

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**Creating New Kits – Maya and Max**

by [EA]Lawrence Brown

[Download tutorial files](#)

Updated: 11/28/2005

Note: This is considered an advanced tutorial. It covers both Maya and Max workflows. Where differences occur in terminology but are essentially the same operation, I will be defaulting to Maya terms and putting the Max equivalent in Parenthesis. For instance: "Import (merge)..." "Importing" is what you would do in Maya, and "merging" is what you would do in Max. Where there is an actual difference in workflow, I will create separate sections and label each accordingly. Skip over whichever one not applicable to you.

It also may help to read through this entire tutorial first before beginning actual work. The most important steps are at the end where you will skin the kit pieces, set up a hierarchy, and export. The creation steps don't need to be as rigidly followed as I as you end up with the pieces that are needed.

Set-up

- Inside the "Tutorial Files" folder that came with this tutorial, locate the folder labeled "My_ Team ".
- Copy this folder to "RawData\Objects\kits\My_ Team ". (You may or may not have to also create the "kits" folder to make this path correct.)

Note: Do not confuse this folder with the "My_ Team " folder in your soldier directory. They are labeled the same, but are not the same thing.

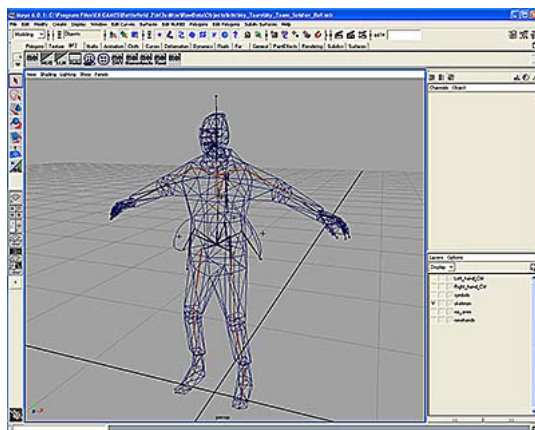
Creating the Geometry

The first step in creating a new set of kits is to create the individual pieces. All pieces should be created at the same time because of the export requirements. This will be explained more later.

1. Start by creating a new scene and importing your soldier mesh as reference. If you are using variations such as Light or Heavy, you can import both if you like and hide whichever one you are not using. If you are following these tutorials in order, you can start by opening up "My_ Team _Soldier_Ref", which can be found in the My_ Team Kits folder along with the other files.

Maya:

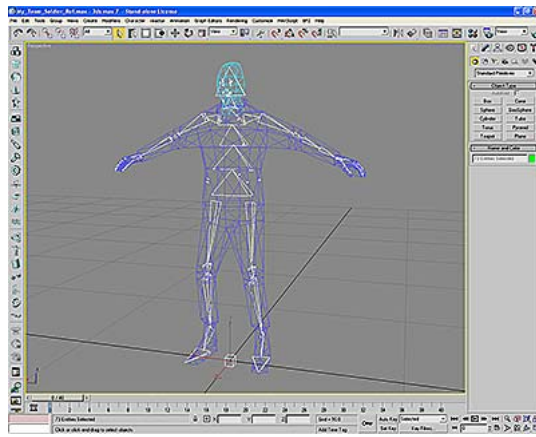
2. Import your 3rd-person soldier skeleton into this scene. Normally this would be the "3p_setup" file found in "RawData\Animation\Setup\3p". Note that we are actually importing the skeleton this time, unlike when setting up a new soldier mesh where we referenced it. You should end up with something that looks like this:



3. Save this file as "My_ Team _Kits". (Note that it is important that "_Kits" be at the end of the name.)

Max:

4. Merge the file "Kits_Skeleton.max", found in the "My_ Team " folder with the other tutorial files that you placed for this lesson. You should end up with something that looks like this:



5. Save this file as "My_ Team _Kits". (Note that it is important that "_Kits" be at the end of the name.)

Note: whatever you use for your soldier reference doesn't need to be textured. It's just there as a guideline for modeling your kits. What's important is that the 3rd-person skeleton that you are using in the game be there.

6. Model and texture all your kit pieces using the following texture notes. Each piece, such as the various packs, helmets, and so on, should be a separate mesh. The names aren't that important, but should be something meaningful so that you can easily pick out the various pieces from a list later.

Note: If a particular piece is going to be used for several kits, you only have to create one for now. We will make the duplicates later.

Don't forget to create the pilot kits and drop kits at this time. Note that none of these pieces need any collision meshes, not even the drop kits. There should just be a single visible mesh for each piece.

Texture Notes

There is no actual restriction on how the textures are set up, but it is strongly recommended that you stick to the following texture sheets and sizes unless you have a specific reason for adding more. Just remember that more textures or bigger textures may affect game performance, so be sure to test if you attempt a different set-up.

The following are the default sizes used in the retail game:

- Kits – 2048x2048

This is the texture sheet used by all of the basic kit pieces of one team. These are the pieces that appear on your player when you choose a class.

- Pilots – 1024x512

This sheet is used by the "pilot" kits that the player gets when they enter a vehicle. The vehicle classes that use these are "_tanker", "_fighter", and "_heli".

- Drop kits – 512x512

This sheet is applied to the drop kits that are left behind after a player dies.

Notes:

- These textures should all be saved as DXT 5 .dds textures with mipmaps, but no alpha. Kits and Pilots each also have a corresponding normal map, but Drop kits do not.
- Additional textures can be created where necessary for special cases. For instance: if you unzip the Objects_client.zip of the retail game and look at the textures for the US team, you will also see a set for specops, the ghilliesuit, and also a woodlands variation.
- It is possible to have an alpha channel with the kit textures. See the ghilliesuit for an example of this.
- The names of all shaders, whether Maya or Max, should end with "_Tangent". This prevents the conversion of the normal maps to object space when importing/exporting, which is what we want in this case.

If you want to use the kit pieces that I created for the tutorial, import (merge) the file "My_ Team _Kit_Pieces" into your scene. Note: you will then already have the duplicates necessary for the next step.

Creating the duplicates

Eventually we will be placing all the pieces for an individual kit under 1 node of a hierarchy the same way we do with everything else, so we now need to make any necessary duplicate pieces. The best way to do this and stay organized is to do something like the following:

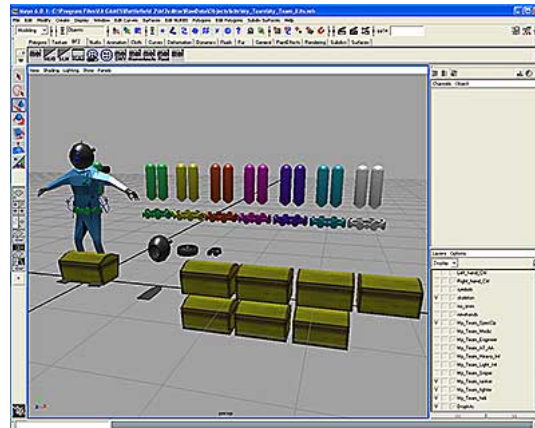
7. Select each mesh that you just created and move it off to the right side. Spread them out so that you can see each individual piece.
8. One at a time, select each mesh piece for the first kit, duplicate (Clone) it, and move the duplicate to the correct place on the soldier reference. (This does not include the drop kits)

9. Select all the pieces that make up this kit and assign them to a new layer (Maya), or create a new set (Max). Rename Layer or Set to match the corresponding kit.

Note: For this operation, there should be a kit for each class, and a "kit" for each "Pilot" kit you are making. For example, if you are creating the kits for "My_Team ", you would have the following sets:

- My_Team _SpecOp
- My_Team _Medic
- My_Team _Engineer
- My_Team _AT_AA
- My_Team _Heavy_Inf
- My_Team _Light_Inf
- My_Team _Sniper
- My_Team _tanker
- My_Team _fighter
- My_Team _heli

When you're done, you should have something that is similar to this:



Note that for simplicity, the kits for my various classes are basically just the same meshes mapped to different color areas of the texture sheet. In this case the drop kits are also all the same. Yours will be different.

Note that you may also wish to assign pieces to various layers (or sets in Max), which you can hide and show individually in order to stay organized. This is purely optional, but will help later once we begin skinning so that you can hide all but the piece you are skinning.

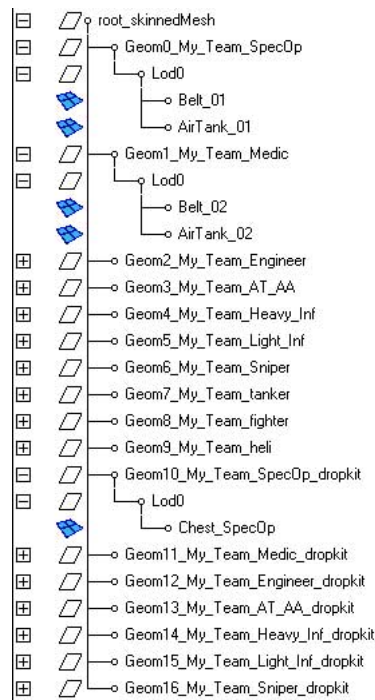
Setting up the Hierarchy

Now that we've created all the pieces, it's time to organize them in the hierarchy. Skip to the section labeled for the modeling package that you are using.

Maya:

10. Create an empty node and rename it "root_skinnedMesh".
11. Create a second empty node and rename it "Geom0_My_Team _SpecOp".
12. Parent the second node under root_skinnedMesh.
13. Create a third empty node and rename it "Lod0".
14. Parent this node under Geom0_My_Team _SpecOp.
15. Select all the mesh pieces that belong to this kit. Make sure that their pivot points have all been moved to the origin and all transforms reset, then parent them to Lod0. (This is where the layers come in handy.)
16. Repeat this for all the other class kits, pilot kits, and drop kits. Note that the drop kits just sit at the origin. (Don't forget to reset their transforms, too.)

When you are done, your hierarchy should look like the following. Notice that I've expanded the first two class kits and one of the drop kits:



The important things to note here are that each set of meshes is grouped under a node named "Lod0", which is in turn grouped under a node that starts with " G eom0...", " G eom1...", and so on. The actual name that comes at the end of those labels isn't that important, though it's best if you label everything correctly to avoid confusion.

Once we get to editing the kit .con files, you will see that the game references the " G eom" number. It is important that they be in order, that the first one be " G eom0", and that no numbers are skipped. You can actually add more if you like, but we'll get into that more later.

Max:

17. Create a point helper at the origin with display set to "Cross" and rename it "root_skinnedmesh_My_ Team _Kits". (Make sure that the name ends in "_Kits")
18. Create a dummy helper and rename it " G eom0_ SpecOp".

Note: you can name the helper with just " G eom0", but it's helpful to add the name of the kit to the end. " G eom" and the number of the kit is what the exporter is actually looking for.

19. Parent the G eom0 node under root_skinnedMesh_My_ Team _Kits.
20. Create a second dummy helper and rename it "Lod0".
21. Parent this dummy under G eom0_ SpecOp.
22. Select all the mesh pieces that belong to this kit. Make sure that their pivot points have all been moved to the origin and all transforms frozen, then parent them to Lod0 (This is where the sets come in handy.)
23. Repeat this for all the other class kits, pilot kits, and drop kits. Note that the drop kits just sit at the origin. (Don't forget to freeze their transforms, too.)
24. At this time, you will also have to rename each of the drop kit pieces (and any other pieces that won't get skinned to a bone) so that they end with "__SimpleObject". For instance: "Sniper_dropkit__SimpleObject". Note the usual double-underscore. If you don't use this label, the export won't work right.
25. Lastly, parent ske:root_skeleton under root_skinnedmesh_My_ Team _Kits.

When you are all done, your hierarchy should look like the following. Click on the image for the full screenshot:



Note that I've labeled the names that were cut off by the schematic view.

Also note that I've only expanded the first class kit and the first drop kit. The important things here are that each set of meshes is grouped under a node named "Lod0", which is in turn grouped under a node that starts with " G eom0...", " G eom1...", and so on. The actual name that comes at the end of those labels isn't that important, though it's best if you label everything correctly to avoid confusion.

Once we get to editing the kit .con files, you will see that the game references the "G_eom" number. It is important that they be in order, that the first one be "G_eom0", and that no numbers are skipped. You can actually add more if you like, but we'll get into that more later.

Skinning the kit pieces

We have all the pieces connected in the hierarchy, but we still need to skin them to the skeleton.

26. Save your work again if you haven't already so that you don't have to repeat the last steps over again should anything wrong.
27. Starting with the first kit, skin the pieces to the appropriate bones.

The skinning process is the same as when creating a new team, so I'm not going to go over the whole thing again. Just remember the following:

- Any single mesh can be skinned to up to 24 bones. It doesn't matter which set or sets the bones belong to. The guillie is a good example of this. It covers most of the body, including the head, but because of the bone limit, only certain bones from the body, head, and hands were included in the skinning of the main piece.
- In many cases, a kit piece may only be attached to one bone, but for larger pieces that need to deform, you can still sit across multiple bones, just like the player skin. Just make sure that a maximum of only 2 bones influence each vertex.
- Drop kits don't get skinned to anything. They just sit at the origin. Just make sure they are connected in their proper places in the hierarchy so that they get exported.
- We are only exporting the kit pieces! Do not skin the body, head, or hand pieces from the soldier mesh that you import as reference! Doing so would cause problems.

28. Once you have skinned all the kit pieces, make sure they all deform correctly by moving the bones around. (Don't forget to return all the bones to their original position. You may want to save the scene, move the bones around for a final check, then reopen the scene to be sure none of them have moved.)
29. When you are satisfied that everything is working correctly, make sure you have unhidden everything and do a final save.

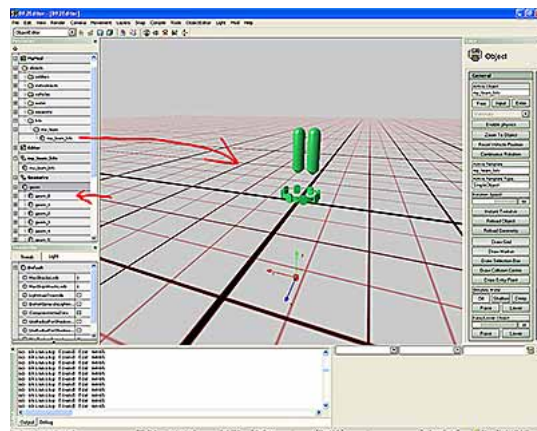
Importing/Exporting the Kits

It is now time to get the kits into the editor. This is where the process will differ most depending on if you are using Maya or Max. We will go through the Maya process first, then Max, so skip to whichever section is appropriate.

Maya:

30. (Optional) With the scene still open, you may wish to delete the body, head, and hands meshes that you imported as reference just to be sure they don't export. Alternatively, just make sure that they are NOT skinned to the skeleton or included in any hierarchy.
31. Once you are satisfied, make sure the scene is saved as "My_Team_Kits" and quit Maya.
32. Open up the editor and select "My_Mod" in the Startup MOD window.
33. Switch to the Object editor and import the "My_Team_Kits.mb" scene you just finished.

Note: In the output window you will see a bunch of lines that read "no skinning found for mesh". This is from the dropkits. As long as the editor doesn't crash, it's safe to ignore it if an error message comes up. If you now drag the kit to the main window you should see something like this:



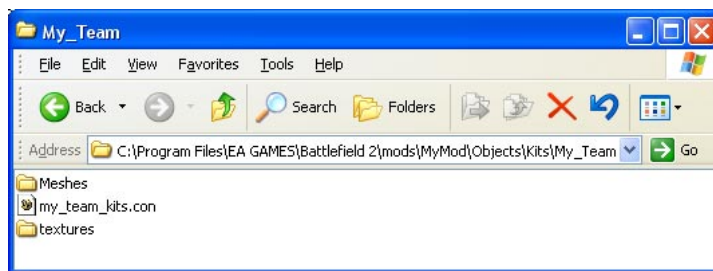
Note that you can expand the geometry section at the bottom of the tweak bar and click on each of the geometries to see the various kits.

Max:

34. (Optional) With the scene still open, you may wish to delete the body, head, and hands meshes that you imported as reference just to be sure they don't export. Alternatively, just make sure that they are NOT skinned to the skeleton or included in any hierarchy.
35. From the main menu, select "BF2>BF2 Exporter".
36. In the "Objects sub-folder path:", enter "/Kits/My_Team /".
37. In the "Object Name:" field, enter "My_Team_Kits".
38. Select the root_skinnedmesh_My_Team point helper in the scene and then click on the "Export Object" button in the BF2 Exporter window.
39. If you get an error saying that at least one vertex was not weighted to any bone, this will be because of something like one of your envelopes not being big enough to cover all vertices of a particular mesh. When the error window comes up

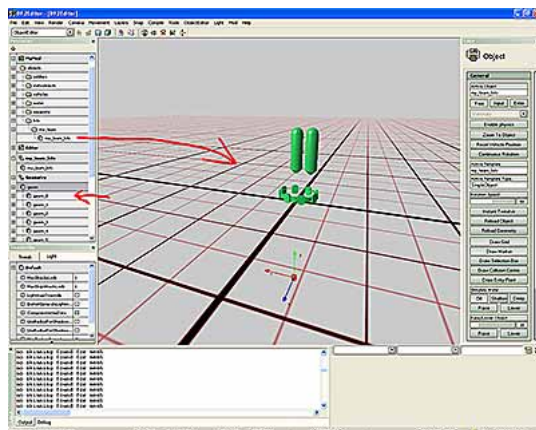
you may be able to get a clue as to which piece is bad by looking at which bone is highlighted in the modifier rollout. You will have to correct these errors before the export will work.

40. You will have to manually copy the "textures" folder from "RawData\Objects\kits\My_Team" to "mods\MyMod\Objects\kits\My_Team" because the Max exporter doesn't move textures. At this point, the folder should look like this:



41. Open up the editor and select "My_Mod" in the Startup MOD window.

42. Switch to the Object editor. You should now see "My_Team_Kits" listed in the tweaker bar. If you drag the new kit to main window, you should see something like this:

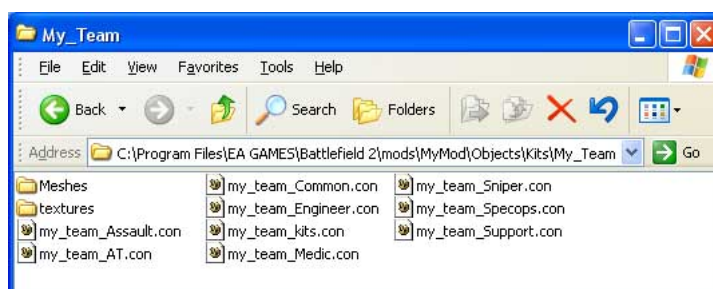


Note that you can expand the geometry section at the bottom of the tweak bar and click on each of the geometries to see the various kits.

Editing the .Con Files

Before we can actually get our team to use the new kits, we need to edit several .con files by hand.

43. Close the editor since we will have to re-launch it to get it to see the new files.
 44. In the tutorial files that came with this tutorial, locate all the .con files that are for the kits, like "my_team_Specops.con" and so on. (This should also include "my_team_common.con")
 45. Copy these files to "mods\MyMod\Objects\kits\My_Team". (This is your mod directory, not the RawData directory!) Your folder should now look like this:



If you are following this tutorial using the "my_team" files, then these will already be set up for you. Let's go through one anyway to see what changes you will have to make for your own custom team.

46. Copy the file "my_team_Support.con" and rename the new file "my_team_Frogman.con".
 47. Open my_team_Frogman.con in your text editor.
 48. Change the first line so that it reads "ObjectTemplate.create Kit my_team_Frogman."
 49. In the third line, note where it says "...kitType Support".
 50. In the next line, note where it says "...aiTemplate Support".

Note: You don't have to actually make any changes in steps 49 and 50. I just wanted you to see where you change these line

51. Two lines below this is the line which ends with "...geometry.kit 4". This is the number that corresponds to the "geom" notes that we set up in the scene file. Change the number to 9. Now our frogman will use the helicopter pilot kit.
 52. The next line ends with "...geometry.drop G eom 14" and of course is where you set which drop kit to use. You would change this number to whichever was appropriate in your scene file just like the other kits. In this case it won't really matter because all our drop kits are the same.

If you look further down in the file, you will see that this is where you set up which weapons, icons, and so on that this class will use. Don't change any of this for now. When you are done, the beginning of the file should look like this:

```

C:\Program Files\EA GAMES\Battlefield 2\mods\MyMod\Objects\Kits\My_Team\my_team_Frogman.con
ObjectTemplate.create Kit my_team_Frogman
ObjectTemplate.saveInSeparateFile 1
ObjectTemplate.kitType Support
ObjectTemplate.aiTemplate Support
ObjectTemplate.geometry my_team_Kits
ObjectTemplate.geometry.kit 9
ObjectTemplate.geometry.dropGeom 14
ObjectTemplate.networkableInfo KitInfo
ObjectTemplate.setHasCollisionPhysics 1
ObjectTemplate.dropHeadwear helmet
ObjectTemplate.abilityRestoreRate 0.04
ObjectTemplate.abilityInVehicleRadius 7
ObjectTemplate.abilityInVehicleStrength 0.2
ObjectTemplate.abilityInVehicleMaterial 91
ObjectTemplate.unlockIndex 5
ObjectTemplate.cullRadiusScale 2.4
rem -----

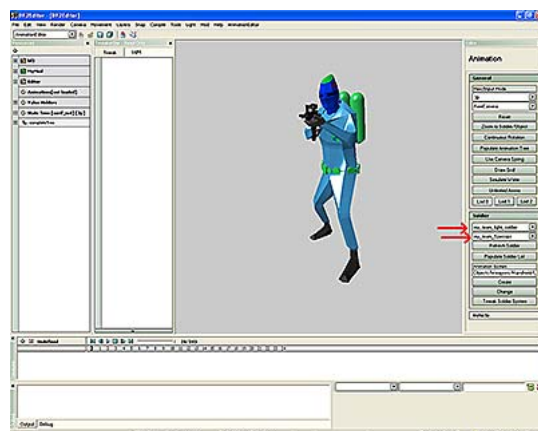
```

This is an example of how you can make extra kits, assign them an additional sequential "geom" Number in your scene file, export, then set up a new .con file that calls the new kit.

Using Your New Kits in the Editor

We've now done everything necessary to get our new kits in the game, so let's do that now.

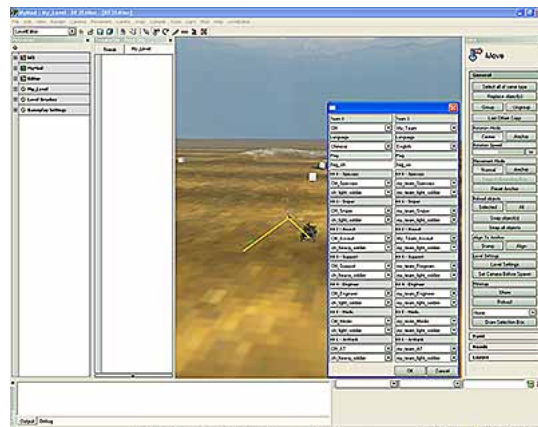
53. Launch the editor again, selecting the appropriate mod from the start-up window.
54. Once the editor is open, switch to the Animation Editor section.
55. In the Soldier section on the right side, select "My_team_light_soldier" or whatever custom soldier you are using. (Note that the custom kits may not show up on the retail soldiers.)
56. Next select "my_team_Specops" or whichever kit you would like to view and press the "Refresh Soldier" button. You should now be able to see your custom kit in action:



The last thing you need to know is how to get the kits into your maps.

57. Switch to the Level Editor and load "My_Level" or whichever level you would like to use.
58. After the level loads, click on the "Level Settings" button on the right side of the screen.
59. Here you will have to manually select which kit and soldier body you want for each class.

Notice in the following screenshot that I have selected the Frogman as the Support kit. This is an example of how you can make extra specialized kits for a team and then select them for certain maps:



You are now ready to package your map and test as usual.

A final note: I've included the finished kit scene files that you can use as reference. The file is named "My_ Team _Kits_final" (one each for Maya and Max). You can open these up to see how everything is set up. If you want to try importing/exporting this file, you will want to rename it "My_ Team _Kits".