to the texture object needed by UT. Some models do not need some indices.

Section2=Head, Face

Section3=Arms & Legs, Team

Section4=Chest, Team

Section5=Talk Texture, Face, OneMipMap, LODSetWorld, TextureIs64x64, TalkTexture

; Presets. You can add any number of presets here. If there is no presets the program will create one front view.

; The semicolon-separated values are: name, camera x, camera y, camera z, target x, target y, target z

; You can create this entries from within the preview panel.

Preset1=Center;0.0;0.0;2000.0;0.0;0.0;0.0

The Formats and TalkTexture values depend on the SetMultiSkin unrealscript function. And the Sections depend on the FaceSkin, FixedSkin, TeamSkin1 and TeamSkin2 values. The program will try to put the correct Face & Team flags to each section.

To adjust the model size and position you can use the following keys in the preview panel (lower case decreases the value, upper case increases the value):

X = X value

Y = Y value

Z = Z value

P = Pitch value

R = Roll value

T = Turn value

S = Scale values

A = shows/hides a surrounding cube and axes to allow precise centering

The statusbar of the preview panel will change showing the current values. Remember that when using the rotation keys you have to rotate in the following order: Pitch, Roll, Turn or it will load incorrectly. Use the Increments applied radio buttons to set the increments used by the keys. When the model is centered and scaled save that values using the Save button in the Model Default Position box.

Provided with the program are the extensions for Male Commando, Female Commando, Male Soldier, Female Soldier, Boss, SkaarjTrooper, NaliCow and Nali models.

## Screenshots



## Features

The program allows you to create a skin project using any model installed in UT'99 and then add the textures. Then you can easily edit the descriptions of the textures and finally create the UTX/INT files. The program takes care of creating the file with the correct setting for things like the LODSet property and creates the MipMaps for the textures.

Now with 3D skin preview and UT model compiling!!!

## Known Issues

Texture corruption. Sometimes a texture may get corrupted on the model preview, press the Refresh button until it is correct.

Texture disappears. Sometimes a texture may not show after a change in the "Enable Masking" checkbox. Reload the project to correct.

May give an wglMakeCurrent error at program exit.

## Notes

The project options (One Mipmap & Palette reduction) can be used to reduce file size to a minimum. This are the sizes for a regular skin (soldier model with team colors for chest and legs and one face, 13 textures in total):

Mipmaps\Reduce Palettes	One Palette per Texture	One Palette per Group or One Palette for all the Package (because there is only one group)
All the MipMaps	1.069.356 bytes	1.056.671 bytes
Only One MipMap	805.702 bytes	793.018 bytes

As you can see, the most reducing option is the MipMaps one. This is because the palettes only have 1027 bytes each, and a typical first mipmap is about 64Kb, with the all-mipmaps texture being about 87Kb (23Kb more). In the example, the file size was reduced from (aprox.) 1Mb to 790Kb. The quality of the skin is basically influenced from the palettes options. Allthouth the algorithm used to reduce the palettes does a very good work, some quality is lost in the process, and some pixelation could be seen due to the reduced number of colors available. The pixelation can be easily seen in the character selection dialog, but when playing will not be seen so much. The reduction of mipmaps is usually undetected because UT seems to create them internally.

## License

This software comes without any guarantee of any kind. I do not take any responsibility for any damage, loss of income, or any other problems you might experience from using this software. You might not charge anything for the distribution of this software, but the costs of media, shipping, etc.

## Thanks

Erik de Neve from Epic Games.  
Scott Martin from Ion Storm.  
My betatesters. They know who they are.  
All the users that gave me ideas and bug reports.



## Back To Top

---

2005 Sniper's Paradise  
All logos and trademarks are properties of their respective owners.  
Unreal™ is a registered trademark of Epic Games Inc.

[Privacy Policy](#)  
Website by Softly  
Powered by RUSH