

**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****Battlefield 2 Modding Tutorial 4 - The Bunker - 3ds MAX version**

by [EA]Lawrence Brown

[Download 3ds max files here](#)

Now that we've created our basic mod, it's time to actually start making the assets to put in it.

We're going to start out with a simple building: the bunker. This will be an all-purpose building, the base providing cover from enemy fire with window slots allowing soldiers to shoot back, and also a small tower providing a better view of the battlefield a vantage point for laying cover fire and sniping, and also a potential platform for mounting static weapons such as machine guns and TOW rockets.

**Set Up:**

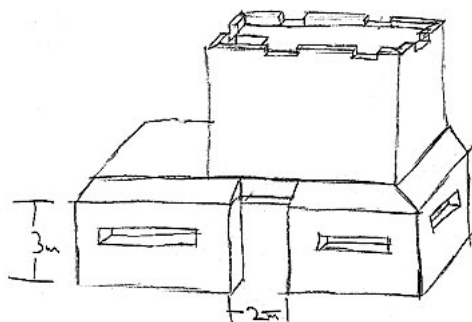
Before we begin, do the following:

1. Open up the "Tutorial Files" folder that came with this document and follow the instructions in the ReadMe to properly place all tutorial files.

This tutorial also requires that you have installed all the Max tools that shipped with the editor.

**Modeling:**

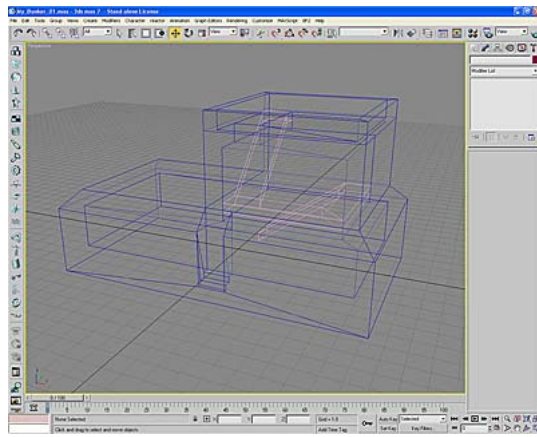
We're going to start with a simple sketch of the building. This could come from a designer on the team, or it may be something you draw up yourself. The point is, even if you are just modeling for yourself, it's good to at least draw out a basic form on paper to help visualize what you're trying to do. Here's our sketch for the bunker:



Now that we've got a blueprint of sorts, we can begin modeling.

The first step is to create the main visible LOD (Level of Detail)

1. Launch Max and open up My\_Bunker\_01.max. Here I've roughed out the shape of the bunker, creating the interior and exterior and a system of ramps to get to the top. Notice that I haven't put in details like the windows or the crenellation at the top of the tower (the notches in the sides of the walls). At this point we just want the basic shape so that we can export and test the overall design. Since we will probably have to make changes, there is no sense wasting time on details that would have to be reworked later.



2. Select the two pieces and combine them into one using the Attach function in the Edit Geometry rollout. Use the default when the "Attach Options" window comes up.
3. Rename this piece "My\_Bunker".

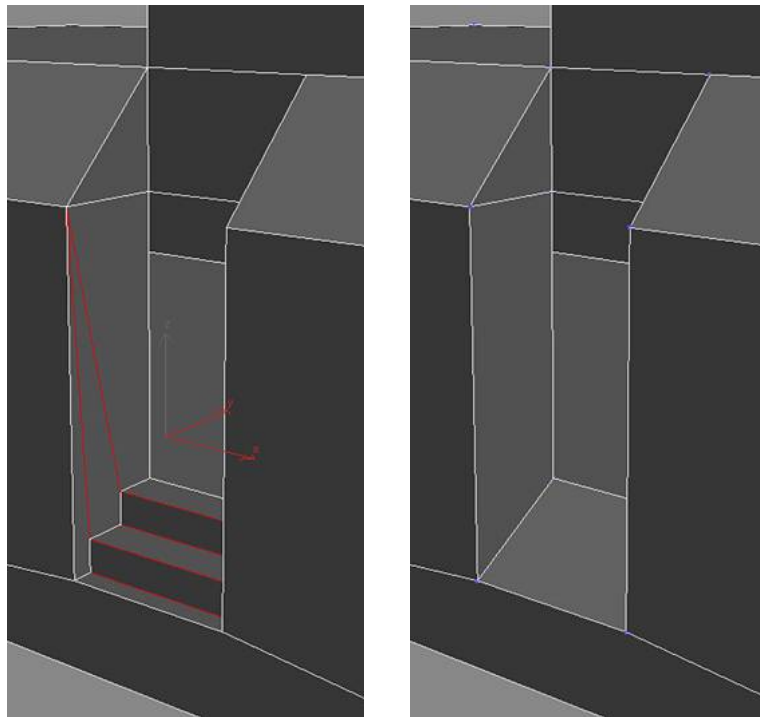
#### Col0:

The next step is to create the collision meshes. First is the projectile collision mesh, Col0.

4. Clone My\_Bunker using the copy option and name it "Col0".

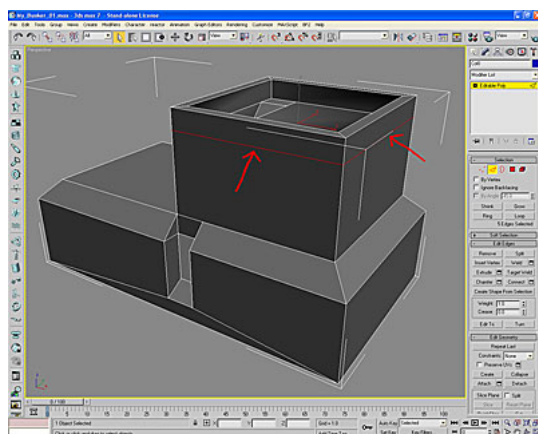
Since this is the projectile collision mesh, we want to optimize it by deleting any small details that we don't need. If there are things like antennas or other small items attached to the building, we would delete these. We also want to simplify the mesh where possible. The less polygons the game has to test against collisions, the faster the server will run. This building is currently so simple that there isn't much to optimize. Even so, we should still do the following.

5. Hide My\_Bunker so that we can clearly see Col0 and zoom in on the stairs. Remove all excess edges until the stairs become a ramp like the following:

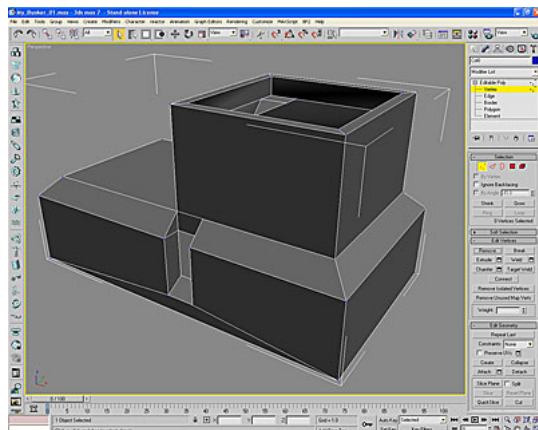


Note: make sure you remove any extra vertices that may have been left behind.

6. Now select the edges highlighted in red in the following screenshot and remove. The edges go all the way around the tower, so make sure you select all four. Once again, don't forget to remove the extra vertices also:



When you're done, the Bunker should look like this:



### Col1:

We've completed the optimization of Col0. The next step is to create the soldier collision mesh, Col1.

7. Clone Col0 as a copy like you did before. Rename the new object "Col1" if it doesn't do so automatically. (Note the spelling: "Col1", not "Col01"!)

Normally we would do further optimization on this mesh, but there is really nothing to remove since it's so simple. More about this later.

### Col2:

The last collision mesh to make is the vehicle collision mesh, Col2.

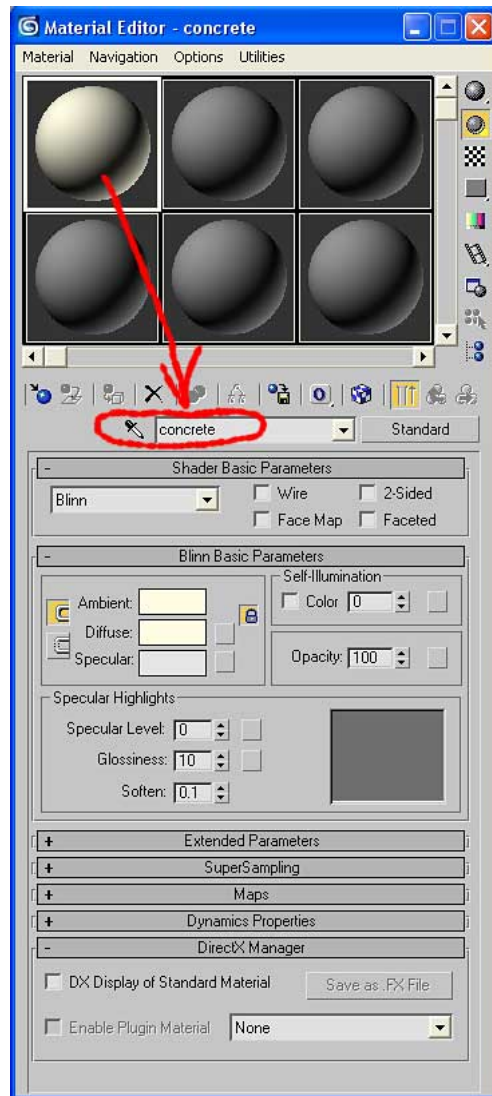
8. Following the same steps as earlier, clone Col1 and rename it "Col2" if it doesn't do so automatically.

This mesh would also be optimized, but we're not going to waste time on it now since it's just for preliminary testing. Keep in mind that we will be going back and adding more detail to the visible LOD once we are satisfied that the overall design work. At that time we will also be deleting the current collision meshes and creating them again from the updated LOD. Reworking objects is just part of the modding process, so it's best to get used to it now.

### Texturing:

The next step is to texture the pieces we created. You may either continue your work or open "My\_Bunker\_02.max" for this section.

9. Open the Material Editor and select the first shader slot.
10. Change the name to "concrete":



You can change the color of the shader to anything you like. The color is just to make it easy for you to identify. All the importer is concerned about is the name. Here I've given it a light yellow color so I can identify it easier. You may also wish to check the box next to "Wire" so that only the wireframe shows. This isn't strictly necessary, it's just an easy way to display and identify collision meshes.

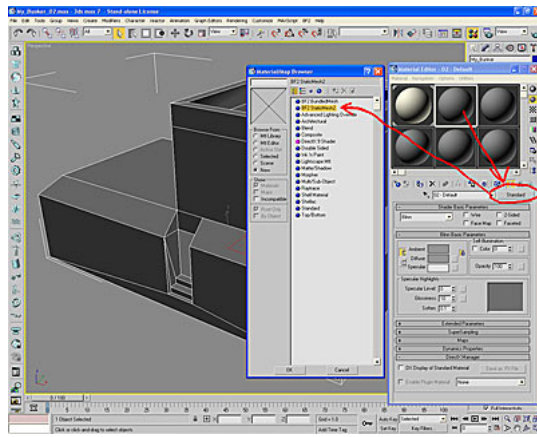
11. Assign the concrete shader we just made to the three collision meshes.

Concrete is the name of one of the materials in the game, so what will happen is that the importer will read this name and apply the material to the collision meshes. This is why it's so important that it is spelled right. For a complete list of materials consult the Material Editor section of the BF2 Editor.

### The Visible Mesh:

Now we have to texture the visible mesh. Since this is just a test and we know we are going to add more detail later, we are just going to apply a simple placeholder shader for now.

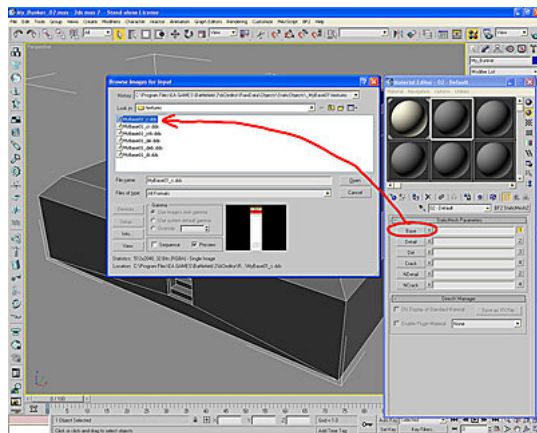
12. Hide everything but My\_Bunker so that it is easier to work on.
13. In the Material Editor, select the second shader slot. Locate the button next to the material name that is labeled "Standard" and click it.
14. In the window that pops up, select "BF2 StaticMesh2" and click "Okay":



15. Make sure that My\_Bunker is selected and click the "Assign Material to Selection" button.

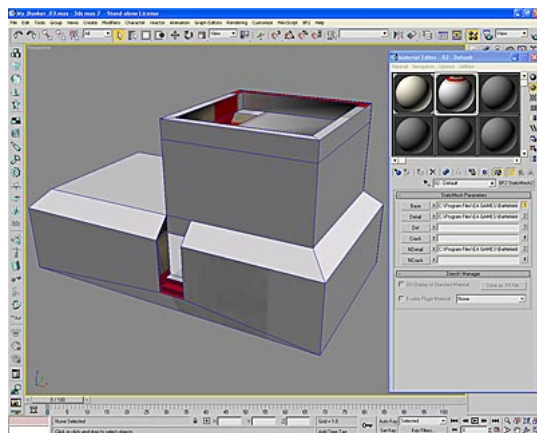
What we've done is to create a cgFX shader and assign it to the selected object. We now have to assign textures to our shader

16. In the "StaticMesh Parameters" section of the Material Editor, click on the button labeled "Base". In the window that pop up, select "C:\Program Files\EA Games\Battlefield 2\bf2editor\RawData\Objects\StaticObjects\\_MyBase01\textures\MyBase01\_c.dds":

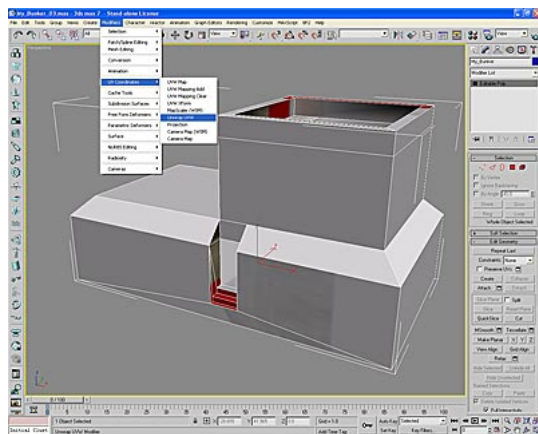


17. Do the same for "Detail", choosing "MyBase01\_de.dds" from the same folder.  
18. Do this a third time for "NDetail", choosing "MyBase01\_deb.dds". ("\_de" stands for detail and "\_deb" stands for detail bumpmap.)

Make sure that you've also clicked on "Show Map in Viewport" in the Material Editor. If you are in "Smooth + Highlights" mode your bunker should look something like the following screenshot. Don't worry if the texture placement is different because we haven't set up the UV's yet:

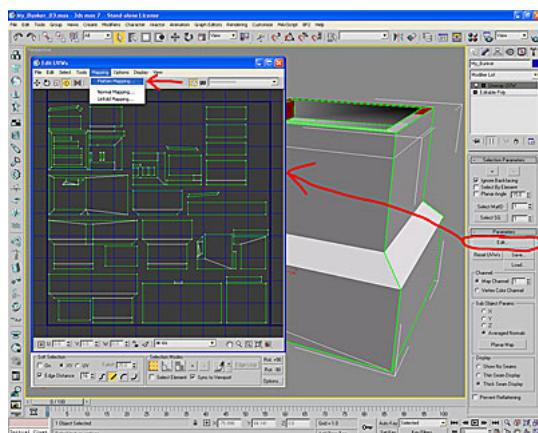


19. The editor and the BF2 engine don't like UV's with zero surface area, so we're going to lay out the UV's the quick way. With My\_Bunker selected, select "Modifiers>UV Coordinates>Unwrap UVW":



20. In the modifier list, make sure that "Unwrap UVW" is selected in the stack.
21. In the Parameters roll out below, click on the button labeled "Edit...". This will open up the "Edit UVW's" window.
22. In the "Edit UVW's" window, select "Mapping>Flatten Mapping". An options window will open up. Leave everything at its defaults and click "OK".

Your UV's should now look similar to this. Don't worry if yours looks a little different. The important thing is that we've given each polygon some surface area and none of the UV's overlap:



Notice that I've also highlighted the buttons from the previous few steps.

23. Unhide the collision meshes and do the same thing to each of them. Even though the collision meshes aren't seen in the game, the engine doesn't deal well with polygons with zero surface area when calculating collisions either. The above steps will prevent those problems. It doesn't matter what the actual coordinates of the UV's are, only that they have some surface area.

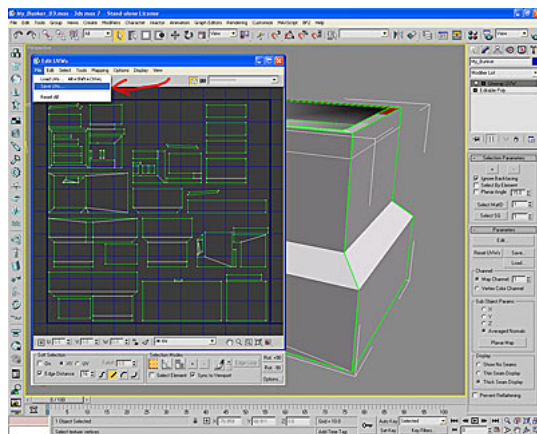
### Creating the Second UV Set:

Now that we've laid out the basic UV's on all the pieces, the last thing we have to do texture-wise is create a second set for visible mesh. More on this in a later tutorial. For now you just have to know that the Base texture in the shader needs one UV set and the detail texture needs a second one.

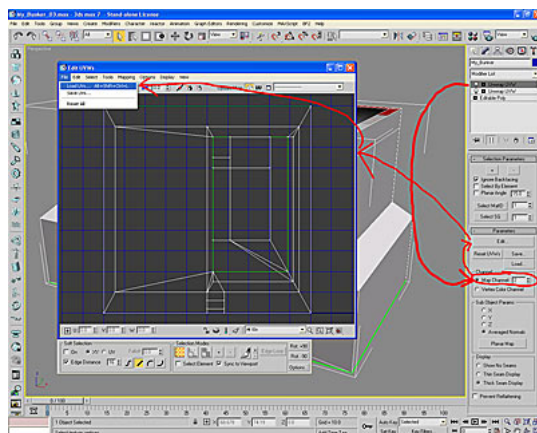
It's important to note that only the visible mesh needs the second UV set. The collision meshes should only have one. If the collision meshes have more than one UV set, it could crash the game!

24. Hide the collision meshes again and reselect My\_Bunker.
25. In the modifier list, make sure that "Unwrap UVW" is selected in the stack.
26. In the parameters rollout, click the "Edit..." button to open up the "Edit UVWs" window again.
27. Select ">File>Save UVs..." in this window:





28. In the window that pops up, enter any name you like (I used "My\_Bunker\_UVs"). This is just a temporary file. Just make sure you look at the path you are saving to in order to make sure you can find it again.
29. Once you have saved your UV set, select "Modifiers>UV Coordinates>Unwrap UVW" to create a second Unwrap UVW in the Modifier List.
30. In the parameters rollout, make sure that "Map Channel is set to 2. This corresponds to the channel used by the "Detail" texture in the Material Editor.
31. Click on the "Edit..." button again to open the "Edit UVWs" window once more. Notice how the UV's don't match the UV's we laid out before.
32. Select File>Load UV's... in the "Edit UVWs" window and select the file you previously saved. In the next screenshot I've highlighted the previous steps:

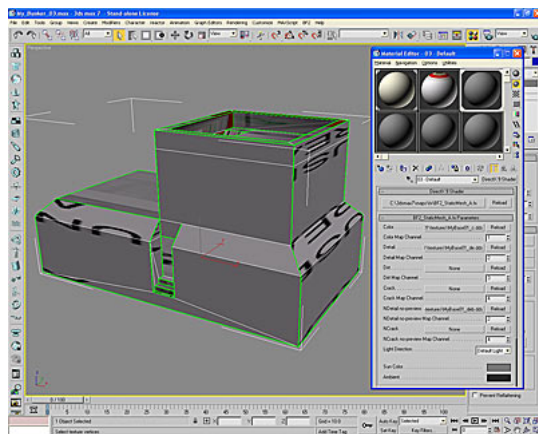


What we've done is to create the two necessary UV sets for the cgfx shader we are using. If you were also using a dirt layer your shader, you would need a third, and if you added the crack layer, then you would need a fourth.

The reason we used the Unwrap UVW method is because this is the only modifier that lets you edit your UV's in the "Edit UV" window. All other methods make you work with your object as a whole instead of on a per-face basis.

Note that you will not be able to see all of your layers at once with the current shader. Right now we are just interested in setting up the staticMesh object for export, so we aren't really concerned with this, but if you want to see what your multilayered shader looks like, do the following:

- In the Material Editor, click on any unused shader slot.
- Click on the button labeled "Standard" like before, and this time choose "DirectX 9 Shader".
- In the "DirectX 9 Shader" rollout section, there will be a button labeled with a path like "C:\3dsmax7\maps\fx\Default.1". Click on this button and choose "BF2\_StaticMesh\_A.fx" in this window that pops up. You should now see a list of texture channels similar to before.
- Click the button labeled "None" next to "Color" and select "MyBase01\_c.dds" the same as you did before.
- Do the same for "Detail", choosing "MyBase01\_de.dds".
- Finally, do this again for "NDetail no-preview", choosing "MyBase01\_deb.dds".
- Assign this new shader to the "My\_Bunker" mesh. You should now see something similar to the following:



If you want, you may now experiment with using the "Edit UVWs" window to move the UV's around on your two Unwrap UVW modifiers and you will see the textures update in your display.

Once you are done, make sure you re-assign the original BF2StaticMesh2 shader that we created or your building won't expo correctly.

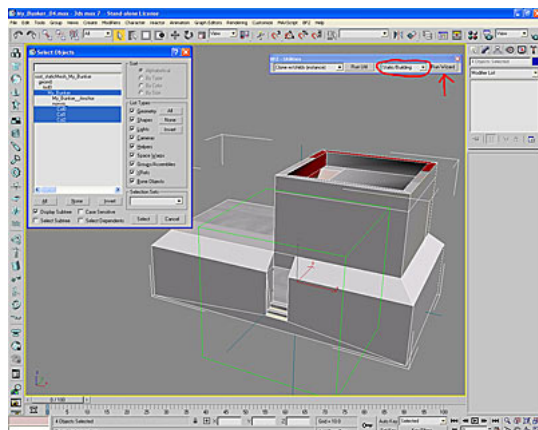
### Running the StaticMesh Wizard:

We've created and textured all the pieces. Now we have to set up the hierarchy before we can export. To do that we'll use one of the wizards.

You may either continue with the scene you've been working on or open "My\_Bunker\_03.mb".

First we have to clean up the pieces so that there is nothing in the modifier stack.

33. Make sure that all pieces are unhidden, then select each piece, and in the Modifier list, Right-click somewhere in the stack and select "Collapse All". If a warning window pops up, click "Yes" to continue.
34. Now that the pieces are clean, select "BF2>BF2 Utilities" from the main menu. This will open up a small window with the same name.
35. There are two drop-down boxes in the window. Make sure the one on the right is set to "Static/Building".
36. Select all three collision meshes and My\_Bunker.
37. Click on the "Run Wizard" button. You should now have something that looks like this:



Notice that I've also opened up the "Select Objects" window and clicked on the "Display Subtree" option so that you can see the hierarchy. Your hierarchy should look the same.

Several new objects have been created. These all have specific names that the exporter needs to read in order to create the proper code for the game. You won't actually see these additional objects in the game. They are just placeholders in the Max file.

### Exporting the Bunker:

We are now ready for export.

38. Save your file if you haven't already.
39. Select "BF2>BF2 Exporter" from the main menu.
40. If this is your first time exporting, you will get an error labeled "Output Directory Invalid". This is because we haven't set the exporter where to put our file yet. Click "Okay" to proceed.
41. A new window will come up labeled "BF2 Tools Setup Wizard". In step 1, click on the "browse" button. Select the "MyM

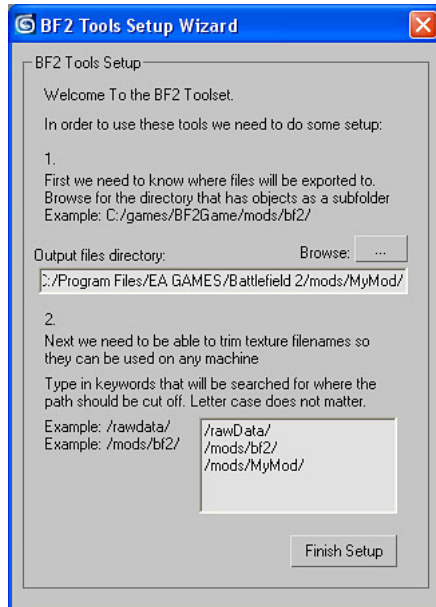


folder and click "Okay".

42. In step 2 it wants you to type in a path. Click in the box with the text.

43. Move the cursor until it is at the very end of the last line of text and hit return. This will begin a new line. Type in "/mods/MyMod/" without the quotes.

44. Click on "Finish Setup" to complete setup:



This will now bring up the normal export window. We have a few more options to set here.

45. The first field is labeled "Objects sub-folder path:". Enter "StaticObjects\\_MyBase01\". You want this to be the path to where your object is going to be in your mod. The path should start with the folder AFTER the "Objects" folder. It can sometimes be easier to open up an explorer window, navigate to where you want the object to be, and copy-and-paste the path in.

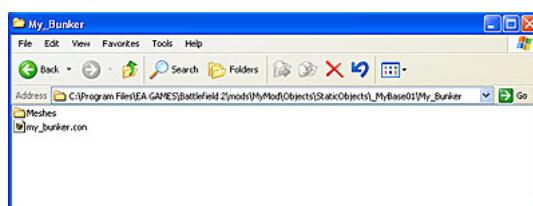
46. The second field is the object name and should normally already be correct. Fix this if it is not. The exporter will also create a folder with this name if it doesn't already exist.

47. Next to "Lightmap Size:", you can leave this set to auto. When you export, the exporter creates a new texture that contains shadow information for your object. This determines the size of that texture. The window should now look like this:



48. Leave all other options at their defaults and click "Export Object".

You have now completed your first staticMesh Export. If all went well, you should now have the directory "C:\Program Files\EA Games\Battlefield 2\mods\MyMod\Objects\StaticObjects\\_MyBase01\My\_Bunker". Inside this folder should be a folder labeled "Meshes" and a file labeled "my\_bunker.con":



In a later tutorial we will import this into a custom level.

If you have any problems or want to compare your version to the finished version, open up "My\_Bunker\_Complete.max". You can use this file as reference for setting up basic staticMesh objects.

You are now ready to move on to "Battlefield 2 Modding Tutorial 5 – Importing the Bunker".