UMA 2.11 FAQ

V1.1

Global Library

Q: How do I access the Global Library?

The global library is an option available from the UMA menu. Just select UMA, and then “Global Library Window”. This will open the global library. Dock it somewhere, and you’re good to go!

Q: What is the Global Library?

In UMA 1.X and 2.0, all slots, overlays and races had to be in a scene library. This works well for some types of games, but also has some drawbacks for other types (for example, having to add items to libraries in every scene). To get around these issues, we created the Global Library. This is a single library that tracks all UMA data (including the DCS data). To use the Global Library, you have to use the “dynamic” versions of all the libs (this is how the “getting started” prefab is setup).

Q: How do I add things to the Global Library?

You can do this several ways. The main way is to drop it into the Global Library window. Open the global library window (if it isn’t already), and then drag drop whatever you want onto the drop pad. (The drop pad is the rectangle that says “Drag Indexable Assets Here. Non-indexed assets will be ignored.”). If you drop a folder, it will recursively scan it for indexable items.

You can also right click on any indexed item (or select multiple items), and choose “Add selected items to the UMA Global Library”.

UMA will also detect if you add non-indexed slots and overlays to a recipe. If so, it will warn you you that they are unindexed, and give you a button to add them to the library. Just click the button to add it and make the warning go away.

Q: Do I have to use the Global Library? I kind of liked using the old libraries.

No, using the global library is a convenience. If you like doing things the old way, you can continue to do so. You will however, miss out on all the features and conveniences of the dynamic libraries (using Asset Bundles, the non-scene specific access, etc.).

Q: I've renamed a bunch of stuff, moved it around, and deleted some files. Now the global library is reporting errors, or some items are not found.

You can do one of two things here – You can empty the global library, and re-add all the items you want indexed (either manually, or using the "rebuild from project" button). Or you can repair the global library. Repairing the library will attempt to fix all the broken links, and will remove anything that is simply missing.

Q: Everything works great in the editor, but when I build the game and play it, nothing shows up!

The most likely thing is that your racedata is not in the library. Races are referenced on some items, even though they are looked up by name. This can cause a build to break if the racedata is not included in the library (since it won't be included in the build).

Q: Why do I have to add references in the Global Library? Can I just have them all the time and not worry about it?

Adding build references forces unity to include your indexed items in the build. Without them, your characters would be running around sans clothing. We don't keep build references for every item because unity likes to reload your project each time you compile something, and the global library can become glacially slow.

Q: My system is pausing for 30 seconds every time I compile when using the global library.

Clear the build references. Be sure to add them again when you're ready to build your executable.

Dynamic Character Avatar

Q: How do I Load and Save a character from the editor?

The "Load and Save" Menu can be accessed by selecting the top toolbar dropdown "UMA>Load and Save". Make sure an UMA is selected in the scene hierarchy before selecting a save option. These functions are designed to be used while the scene is running.

Saving

There are 4 different save options.

The "Save Selected Avatar(s)" options will works for non-dcs **avatars** and dcs **avatars**. The "Save DynamicCharacterAvatar(s)" options will only work for dcs **avatars**. This is optimized for dcs **avatars** and results in a smaller file.

Each version also has a save to *Text* option and a save to *Asset* option.
The Text option will generate a text file listing all the UMA's attributes.
The Asset option will generate a Unity asset file with the UMA's attributes. This has the added benefit of being able to see and modify it in the inspector.

Loading

There are two load options to select from when loading an UMA. Either will work for a dcs avatar or non-dcs avatar.
Use the *text* option if the avatar was saved in text format or *asset* if the avatar was saved in assets format. This should only be done at runtime.

Q: How do I programmatically Load and Save a character to/from a string?

You should get an AvatarDefinition from the character like this:

```
compressedString = Avatar.GetAvatarDefinition(true).ToCompressedString("|");
```

This will return the smallest possible string to rebuild the avatar. To load that string, rebuild the AvatarDefinition, and the load that. Example:

```
AvatarDefinition adf = AvatarDefinition.FromCompressedString(compressedString, '|');
Avatar.LoadAvatarDefinition(adf);
Avatar.BuildCharacter(false); // don't restore old DNA...
```

Q. Why do I need to call ForceUpdate() after calling BuildCharacter()? What does BuildCharacter actually do?

BuildCharacter() takes all the data that makes up your avatar, and generates an internal recipe for it. It merges the recipes, updates the colors and DNA, etc. But it doesn’t generate the

character's textures or mesh. To do that, you have to set the character as "Dirty" and ask UMA to rebuild it. That's what ForceUpdate() does.

Q. After telling the avatar to Build and Update, I can't programmatically change the data on it – it's being ignored!

This is because the update doesn't happen right away – it's queued and sent to the generator. You'll need to wait for it to be generated to alter it. The simplest thing to do is to add an event handler to the CharacterCreated event, and do it there.

Importing

Q: My models are really small or really large, but they look right when I drop them into a scene manually.

Meshes can have a different scaling factor than their game objects. Unity does some "scaling magic" behind the scenes, so you may have to adjust the import scale to account for the magic. In most cases, the scaling should be either 100, 1 or 0.01.

Q: When I import my model, it looks right, but when the slot is actually added to the avatar, it's rotated 90 degrees.

This can be one of two things. First thing to do is to make sure the FBX is correctly exported. In Blender, you should change "Forward" to "Z-Forward" (it's usually set to "- Z-Forward" by default) "Up" should be set to "Y-Up". 3dsmax and Maya do not need any special settings for export.

If this does not help, it's possible there is a problem with the transform. Make sure in Blender, you have applied any modifiers (except the Armature modifier), and that you apply the rotation/scale/Location also. In 3ds max, to do this you would use "Reset xform".

Q: When I import my model, it looks right in the scene, but when the slot is added to the avatar, parts of it stretch off the screen, or down to the floor!

This is usually a problem with weighting – specifically, weighting to a bone that does not exist. Review your vertex weightings in your application. Typically, you would do a "transfer weights" in Blender, or in 3ds Max, you would do a Skin Wrap. This would pull the weights from the base model to the new outfit. (Typically you would also need to tidy up the weighting using weight painting).