

Creating Custom Assets for Multiverse

This document addresses the rather simple task of creating models in 3D Studio Max for use in the Multiverse MMO engine. It assumes a general knowledge of Max, and does not address any particular aspects of that program, save for those which directly impact its use with Multiverse.

Required Plug-ins and Applications

In order to export models from 3D Studio Max to Multiverse, we will need a handful of useful applications and plug-ins. They are:

Plug-In or Application	Available From
ColladaMax.dll	www.feelingssoftware.com
Moonshine Model Converter	bit.polter.pl/repository/BlackIceEng.rar
Multiverse Model Viewer	update.multiverse.net/tools/MultiverseToolsSetup.exe
Multiverse Asset Importer	update.multiverse.net/tools/MultiverseToolsSetup.exe

We will deal with each tool when necessary. For now, it's enough to simply download and properly install all of these utilities.

General Guidelines

A few notes on using Max with Multiverse in mind.

First, Multiverse favors the DirectDraw Surface (.DDS) file format for rendering textures in the engine. While Multiverse supports additional formats (including .PNG), it converts all such textures to .DDS during load time. Therefore, to avoid a lengthy overhead, it is best to use .DDS files for the majority of your textures.

Second, the reflectivity of a material in Multiverse can be controlled from 3d Studio Max's Material Editor window. You can do so by altering the color in the Specular box in the Blinn Basic Parameters rollout panel for any material. For a given texture, white in Max indicates a highly reflective surface in Multiverse. On the other hand, black indicates a matte surface. By default, materials in Max have a specularity that is 100% white. Therefore, it is important to always check the specular color assigned to a material before exporting it.

Finally, the proper size of a model in 3D Studio Max should reflect a 1-world-unit-to-1-centimeter ratio. This proportion can be set by accessing the Units Setup dialog box from the Customize menu, and clicking the System Unit Setup button that appears at the top of the window.

Collision Detection

Assets in Multiverse require collision volumes to be created in 3D Studio Max prior to export. Creating these collision volumes is fairly simple. To begin, take a moment to consider the easiest way of representing collision with your model. Multiverse allows for four types of collision volumes – Oriented Bounding Boxes, Axis Aligned Bounding

Boxes, Spheres, and Capsules. Oriented Bounding Boxes (OBBs) and Spheres are perhaps the easiest – and most common – collision volumes used, so we'll work exclusively with them.

To create an OBB, use the Box tool in the Create panel of 3D Studio Max. All OBBs must begin with the name prefix *mvcv_obb_*, followed by the name of the box. For example, an OBB might be named *mvcv_obb_Main*. Please note, while the appended name of the collision volume can contain capital letters, the prefix must be all lowercase.

To create a Sphere, use the Sphere tool in the Create panel of 3D Studio Max. All Spheres must begin with the name prefix *mvcv_sphere_*, followed by the name of the sphere.

Finally, you can create multiple collision volumes for a model, but consider that the more collision volumes present, the more processing power required. Stay simple!

Converting 3D Studio Max Models to Multiverse Format

The process of exporting models in Max to a format Multiverse understands is actually quite simple. Create a new model, or open an existing one. Take a moment to make sure that all of our textures are in .DDS format, and that they have been given an appropriate name – one that contains only letters, numbers, and underscores. Check the specularities of each material, so that reflective textures are more white, and matte textures are more black.

Next, highlight all models in the scene and select Export Selected... from the File menu. In the Select File to Export dialog box, set the Save As Type dropdown list box to COLLADA (*.DAE). Enter a name for your model, and navigate to a suitable directory in which the file can reside. Finally, click the Save button.

A ColladaMax Export window will appear, with options for saving your file. In the Standard Options field, make sure both Bake Matrices and Relative Paths are checked. In the Geometry field, check every box – Normals, Triangulate, XRefs, and Tangents/Binomials. Finally, make sure Enable Export is the only checked box in the Animation field, and click the OK button.

At this point, your model has been saved in Collada format, and can be tested using the Multiverse Model Viewer. To do so, open the utility and select Load Model... from the File menu. Make sure the Files of Type dropdown list box is set to Collada DAE Files (*.dae), and navigate to the directory where you saved your model. Select the model and click the Open button.

You may notice that your model is lying prone. This is because the coordinate system shared by Max and Collada is not the same as the coordinate system used by Multiverse. We'll take care of this in a moment. For now, check that the materials assigned to your model are present, and that the model is the correct size.

Next, we need to make some final adjustments to our model using the Moonshine Model Converter tool. First, open the utility. Click the Add File button, and navigate, select, and open your .DAE file. We only need to check one box in the Parameter field – the one marked “--3ds”. This indicates to the utility that our model’s coordinate system doesn’t sync with Multiverse. With that option selected, hit the Run button.

When Moonshine Model Converter has finished running, it will have created a handful of files located in the same directory as your .DAE file. Let’s say your model was saved as *barrel.dae*. Moonshine will have created two files – *barrel.mesh* and *barrel.material*. These files belong in the Meshes and Materials directories within your Asset Repository, respectively. Take a moment to copy and paste them where they belong. Likewise, any .DDS texture files that are utilized by your model should be copied and pasted within the Textures directory.

If you created collision volumes for your model, Moonshine will also have created a .physics file. The .physics file should be copied into the Physics directory in your repository.

If we were to view your model in the Multiverse Model Viewer utility right now, we would find that it is standing upright, as it should, but that our textures are not visible. This is because – for some odd reason – the .material file we just created is missing the filenames associated with our model. Fixing this is simple. First, open the .material file using WordPad. For each material block listed, look for a Texture_Unit block, and add the name of the associate .DDS file to the end of the line that reads “texture.” For example, this is an original .material file:

```
material grandfatherclock.clockface
{
    technique
    {
        pass
        {
            shading phong

            ambient 0.58824 0.58824 0.58824 1.00000
            diffuse 1.00000 1.00000 1.00000 1.00000
            specular 0.00000 0.00000 0.00000 1.00000 1.00000
            emissive 0.00000 0.00000 0.00000 1.00000

            texture_unit
            {
                texture
                tex_coord_set 0
            }
        }
    }
}
```

This is the same file with the filename amendment (which has been made boldfaced for your sake):

```

material grandfatherclock.clockface
{
    technique
    {
        pass
        {
            shading phong

            ambient 0.58824 0.58824 0.58824 1.00000
            diffuse 1.00000 1.00000 1.00000 1.00000
            specular 0.00000 0.00000 0.00000 1.00000 1.00000
            emissive 0.00000 0.00000 0.00000 1.00000

            texture_unit
            {
                texture map_clockface.dds
                tex_coord_set 0
            }
        }
    }
}

```

With our materials fixed, we can test our model in the Model Viewer by selecting Load Model... from the File menu, and navigating to the directory where our .mesh file resides (also, be sure Multiverse Mesh Files (*.mesh) is selected in the Files of Type dropdown list box).

A successful import in the Multiverse Model Viewer means we're ready to move on to the next section.

Importing Assets into Multiverse

The final step in the conversion process is to designate our model as an asset in Multiverse. Assets give an ordinary model another layer of complexity beyond simply a set of vertices and textures. That layer of complexity is basically how the model will be used within the engine.

Fire up the Multiverse Asset Importer utility. Select "Mesh" from the Asset Type dropdown list box. From the Category dropdown list box, select the category type that most reflects how your model will be used – for example, a barrel would most likely be a Prop. Give the asset a name in the Asset Name textbox. Finally, click the Browse... button to the left of the Mesh textbox, and navigate to the .mesh file we created earlier. If you'd like, you can also give your model an optional description at the bottom of the window.

With all necessary fields populated, select File > Save Asset As... from the menu bar. Give the asset an appropriate name, and save it in the AssetDefinitions directory of your Repository. Finally, our model is ready to hit the field.

Placing the Asset in an Existing World

To add a custom asset to a Multiverse world file, open the Multiverse World Editor utility. Load an existing world file, or create a new one.

In the hierarchy that exists on the right side of the screen, right-click the first item in the list – the one called World: *filename* – and select Create Object Collection. Give this collection a name.

Right-click on this new Collection item in the list, and select Add Object. In the Category dropdown list box, select the category in which you designated your asset – for example, props. Then, select your asset from the Objects dropdown list box. Give it a unique name, and then click the Add button. Your asset will be created as a child of the Collection item in the world hierarchy.

Although its been added to the world, its position may not be immediately apparent. Click on the name of your custom asset in the world hierarchy, and click the Position tab that appears below. Move your mouse over a section of terrain in the world viewport, and note the X, Y, and Z coordinates. Enter these coordinates in the X, Y, and Z fields, respectively. Your asset should appear at that exact location in the world viewport. You can also use the Scale and Rotation tabs to better position your asset.