EA official tutorials:
Terrain editor
Leveleditor
Custom Mod Setup
Understanding Heightmaps
Build the Bunker - 3ds Max
Build the Bunker - Maya
Import the Bunker
Build the Car - Maya
Import the Car
Create new team - Maya
Create new team - 3ds Max
Create new kits - Maya and Max

Mapping - General
Setup FH2/bf2editor
FH2 mapping guide
FH2 mapping standards

Mapping - Texturing
Texture system in bf2
Terragen skies
Terragen texturing

Mapping - Heightmaps
World Machine
Geocontrol and bf2hmt

Mapping - Lightmapping
Simple lightmap setup

Mapping - Downloads
Sky settings bf2
Water settings bf2
Lowdetailtextures/settings bf2

3D modelling/texturing
Skinning the Daimler Dingo

Coding

FORGOTTEN HOPE 2
Mapping tutorials

Creating a New Team for Battlefield 2 – Maya Version

by [EA]Lawrence Brown

Download tutorial files

Updated: 10/11/2005

This tutorial will cover all steps necessary to create a new team and get it functional in the game.

This is considered an advanced tutorial and should be attempted only after you have become familiar with how modding BF2 and the Editor work. The process isn't particularly difficult, but there are many steps involved, and missing one or performin step incorrectly can result in unexpected results or the game crashing. Please try to be patient and double-check your work.

Set-Up

1. Open up the tutorial files folder you downloaded and locate the "1p_setup.mb" file. This is an updated version of the fil so copy this to "…\Rawdata\Animation\Setup\1p\1p_setup.mb", overwriting the version already there.

This tutorial assumes that you have created a new player model and textured it. The model should consist of three meshes: body, the head, and the hands. Note that the hands should be merged together and considered as one mesh.

For the rest of the tutorial, we will be working with the "light" soldier. All the same steps apply to the "heavy" soldier, just substitute "heavy" where appropriate.

The textures for the body and hands should be contained on one 1024x1024 image. The head should be on a separate 512x25 image. You may want to unzip the retail BF2 Objects_client.zip to somewhere on your hard drive (not in the game) and look the existing textures for reference.

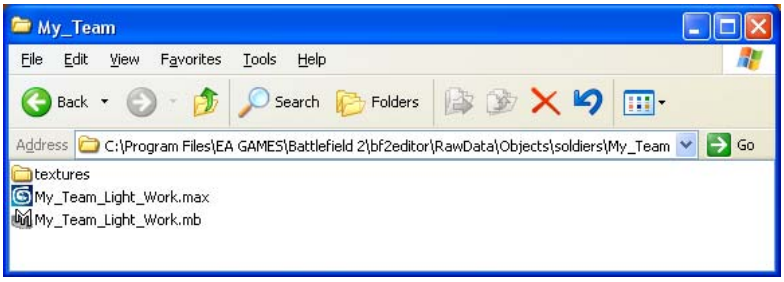
If you have not created the models for your team, you can use the tutorial files provided. After downloading, place the folde "My_ Team " in bf2editor\RawData\Objects\soldiers\My_ Team . Create any folders necessary to create this path. We'll deal with the other tutorial files as we go.

Double-Checking your textures and Paths:

The first thing we're going to do is double-check that you have the textures set up right and that everything is in the correct path.

2. The path to your work files should be as follows:

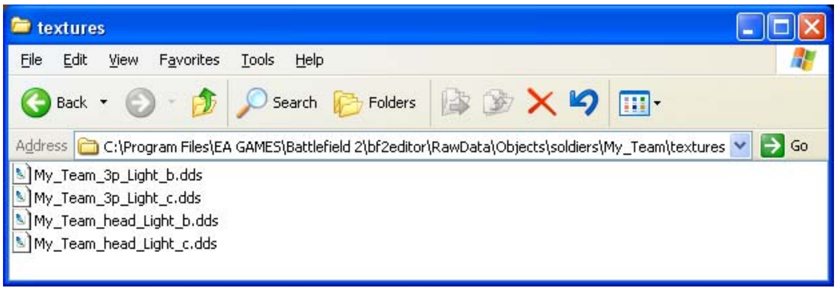
C:\\Program Files\EA G ames\Battlefield 2\ bf2editor\RawData\Objects\soldiers\My_ Team .



C:\Program Files\EA GAMES\Battlefield 2\bf2editor\RawData\Objects\soldiers\My_Team

textures
My_Team_Light_Work.max
My_Team_Light_Work.mb

3. Inside the "My_ Team " folder should be the "textures" folder:



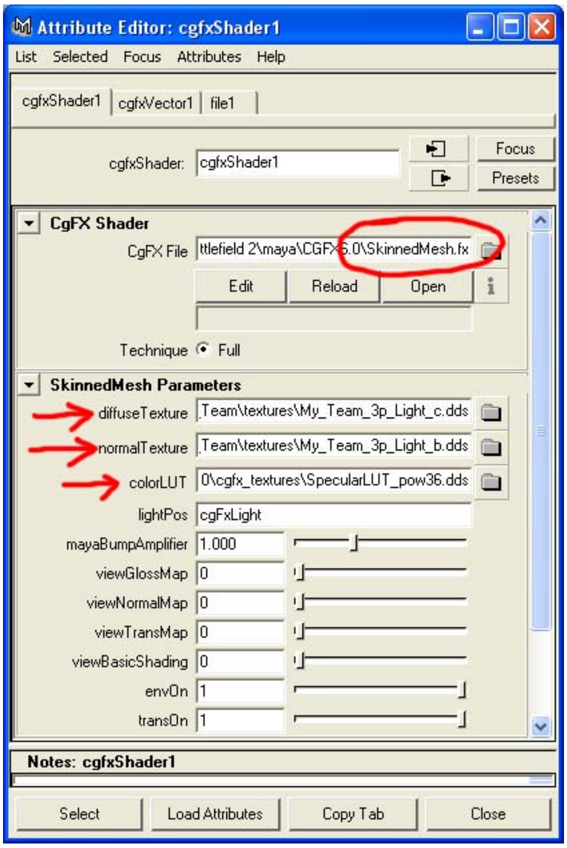
C:\Program Files\EA GAMES\Battlefield 2\bf2editor\RawData\Objects\soldiers\My_Team\textures

My_Team_3p_Light_b.dds
My_Team_3p_Light_c.dds
My_Team_head_Light_b.dds
My_Team_head_Light_c.dds

Note that you should have both a color texture and a normal map for both the body and the head. The normal map is what should provide small details such as wrinkles in clothing and so on.

4. Open your work file. If you are using the tutorial files, this will be "My_ Team _Light_Work". I add "Work" to the end o the label of my initial file so that I know this is the original. This way it won't be confused with the final version we create. You may want to label your files the same.

5. Check to be sure that you are using the SkinnedMesh cgFX shader and that you have both the diffuse and normal chanı hooked up correctly. Also check to be sure the colorLUT path is correct.



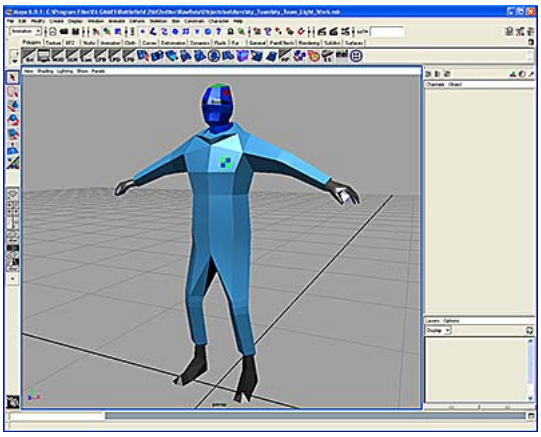
cgfxShader: cgfxShader1

CgFX File: ttlefield 2\maya\CGFX6.0\SkinnedMesh.fx

Technique ● Full

diffuseTexture: Team\textures\My_Team_3p_Light_c.dds
normalTexture: Team\textures\My_Team_3p_Light_b.dds
colorLUT: 0\cgfx_textures\SpecularLUT_pow36.dds
lightPos: cgFxLight
mayaBumpAmplifier: 1.000
viewGlossMap: 0
viewNormalMap: 0
viewTransMap: 0
viewBasicShading: 0
envOn: 1
transOn: 1

6. Lastly, your new player model should be in the classic "DaVinci" pose so that it matches the skeleton. If you used the existing soldier as a reference, this shouldn't be a problem. If you are using the tutorial file, it should look like this:



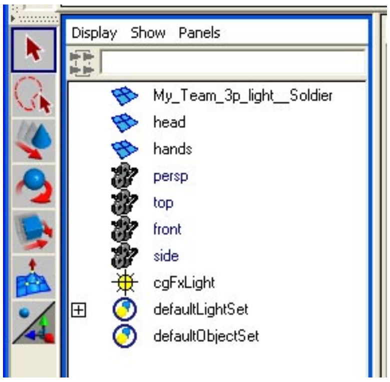
Creating the Third-Person File

Before we can get the new team into the game, we need to create both a third-person and First-person file. We are going to start with the third-person file.

Naming the meshes

The first thing we want to do is name the meshes correctly because this will be important later on.

7. Select the body mesh and rename it "My_ Team _3p_light__Soldier". If you are using your own player model instead o the tutorial file, replace "My_ Team " with the name of your team. The rest of the name needs to be exactly the same. Note that "Soldier" is capitalized and there is a double-underscore before it. These are very important.
8. Select the head mesh and rename it "head".
9. Select the hands mesh and rename it "hands":

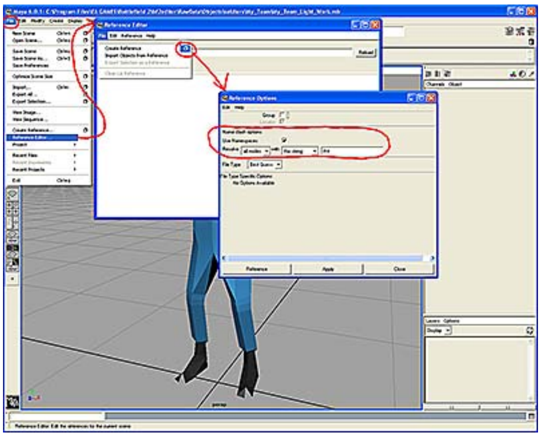


My_Team_3p_light__Soldier
head
hands
persp
top
front
side
cgFxLight
defaultLightSet
defaultObjectSet

Referencing the Skeleton

With everything named correctly, the next step is to reference the skeleton into the scene.

10. Select "File>Reference Editor…" to open up the Reference Editor window.
11. In the Reference Editor window, select "File>Create Reference>Options".
12. In the Options window, make sure that "Resolve all nodes with this string: ske" is selected in the "Name clash options" section. Refer to the following screenshot for details:



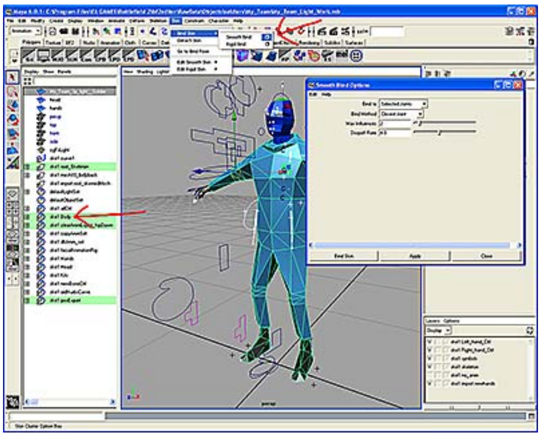
13. Once you have all the options set, click on the "Reference" button at the bottom of the options window and select "…\Rawdata\Animation\Setup\3p\3p_setup.mb".
14. Open up the outliner window. You should now see several sets, all starting with "ske1:".

Skinning the Meshes

Now that we have the skeleton referenced, it's time to skin the meshes to it.

15. In the outliner, find the set labeled "ske1:Body".
16. Select it, and then Right-click on it and choose "Select Set Members". This selects all the bones in the set rather than j the top node.
17. With the bones now selected, Shift-Select the body mesh from the main window. It has to be picked last in order for th to work. You must also be sure to select it from the main window rather than the outliner or the selection will get mes up.
18. Make sure that you have switched to the animation sub-menu in the drop-down box at the top-left, then select "Skin>E Skin>Smooth Bind>Options". (You can also do this with the marking menu.)
19. Set the following options:

    - Bind to: Selected Joints
    - Bind Method: Closest Joint
    - Max Influences: 2
    - Dropoff Rate: 4



20. Do the same for the head and the hands using the “ske1:Head” and “ske1:Hands” sets.
21. Save your work now to avoid having to start over. The naming convention should be “My_ Team _light_3p” if you are using the tutorial files. Substitute the name of your team for “My_ Team ” if you are using your own custom model.

Weighting the skin

Now that you’ve skinned your meshes, you will probably want to adjust the weights. G et a feel for how your skinned charac behaves by selecting the different controllers and moving them around. You’ll probably want to adjust the weights on the various vertices to get better deformations, especially around the face area. In the case of my tutorial files, I’ve gone with a helmeted approach to keep things as simple as possible.

It is beyond the scope of this tutorial to teach the weighting of vertices. There are many different techniques, and you shoulc use the one that works best for you.

Note: You do not have to get the weights perfect now. It may be preferable to wait until we get to viewing your player skin v animations in the editor and in-game, then come back and fix the trouble spots.

Additional LOD ’s

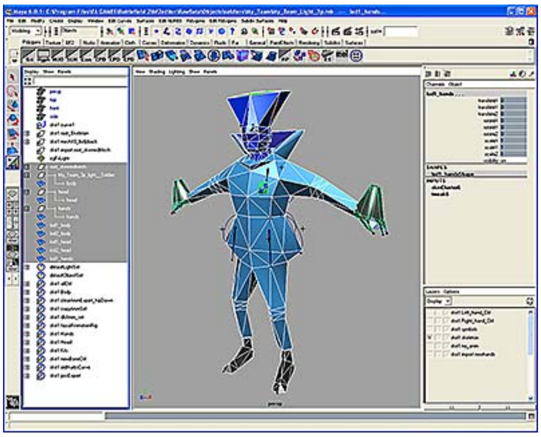
Note: This is an optional step that you may want to put off until after you have successfully imported your first team.

When you get to the point where you would like to create additional LOD ’s, follow this procedure:

22. Create your additional geometries however you want and name them as follows:
    - lod1_body
    - lod2_body
    - lod1_head
    - lod2_head
    - lod1_hands

You want 3 LOD ’s total. The meshes you previously skinned are considered lod0. Notice that there is an lod1_hands, but not lod2_hands. This is because at the distance these would switch in, you wouldn’t see them anyway, so they aren’t necessary.

23. Once you have created the additional LOD ’s, smooth bind them exactly the same way you did the other meshes. The c difference is that you do not parent them to anything in the geometry. They just remain separate for now. If you use additional LOD ’s, your hierarchy will look like this:



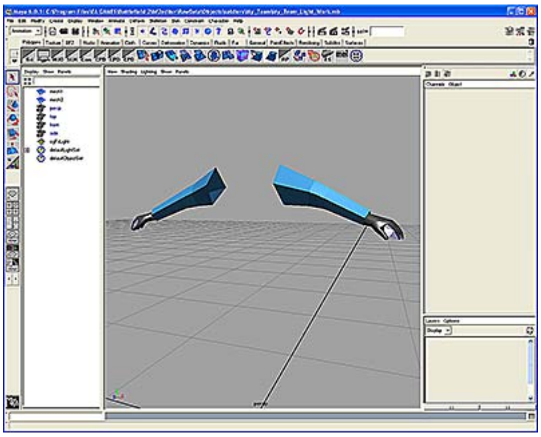
Notice that I’ve created some pretty whacky meshes here. The reason is because this is a test and this way I can see from a distance when the LOD ’s change.

Also remember that there wouldn't be any additional LOD 's for the first-person meshes because only the player will see ther and the distance to the camera never changes.

Creating the First-Person Meshes

The third- and first-person versions will be stored in seperate files and then referenced into yet a third scene. Before we can all this referencing, we have to create the first-person file using the third-person file as a start.

24. Open up the original work file again, the one that is just the model without the skeleton referenced into it. If you are using the tutorial files, this will be "My_ Team _Light_Work".
25. For the body mesh, delete all the polygons except those used in the arms. These are all you will see from the first-per view, so the rest is not needed.
26. Delete the head. This won't be seen at all. You should now be left with this:



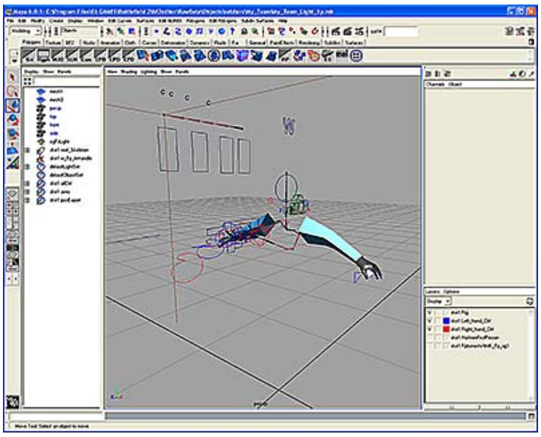
27. Before going any further, save this file as "My_ Team _light_1p". This way you won't accidentally overwrite your origin file.

Referencing the First-Person Skeleton

The steps for referencing the skeleton are the same as before, only this time we are using the first-person version.

28. Select "File>Reference Editor…" to open up the Reference Editor window.
29. Make sure that all options are set the same as previously, then click on the "reference" button and select the file "…\Rawdata\Animation\Setup\1p\1p_setup.mb".

Note: Make sure that you are using the updated 1p setup file included with the tutorial files. If you use the old version, not a necessary sets will be present. You should now have this:

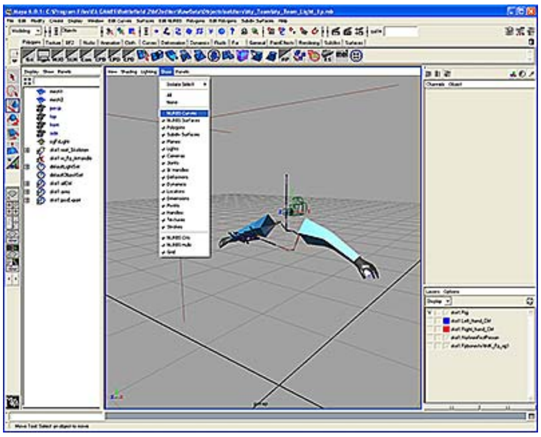


Posing the arms

This is the fun part. Obviously the arms and hands don't match the skeleton, so we have to reposition them so that they do. reason we are doing this the "hard way" is because the bones are posed in the position that you will see in the first-person v 90 percent of the time. If we skinned the meshes to the arm bones in the same position they were in before and then movec them, they wouldn't deform correctly. It's just one of those things that animators have to deal with.

Before we do anything else, let's hide a lot of bits that we don't need right now.

30. In the Layers window at the bottom-right, click on the "V" next to everything except "ske1:Rig" This turns visibility off everything but what's in that layer.
31. Next, in the main window, select "Show>Nurbs Curves" to deselect it. (This is an on-off toggle and you'll have to do th for each of the windows you use):

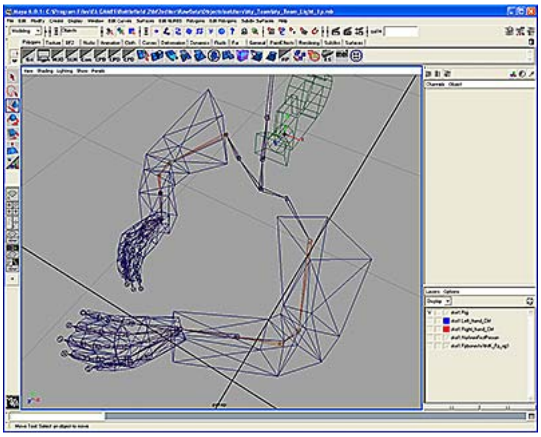


You are now ready to do the posing.

32. Switch to vertex selection mode and move the vertices of the arms so that they line up with the skeleton. Don't worry you don't get them exactly perfect or if there is a bit of stretching and so on. Remember, this will only be seen from y first-person view. The most important parts are the ends of the sleeves. You will probably never even see the shoulder

Again, you don't necessarily have to spend a lot of time right now getting everything perfect. You can always come back late and readjust anything that looks off. Also be aware that once you view this in the editor or in the game, you may decide that you need to make the sleeves larger or distorted in other ways to make them appear correctly. The first-person meshes do r need to match up with the third-person version at all. The important thing is that it looks correct in the game.

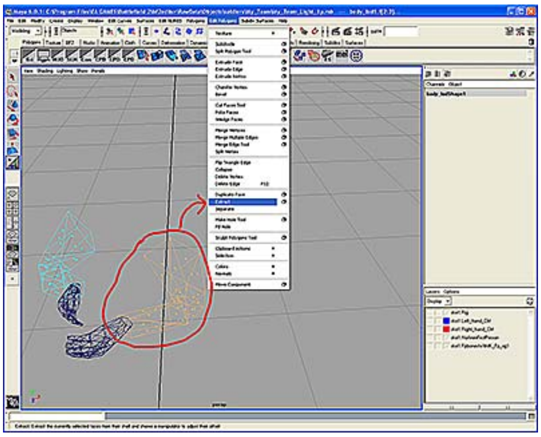
You should now have something like this:



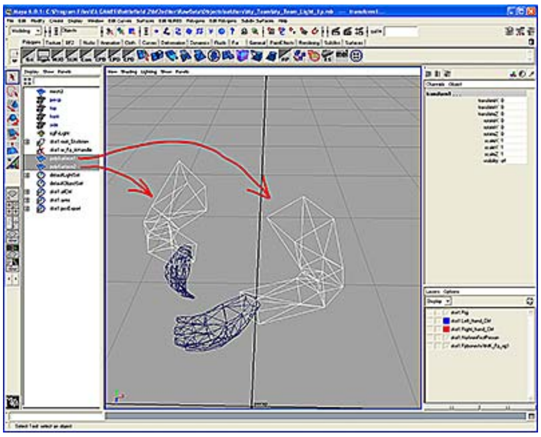
Redefining the pieces

When we set up the third-person pieces, the arms were part of the body and the hands were considered one separate piece. need to change things so that we have just a left and right arm.

33. Hide the skeleton by clicking on the "V" next to "ske1:Rig" in the layer window at the bottom-right of the screen. This toggles visibility on and off. (This is necessary because of Maya's selection priority. If the skeleton is visible, it will pic the skeleton instead of the faces when you use drag-select.)
34. Select the arms, then Right-Click and choose "face" to switch to face selection mode.
35. In face selection mode, select all the faces of either the left or right arm. (Not both!)
36. Make sure the modeling submenu is chosen from the drop-down box at the top-left and then choose "Edit Polygons>Extract". This will split the selected polygons into a new piece:



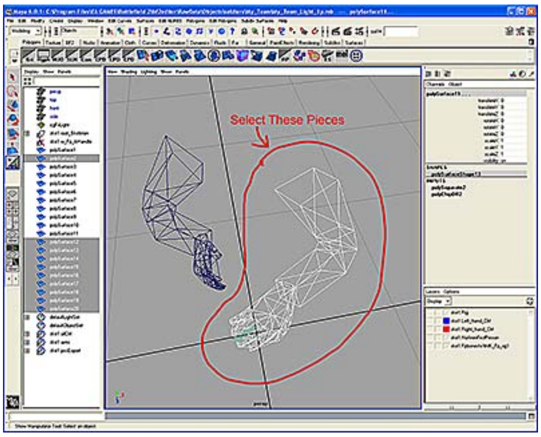
37. After you perform this operation, a new group node is created, but the individual pieces should still be highlighted. Do deselect them yet. We have to ungroup them, but before we do that, select "Edit>Delete by Type>History".
38. With the history deleted and the pieces still selected, press the Up arrow key to select the group node. (This will cause both pieces to be G reen in wireframe.)
39. Select "Edit>Ungroup" from the main menu. You should now have two separate pieces. (They should now both be whit wireframe.) You should now have two individual, ungrouped pieces with no history. Double-check this by looking at the Outliner:



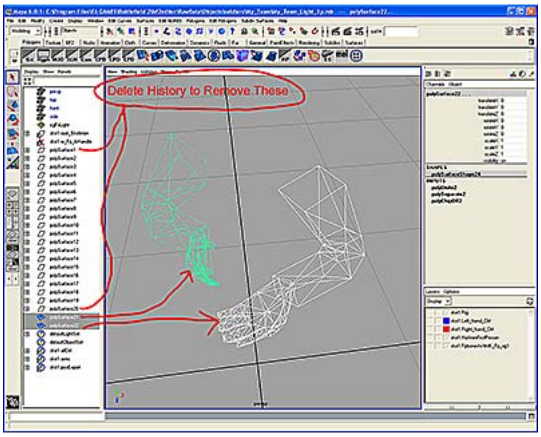
40. Now do the same thing with the hands.

Note: If you are using my tutorial files, you may end up with several other pieces besides the two hands. The extra pieces would be the webbing I put between the fingers. (just for fun) This is fine for now, so don't worry about it. Just make sure t you ungroup all the pieces and delete the leftover transforms as before.

41. Maya may switch you to component mode automatically when you do the extractions. If so, switch back to object mod then select all the pieces that make up the left arm and hand. (Make sure you are selecting whole pieces and not faces Your selection should look like this:



42. Select "Polygons>Combine" from the main menu. You should now have one mesh.
43. Do the same for all the pieces of the right arm.
44. You should now have two meshes, one for each arm, and you will probably notice a bunch of leftover nodes. This happ if you don't delete history before doing the merges. To be safe, select both arms now and delete history to get rid of a the extra nodes and to insure that you have "clean" geometry for later:



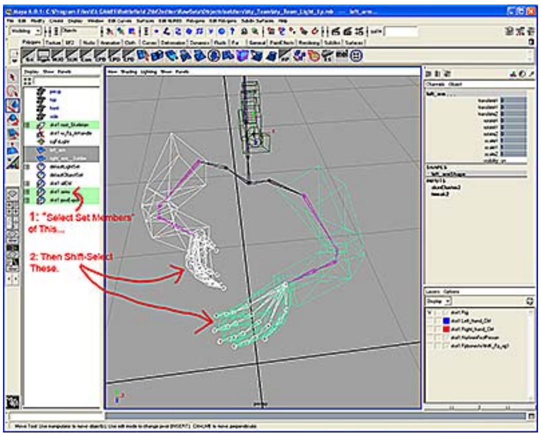
45. Rename the pieces "left_arm" and "right_arm__Soldier". It's very important that you name these pieces exactly becau the importer will be looking for these names. Note the double-underscore before "Soldier". Also note that "Soldier" is capitalized. (And no, "left_arm" should not have "__Soldier" at the end.)

Skinning the First-Person Arms

We are finally ready to do the actual skinning.

46. Make “ske1:Rig” visible again by toggling the visibility button that we clicked on before.
47. In the Outliner, select “ske1:arms”.
48. With “ske1:arms” selected, Right-Click on it and choose “Select Set Members”.
49. In the main window, Shift-Click on the two arms to add them to the selection.

Note: You must Shift-Click on the arms in the main window. If you try to do so in the outliner, you will lose your original skeleton selection:


1: "Select Set Members" of This...
2: Then Shift-Select These.

50. With everything selected, do a Smooth Bind operation the same way you did for the third-person player. Use all the sa option settings.

Checking your work

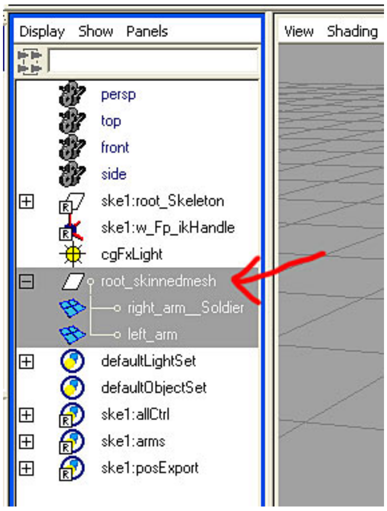
You have now bound the arms to the first-person skeleton. You may wish to check your work and adjust the weighting as necessary before moving on.

51. In the layers window, click the visibility box to the left of “ske1:NoAnimFirstPerson” to make the layer visible. This wil unhide several IK handles and some locators.
52. Select the locator (not the IK handle) for the left hand and move it about to see how your arm deforms. This is connec to the wrist joint, so you can also rotate it to move the hand.
53. Save the file before moving any of the joints just to be sure you can get back to the bind pose, then fix any weighting don’t like. Notice that the end of the sleeve will rotate and deform when you turn the hand. You may want to re-weight this so that the arm joints don’t affect the hand mesh and the wrist joint doesn’t affect the sleeve.
54. Once you are ready to move on, click the visibility box next to “ske1:NoAnimFirstPerson” again to make the scene less cluttered.

Setting up the First-Person Heirarchy

We have skinned the meshes, but we now have to create a hierarchy so that the importer knows what to do when we import into the editor.

55. Select the right arm mesh (not the skeleton), then Shift-Click on the left arm mesh to add it to the selection.
56. G roup the two together and name the new node “root_skinnedmesh”.
57. In the outliner, check to be sure that the right arm comes before the left arm in the hierarchy. If you followed step 48, then this should already be correct. If they are in the wrong order, switch them now. The hierarchy should look like thi when you’re done:



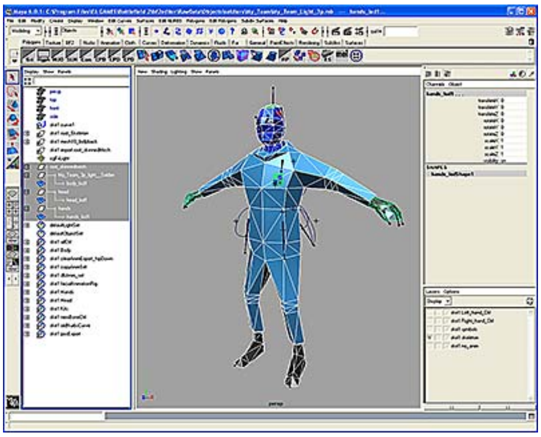
58. Save your work again, making sure that the file is in “RawData\Objects\soldiers\My_ Team ” with the others.

Setting up the Third-Person Hierarchy

We now have to set up the hierarchy of the third-person meshes.

59. Close the first-person file and reopen "My_ Team _light_3p".
60. Create an empty node by either selecting "Create>Empty G roup" from the main menu or by pressing Cntrl- G with nothing selected.
61. Rename this empty group node "root_skinnedmesh".
62. Create another empty node and rename it "My_ Team _3p_light__Soldier". As before, replace "My_ Team " with the na of your team and be sure that there is a double-underscore at the beginning of "__Soldier". Also pay attention to the capitalization of "Soldier".
63. Parent My_ Team _3p_light__Soldier to root_skinnedmesh.
64. Create two more empty nodes and rename them "head" and "hands".
65. Parent these two nodes also to root_skinnedmesh.
66. Parent the actual body mesh under the "My_ Team _3p_light__Soldier" node.
67. Parent the head mesh to the "head" node.
68. Parent the hands mesh to the "hands" node.

You should now have a hierarchy that looks like this:



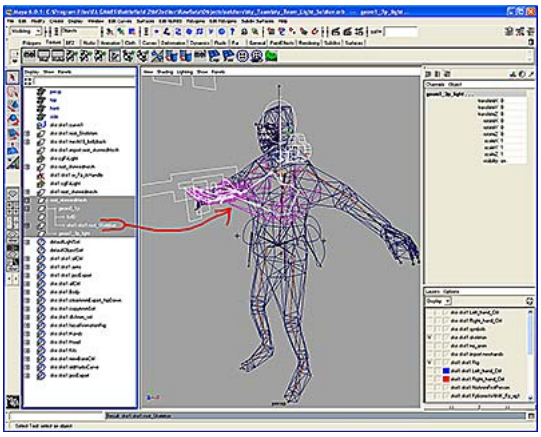
69. Save your scene again.

Creating the Soldier File

We've created the first- and third-person meshes, skinned them to the skeletons, and set up the hierarchies. The next step i reference both of these files into a third scene, which is the one that will be used for the actual import into the editor.

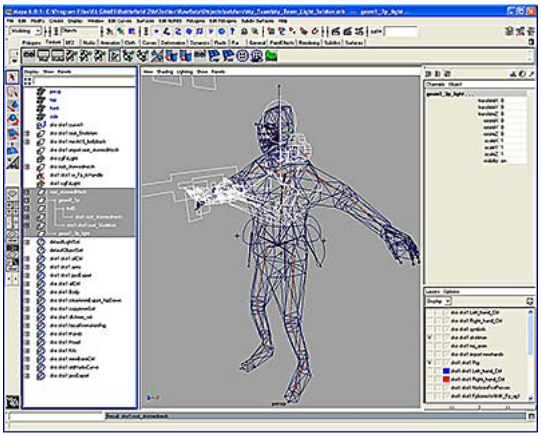
70. Create a new, empty scene.
71. Reference "My_ Team _light_3p" and "My_ Team _light_1p" into the new scene with the same options as before.
72. Create an empty node as before and rename it "root_skinnedMesh". Note the capitalization.
73. Create two more nodes and rename them "geom0_1p" and "geom1_3p_light".
74. Parent these two nodes to the "root_skinnedMesh" node.
75. Create another node, rename it "lod0" and place it under the "geom0_1p" node.
76. In the outliner there will be two skeleton hierarchies, both ending with "root_Skeleton". Click on each of these and find the one for the first-person skeleton.
77. Parent this hierarchy under "geom0_1p".

Note: it is very important that you get the right hierarchy under the right geom node. If you get them reversed, you will get errors upon importing. It's also important that you parent these to the geom nodes, not the lod node. If you have done everything correctly so far, your outliner should look like this:



Notice that in my example, the root_Skeleton node name starts with "ske1:ske1:" This is a product of the referencing proces If your looks a bit different, it's okay. Just make sure it's the right node and that you parent it to the right place. In this case the importer just looks for the "root_Skeleton" text at the end of the name.

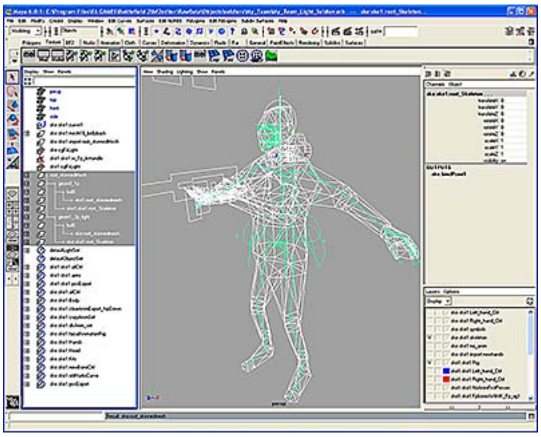
78. Now find the hierarchy whose name ends in "root_skinnedmesh" that is for the first-person arms. (Pay close attention the names. The previous hierarchy was for the skeleton. This one is for the meshes.)
79. Parent this node under "lod0". Your hierarchy should now look like this:



Now we do the same for the third-person files.

80. Create another empty node and again call it "lod0" and parent it under "geom1_3p_light".
81. Parent the other "root_Skeleton" also under "geom1_3p_light". (Double check to make sure this is the third-person skeleton.)
82. Parent the second "root_skinnedmesh" hierarchy under the "lod0" node you just created. (Double check this, too, to ma sure it's the third-person mesh.)
83. Save this file as "My_ Team _Light_Soldier". Save it to the same directory as the previous files.

You have now completed the Maya creation and set-up process. You should check your hierarchy to make sure it matches th following exactly. Remember that the "ske" portions of the names may vary. The important thing is that the portion of the names beginning with "root_" match exactly:

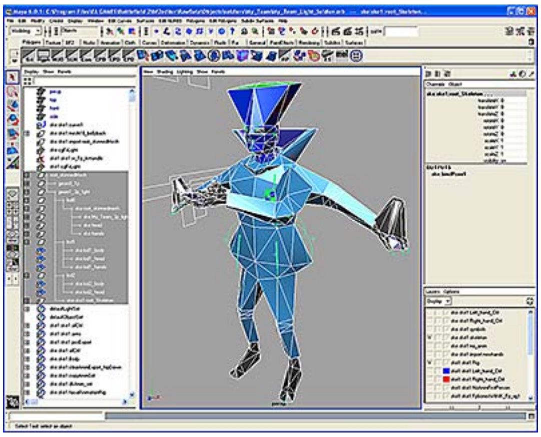


Skinning the Additional LOD 's

This is another optional step. If you had previously created and smooth bound the additional LOD 's in your third-person scer then you must parent these now.

84. Under "geom1_3p_light", create another node and rename it "lod1".
85. Find the meshes with the names that end in "lod1_body", "lod1_head", and "lod1_hands" and parent them all under "lo
86. Create another node under "geom1_3p_light" and rename it "lod2".
87. Find "lod2_body" and "lod2_head" and parent them under "lod2".

Your hierarchy should now look like this:

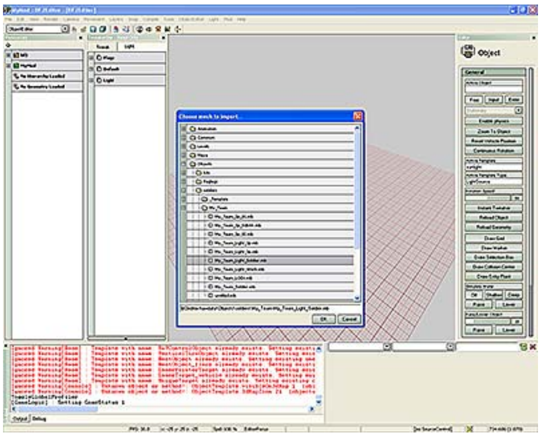


Note again that I've exaggerated the extra LODs here as a test so that I can see when they switch. You are now ready to import.

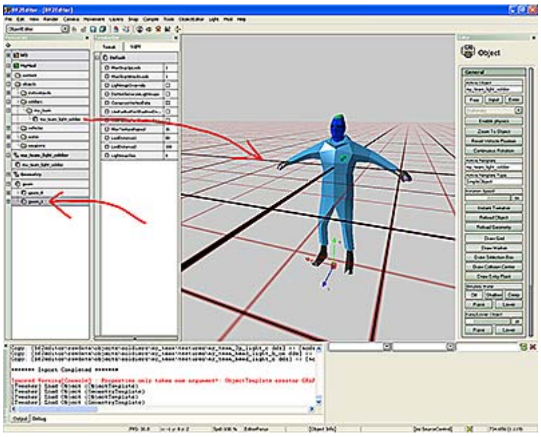
Importing Into the Editor

You've completed set up of the files. It's now time to import your new soldier into the editor.

88. Quit Maya if you haven't already and launch the Editor. Be sure to select the mod you are going to be putting the soldie into when the editor starts.
89. Switch to the Object Editor subsection and select "File>Import".
90. Select "My_ Team _Light_Soldier" from the list and click "Okay":



91. Once the soldier is imported, you can drag it from the resource list into the main window to look at it:

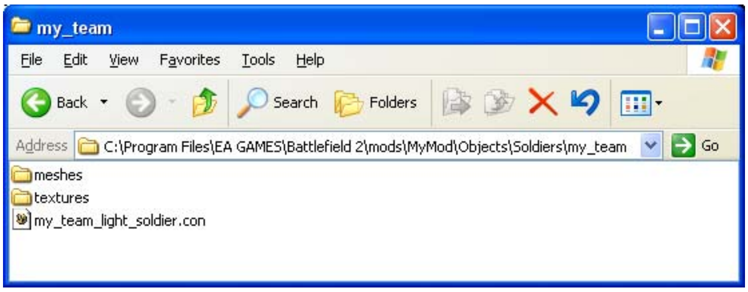


Note: When you drag your soldier to the main window, it will first show the first-person geometry, so you will only see the hands. To see the body, click on the "geom_1" entry in the " G eometry" section of the resource bar as shown above.

Setting up the .con Files.

If you've experimented by switching to the Animation editor, you've found that you can't actually see your new soldier in it c do anything with it yet. This is because we need to do a bit of .con file editing that the editor doesn't do. The reason we imported into the editor first is so that the editor will set up the appropriate directory structure and create some of the .con files we need to edit. We'll finish the rest of this now.

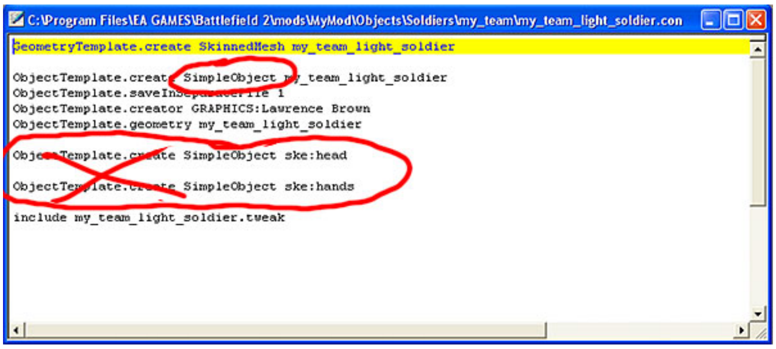
92. Quit the editor. After we edit the .con files, it's best to restart the editor so that it reloads everything correctly.
93. You should now have a folder with the path "…\mods\MyMod\Objects\Soldiers\my_team\". Inside this folder, open the "my_team_light_soldier.con" with your text editor of choice:



C:\Program Files\EA GAMES\Battlefield 2\mods\MyMod\Objects\Soldiers\my_team

meshes
textures
my_team_light_soldier.con

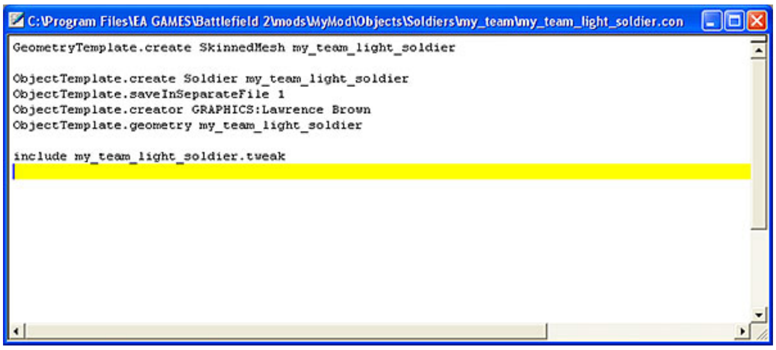
94. Near the top of the .con file will be a line that starts with "ObjectTemplate.create SimpleObject…" Change "SimpleObje to "Soldier".
95. Near the bottom are two more lines starting with "…create SimpleObject" and ending with "…ske:head" and "…ske:hanc Delete these two lines:



C:\Program Files\EA GAMES\Battlefield 2\mods\MyMod\Objects\Soldiers\my_team\my_team_light_soldier.con

```
GeometryTemplate.create SkinnedMesh my_team_light_soldier

ObjectTemplate.create SimpleObject my_team_light_soldier
ObjectTemplate.saveInSeparateFile 1
ObjectTemplate.creator GRAPHICS:Lawrence Brown
ObjectTemplate.geometry my_team_light_soldier

ObjectTemplate.create SimpleObject ske:head

ObjectTemplate.create SimpleObject ske:hands

include my_team_light_soldier.tweak
```

96. Save the file. It should now look like this:

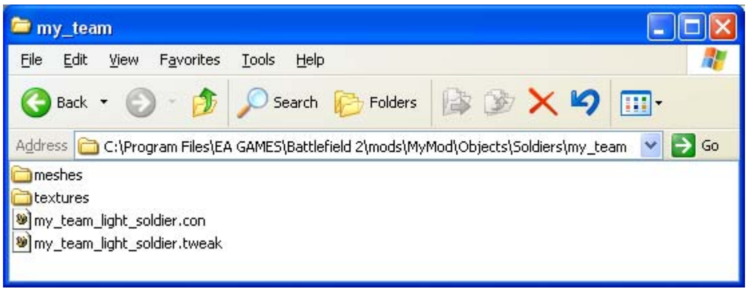


C:\Program Files\EA GAMES\Battlefield 2\mods\MyMod\Objects\Soldiers\my_team\my_team_light_soldier.con

```
GeometryTemplate.create SkinnedMesh my_team_light_soldier

ObjectTemplate.create Soldier my_team_light_soldier
ObjectTemplate.saveInSeparateFile 1
ObjectTemplate.creator GRAPHICS:Lawrence Brown
ObjectTemplate.geometry my_team_light_soldier

include my_team_light_soldier.tweak
```

We have to edit this file manually because the editor doesn't recognize the "Soldier" classification, so it just treats the Maya like a vehicle. Next we have to create the Tweak file.

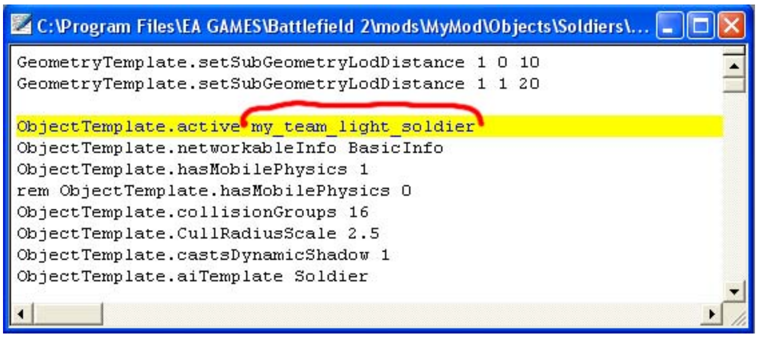
97. In the tutorial files folder that came with this tutorial, find the "us_light_soldier.tweak" file and copy it to the same folc as the "my_team_light_soldier.con" file. (You can also get this file by unzipping the "Objects_client.zip" folder.)
98. Change the name of the file so that it matches the .con file we just worked with. In this case it would be "my_team_light_soldier.tweak":



C:\Program Files\EA GAMES\Battlefield 2\mods\MyMod\Objects\Soldiers\my_team

meshes
textures
my_team_light_soldier.con
my_team_light_soldier.tweak

99. Open the file with your text editor.
100. Near the top will be a line that reads "ObjectTemplate.active us_light_soldier". Change the name of the team to match what's in the .con file. Again, in our case it would be "ObjectTemplate.active my_team_light_soldier":



C:\Program Files\EA GAMES\Battlefield 2\mods\MyMod\Objects\Soldiers\...

```
GeometryTemplate.setSubGeometryLodDistance 1 0 10
GeometryTemplate.setSubGeometryLodDistance 1 1 20

ObjectTemplate.active my_team_light_soldier
ObjectTemplate.networkableInfo BasicInfo
ObjectTemplate.hasMobilePhysics 1
rem ObjectTemplate.hasMobilePhysics 0
ObjectTemplate.collisionGroups 16
ObjectTemplate.CullRadiusScale 2.5
ObjectTemplate.castsDynamicShadow 1
ObjectTemplate.aiTemplate Soldier
```

101. Save the file and close it.

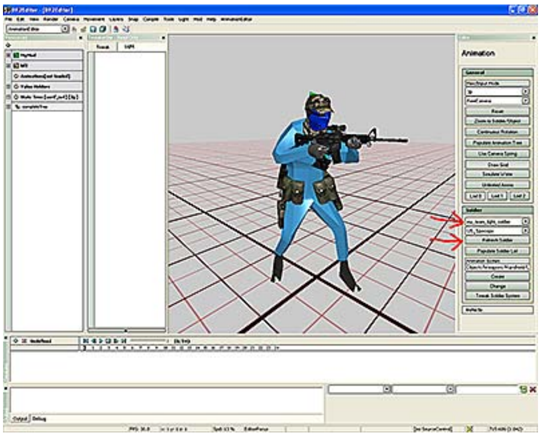
Remember: If you have copied this file from the unzipped Objects zip archive, it will be Read-Only until you change it in it's properties box. I've already done this to the one included with the tutorial files.

Viewing Your Soldier in the Animation Editor

We are now finally ready to view our new soldier in action in the editor.

102. Launch the editor again (selecting the correct mod at start-up) and switch to Animation.

103. On the right side, in the section labeled "Soldier", click on the drop-down list and select "my_team_light_soldier".
104. To load your soldier into the view window, you now have to click on the "Refresh Soldier" button:

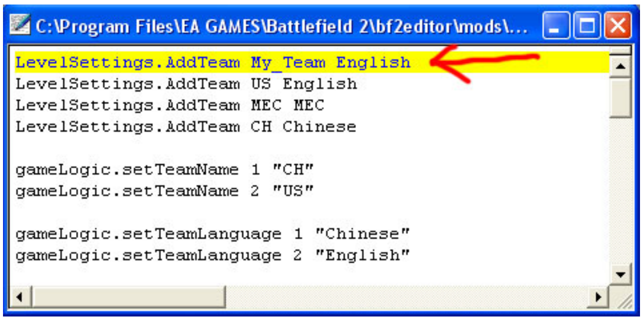


You should now be able to use all the normal animation window controls to view your new soldier in action with various kits animations.

G etting Your New Team Into Your Maps

There are several more things we have to do before we can actually place the team in a map and play. The first is to edit on of the editor files.

105. Quit the editor again. We will need to restart once we finish so that the editor loads the updated files correctly.
106. Locate the file "...\Battlefield 2\bf2editor\mods\MyMod\LevelEditor\LevelSettings.con" and open it.
107. Insert a line at the very top with the text "LevelSettings.Add Team My_ Team English". (You can copy and paste an existing line and just change the team name):

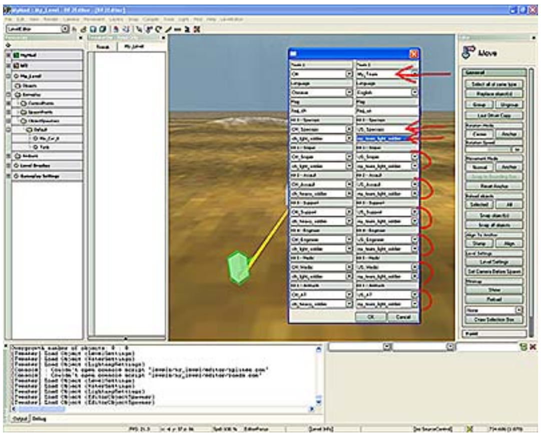


C:\Program Files\EA GAMES\Battlefield 2\bf2editor\mods\...

```
LevelSettings.AddTeam My_Team English
LevelSettings.AddTeam US English
LevelSettings.AddTeam MEC MEC
LevelSettings.AddTeam CH Chinese

gameLogic.setTeamName 1 "CH"
gameLogic.setTeamName 2 "US"

gameLogic.setTeamLanguage 1 "Chinese"
gameLogic.setTeamLanguage 2 "English"
```

Note: Do not change any of the other text in this file. It will currently be incorrect, but you can ignore this because the edito changes this each time you work on a level.

108. Save this file and launch the editor yet again.
109. Switch to the Level editor and load up the level you want to insert your new team into.
110. On the right side of the editor, click on the "Level Settings" button.
111. Click on the drop-down box directly under Team 1 or Team 2 and select "My_ Team ". Notice that the fields below char
112. Manually select what you want in each slot below the team you changed. If you leave any blank, the game may crash. For each of the kits, you will have to use a kit from an existing team (US_Specops, etc.). In the soldier class listing un each of the kits, for now just use "my_team_light_soldier". Later if you make a heavy soldier, you can switch these wh appropriate:



113. Click "Okay" to apply the changes.
114. Add control points, spawn points, and whatever else you want as usual and save your map.

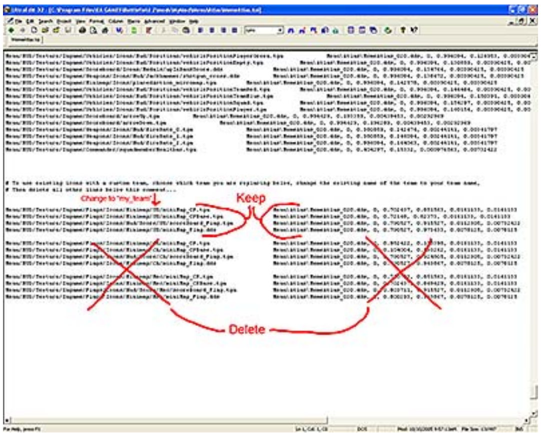
Editing the MemeAtlas.tai File

Because of the way the game is structured, when you add a new team, the game automatically looks for certain minimap icc associated with the team. This is good in that you don't have to do a lot of custom coding to get your new icons in. It's not s good in that if you don't account for this now, the game will crash.

There are two ways to fix this: we can create new custom icons, create a new texture atlas, and edit the MemeAtlas.tai file t find these new icons, or we can just edit the .tai file to use existing icons with the new team. We are going to use the seconc option in this case because it will get us up and running faster.

One limitation of the game is that it will only recognize one MemeAtlas.tai file, so we have to edit the existing file and then make sure the game loads our custom version first.

115. Close the editor if you haven't already. As usual, this is to be sure that once we have finished editing the files, the edit loads the updated versions correctly.
116. Find the "MemeAtlas.tai" file that was included with this tutorial and copy it to "…\mods\MyMod\Menu\Atlas\". (If the "Atlas" folder doesn't exist inside the "Menu" folder, you will have to create it.)
117. Once you have copied this file over, open it with a text editor and scroll to the very bottom to find this section:

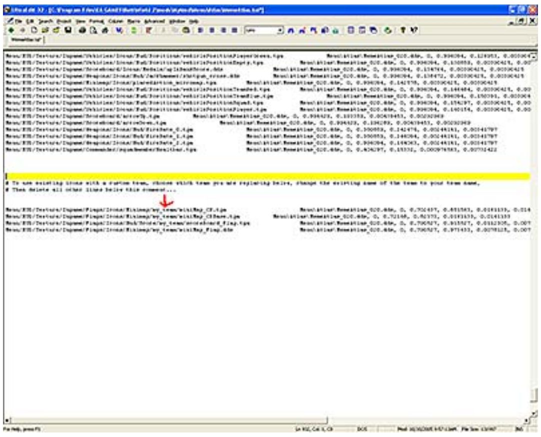


A texture atlas is simply one huge image into which a whole bunch of smaller images have been placed. Texture atlases are used for optimizing the game. It's a lot faster for the video card to load one big image than a whole bunch of tiny ones.

A .tai file is just a text file that tells the game where inside the huge texture atlas each original tiny image was placed. ".tai" stands for "Texture Atlas Index".

In the .tai file you'll notice two columns: the one on the left lists the image that was added to the texture atlas, and the one the right gives the coordinates and size of the image inside the texture atlas.

118. In the section highlighted above, find the group, either US, Chinese, or MEC, that you want to replace with your team. (Remember, this only affects the flag icons, nothing else.)
119. In this tutorial, we are going to replace the US side, so in each line where it says "US", replace the text with "my_tean You have to do this in each of the four lines of the group. Note that I have this highlighted in the screenshot above.
120. Once you are done replacing the text, delete the lines in this section for the two teams you are not using. If you don't, game may crash. You should now have this:

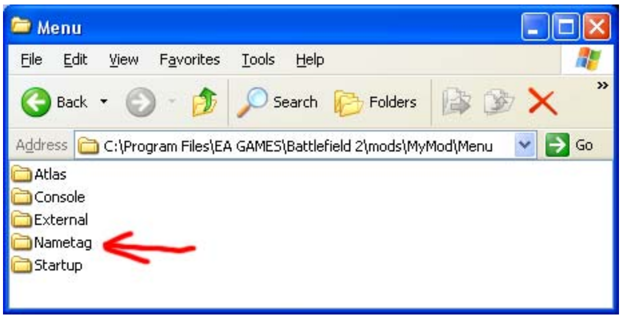


121. Save and close the file.

Placing the Nametag Folder

There are a few icons we have to copy manually to the Menu folder because the .tai file doesn't look for them.

122. Locate the folder labeled "Nametag" that is included with the tutorial files.
123. Copy this to the "…\MyMod\Menu\" folder in your mod directory:

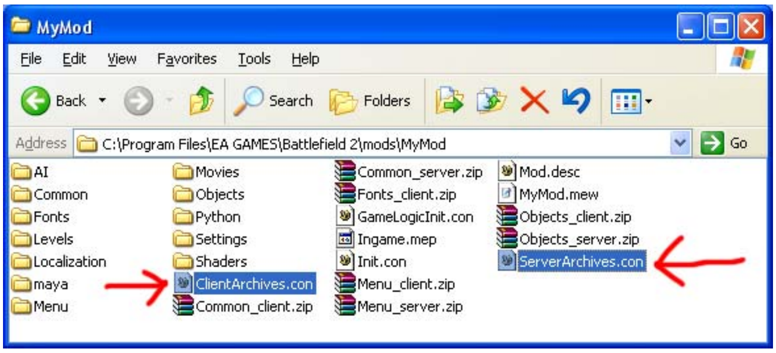

Menu
File Edit View Favorites Tools Help
Back Search Folders
Address C:\Program Files\EA GAMES\Battlefield 2\mods\MyMod\Menu Go
Atlas
Console
External
Nametag
Startup

Note: the contents of your Menu folder may or may not include all the folders seen here. Some are empty but may have bee created if you have previously used the packager.

Correcting the .con File Loading Order

As mentioned above, the game only permits one MemeAtlas.tai file, so if it finds the one in retail BF2 first, it will ignore the changes we made in the mod. We will correct this now by telling the game which order to load things in.

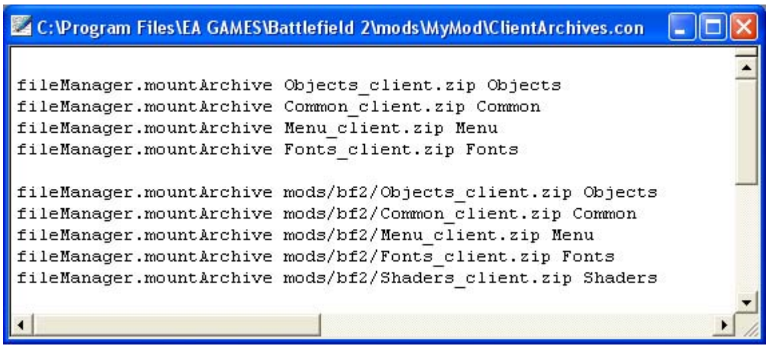
124. In the root folder of your mod, find the two files we need to edit: “ClientArchives.con” and “ServerArchives.con”:



125. Open “ClientArchives.con” in your text editor.

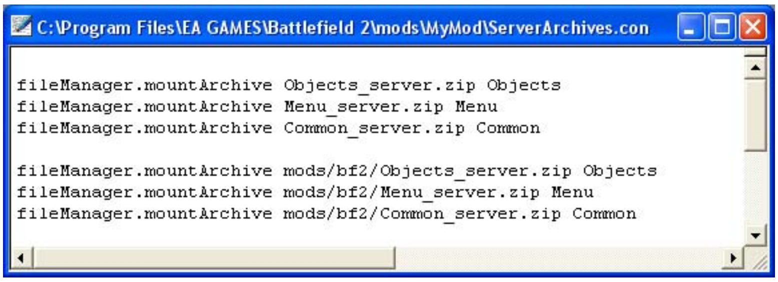
In this .con file, you will see two sets of text. They all start with “fileManager.Archive”. The first set then has a path that sta with “mods/bf2/” and ends with “Objects_client.zip Objects” and so on. The second set doesn’t have a path. All we need to do change the order.

126. Rearrange the lines so that the set without the path to bf2 comes first. When you’re done, your file should look like thi


C:\Program Files\EA GAMES\Battlefield 2\mods\MyMod\ClientArchives.con

```
fileManager.mountArchive Objects_client.zip Objects
fileManager.mountArchive Common_client.zip Common
fileManager.mountArchive Menu_client.zip Menu
fileManager.mountArchive Fonts_client.zip Fonts

fileManager.mountArchive mods/bf2/Objects_client.zip Objects
fileManager.mountArchive mods/bf2/Common_client.zip Common
fileManager.mountArchive mods/bf2/Menu_client.zip Menu
fileManager.mountArchive mods/bf2/Fonts_client.zip Fonts
fileManager.mountArchive mods/bf2/Shaders_client.zip Shaders
```

127. Save the file and do the same to ServerArchives.con. When you’re done, it should look like this:


C:\Program Files\EA GAMES\Battlefield 2\mods\MyMod\ServerArchives.con

```
fileManager.mountArchive Objects_server.zip Objects
fileManager.mountArchive Menu_server.zip Menu
fileManager.mountArchive Common_server.zip Common

fileManager.mountArchive mods/bf2/Objects_server.zip Objects
fileManager.mountArchive mods/bf2/Menu_server.zip Menu
fileManager.mountArchive mods/bf2/Common_server.zip Common
```

128. Make sure this file is also saved and close your text editor. The game will now look for the mod files first and then loa anything additional it needs from the retail BF2 archives. This way our custom MemeAtlas.tai file will be loaded insteac the original.

Packaging for Testing

Because of the particular files we have added, we have to package the mod before the changes will show up. This is a good habit to get into anyway.

129. Launch the editor and choose "MyMod" or whichever mod you are working with from the start-up screen.
130. Switch to the Level Editor and load up the map that you added the new team to.
131. With the level loaded, select "Mod>Package Mod" from the main menu.
132. In the options window that pops up, make sure that both "Package this level:…" and "Package the active MOD:…" are checked and click "OK".

This makes sure that all changes we've made to both the level and the mod get added to the zip files.

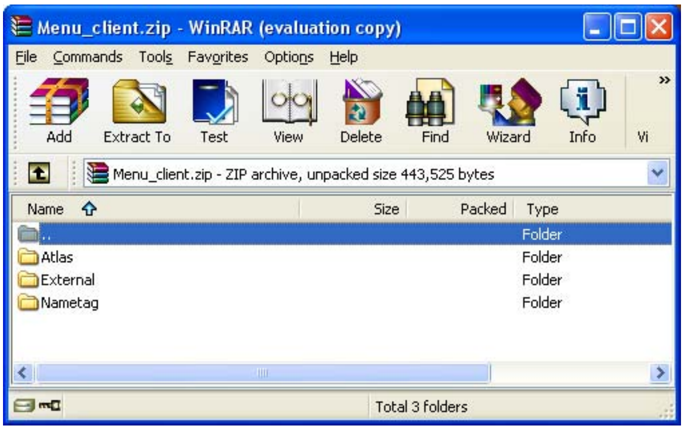
Every time you want to test, you should get in the habit of doing a "Save All" from the file menu and then packaging. This is best way to be sure you are testing the most recent changes.

Correcting File Placement After Packaging

We have now done everything we need to get our new team in. Unfortunately, the packager puts the MemeAtlas.tai file in th wrong place. We must correct this now.

133. Close the editor one last time. We are done with it now.
134. In your root mod folder, find the "Menu_client.zip" file and open it with with an archiving program such as WinRar.

Note: we are not extracting the file, just opening the archive so we can edit it. Here I'm using WinRar, but you can use any archiver that works for you:

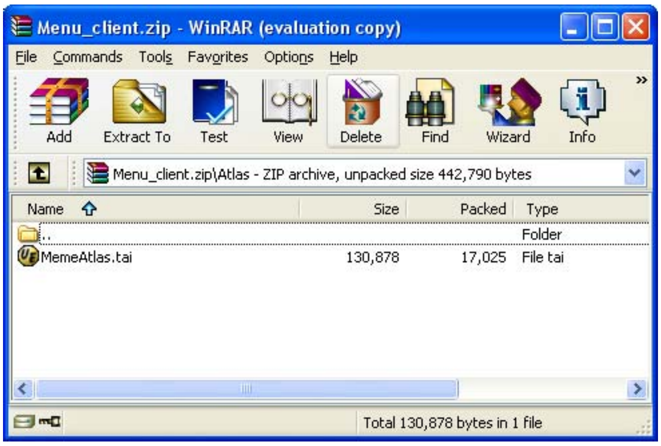

Menu_client.zip - WinRAR (evaluation copy)
File Commands Tools Favorites Options Help
Add
Extract To
Test
View
Delete
Find
Wizard
Info
Vi

Menu_client.zip - ZIP archive, unpacked size 443,525 bytes

Name	Size	Packed	Type
..			Folder
Atlas			Folder
External			Folder
Nametag			Folder

Total 3 folders

135. open the "Atlas" folder so that you can see the "MemeAtlas.tai" file:

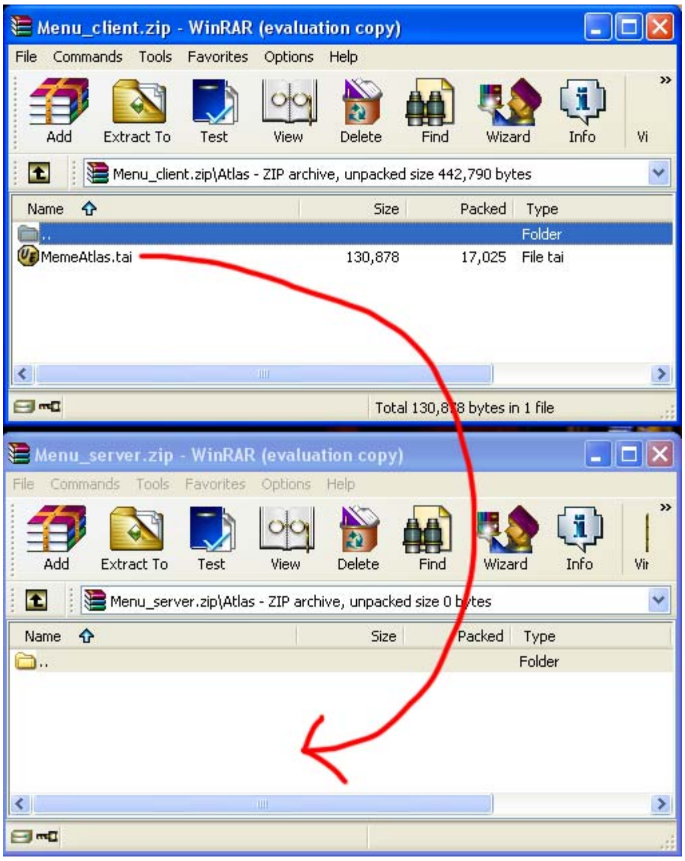

Menu_client.zip - WinRAR (evaluation copy)
File Commands Tools Favorites Options Help
Add
Extract To
Test
View
Delete
Find
Wizard
Info

Menu_client.zip\Atlas - ZIP archive, unpacked size 442,790 bytes

Name	Size	Packed	Type
..			Folder
MemeAtlas.tai	130,878	17,025	File tai

Total 130,878 bytes in 1 file

136. Leave this open and with a second instance of your archiver, open the "Menu_server.zip" file. Depending on what else have done in your mod, this may or may not be completely empty.
137. In the "Menu_server.zip" file, you will need to create an "Atlas" folder if one doesn't already exist.
138. With both the "Menu_client.zip" and "Menu_server.zip" archives open, copy "MemeAtlas.tai" from the "Menu_client\Atla directory to the "Menu_server\Atlas" directory of the server zip:



Menu_client.zip - WinRAR (evaluation copy)

Menu_client.zip\Atlas - ZIP archive, unpacked size 442,790 bytes

Name	Size	Packed	Type
..			Folder
MemeAtlas.tai	130,878	17,025	File tai

Total 130,878 bytes in 1 file

Menu_server.zip - WinRAR (evaluation copy)

Menu_server.zip\Atlas - ZIP archive, unpacked size 0 bytes

Name	Size	Packed	Type
..			Folder

Note: This method works with WinRar. If you are using a different archiver, you may need to use a different method. Just be sure that the path is "Menu_server.zip\Atlas\MemeAtlas.tai". Placing this file anywhere else may cause the game to crash.

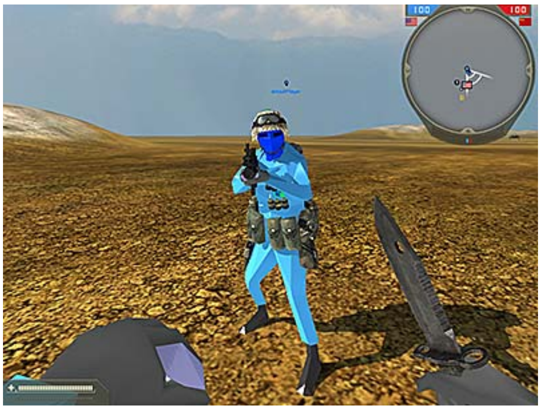
Also note: If the only file in the "Menu_client\Atlas" directory is MemeAtlas.tai, as it is here, you can just copy the "Atlas" fo to the Menu_server zip archive instead of creating a new "Atlas" folder inside it.

139. After you have copied the "MemeAtlas.tai" file to the proper place in the server zip, delete it from the client zip. If you leave it in both places, it may cause problems. (Deleting the "MemeAtlas.tai" file may also delete the "Atlas" folder if i empty. It's okay either way as long as the "MemeAtlas.tai" file is deleted from the client zip.)

Note: Unfortunately, until the packager is fixed, you will have to move this file using the above method every time you repackage the game or the game will crash. (You do not have to do this if you only repackage a level. This applies only whe you repackage the game itself, such as when you add new content.)

Close these archives. You are now ready to test locally.

Important! The files are now ready for you to test by yourself on a local server. If you want to test with others or over the internet, there are some additional changes you need to make to the packaged files. Please see the "Internet Mod Testing Procedure" tutorial for details.



My_ Team in-game…

Congratulations. You have just inserted your first custom team into the game.