From: <http://jabelarminecraft.blogspot.com/p/creating-custom-entities.html?m=1>

**[Jabelar's Minecraft Forge Modding Tutorials](http://jabelarminecraft.blogspot.com/?m=1)**

▼

[//www.blogger.com/rearrange?blogID=8871285205929815684&widgetType=PageList&widgetId=PageList1&action=editWidget&sectionId=crosscol//www.blogger.com/rearrange?blogID=8871285205929815684&widgetType=PageList&widgetId=PageList1&action=editWidget&sectionId=crosscol](//www.blogger.com/rearrange?blogID=8871285205929815684&widgetType=PageList&widgetId=PageList1&action=editWidget&sectionId=crosscol)

**Minecraft Modding: Creating Custom Living Entities**

**Introduction**

A living entity is actually fairly complex, in that it needs to have a textured model that gets automatically rendered, that probably moves (perhaps with artificial intelligence), is probably animated (with model and perhaps texture), makes sound, has a name, spawns naturally, has a spawn egg, etc.

Each step is fairly simple, but I found every time I would forget a step and wonder why my custom entity didn't appear, or why it didn't move correctly, and I realized I needed a list to follow every time I made a custom entity. This tutorial is that list -- I made it for myself but think it contains enough useful information that it is worth sharing. I hope you find it useful!

**Create New Entity Class**

**Extend And Implement The Appropriate Vanilla Class And Interfaces**

Making a custom entity usually involves extending some class and copying/modifying some code from similar classes. To understand what you should extend versus what you should copy, check out my explanation: [Jabelar's Know When To Extend Versus Copy Tutorial](http://jabelarminecraft.blogspot.com/p/minecraft-forge-172-know-when-to-copy.html)  
  
Your custom entity should be an extension of some vanilla entity kind of class, but it is important to figure out which one is best on which to base your entity.

Some major features of the vanilla entity base classes and interfaces to help you choose:

* EntityCreature is an EntityLiving that has AI (it moves with path-finding and attacks).
* EntityAgeable is an EntityCreature that can come in various sizes/ages.
* EntityAnimal is an EntityAgeable that enables breeding by implementing IAnimal.
* EntityTameable is EntityAnimal that implement IEntityOwnable

**Use Encapsulation When Coding**

I tend to use protected fields to allow classes to easily be extended, but full encapsulation can be done if desired. Encapsulation is basically the technique of using public methods to access class fields instead of accessing them directly (in strict encapsulation all your fields should be private).

So don't access the fields directly but rather "setter" and "getter" functions for all modifications of class private and protected fields, including those within the class.  
  
**Tip**: Encapsulation setter methods are highly recommended to facilitate good packet sync because it is the point at which a change happens (the setter functions can ensure to send a sync packet every time a field is changed).

**Implement (and override) all the relevant methods**

You should inspect the source code for the class you have extended. I recommend copying it all into your class, and adding @Override to each method. Then go through each method and decide whether you need to change the behavior. If not, you can delete it from your class. Otherwise, code it up for your custom behavior.

If there are other entities in *Minecraft* that have behavior that is similar to what you like, go ahead and copy it over.

Whenever you copy code, first of all use *Eclipse* to help you clean it up (fix the imports, add the @Override, fix any warnings, and so forth).

Key point: When you copy code make sure you try to understand it. As you learn how the code works, add comments and reorganize the code to suit your style.

**Apply Entity Attributes In Your Entity Class**

I explain the attributes fully in this supplementary tutorial: [Attributes For Custom Living Entities](http://jabelarminecraft.blogspot.com/p/minecraft-forge-1721710-apply.html)

**Set Up AI, Including Custom AI Classes If Necessary**

I explain the AI fully in this supplementary tutorial: [Jabelar's Custom Entity AI Tutorial](http://jabelarminecraft.blogspot.com/p/minecraft-forge-1721710-custom-entity-ai.html)

**If Your Entity Attacks, Override attackEntityAsMob()**

This method is one of the most important methods in an Entity as this is where much of the effects of attacks is evaluated, including invulnerability, knockback, hurt timer cooldown, etc. I explain how to override this method in this supplementary tutorial: [Jabelar's Custom Entity Attacks Tutorial](http://jabelarminecraft.blogspot.com/p/minecraft-forge-1721710-custom-entity.html)  
  
Also set attackDamage shared monster attribute as explained in the referenced tutorial on attributes above. If you're interested in knowing the attackDamage value of vanilla *Minecraft* entities, it is listed on the [*Minecraft* Gamepedia](http://minecraft.gamepedia.com/Damage#Damage_inflicted_by_mobs).

**To Set Behavior Of Entity In Response To Attacks, Override attackEntityFrom()Method**

This method is where the damage is actually inflicted on the victim, and there is some processing for invulnerability.  
  
You may or may not need to do much here. For example, for my EntitySerpent the super method (from EntityAnimal) was fine for my purpose so I did not override the method. But for my EntityBigCat which has ability to sit like an EntityWolf I needed to override to ensure that it would stop sitting when attacked. So method looked like:

@Override

public boolean attackEntityFrom(DamageSource par1DamageSource, float par2)

{

if (isEntityInvulnerable())

{

return false;

}

else

{

aisit.setSitting(false);

return super.attackEntityFrom(par1DamageSource, par2);

}

}

**If Your Entity Is Tameable, Implement IEntityOwnable (Or Extend EntityTameable)**

After you add the implement IEntityOwnable to your class, *Eclipse* should warn you about unimplemented methods. Accept *Eclipse*'s suggestion to add those methods and you should see some new methods added to the bottom of your class file.

You then need to put in code into the interface methods -- look at other entities that implement IEntityOwnable for ideas.

**If Your Entity Should Mate And Have Children, Implement IAnimal (Or Extend EntityAnimal)**

If you wish to have mating behavior add the implement IAnimal to your class, and *Eclipse* should warn you about unimplemented methods. Accept *Eclips*e's suggestion to add those methods and you should see some new methods added to the bottom of your class file.

You then need to put in code into the interface methods -- look at other entities that implement IAnimal for ideas.

**Create New Model Class**

If you just want to re-texture an existing entity, you can use an existing model, or you can copy or extend an existing one.

For custom models (including possibly animating them), consult my tutorial here: [Jabelar's Complex Entity Animations Tutorial](http://jabelarminecraft.blogspot.com/p/minecraft-forge-1721710-creating.html)

**Create Custom Renderer Class**

The Renderer class is mostly for associating the textures to the model, and should extend RenderLiving.

**Tip**: the float parameter in RenderLiving() is the shadow size for the entity.

For some entities, you may want to change textures based on some field in the entity, for example vanilla wolves change eyes to blue when tamed, or red when angry -- this is implemented by switching to a different texture. So it is good to pass the render call to one that casts the entity appropriately then queries the fields/methods of interest.

Note that you do NOT do model animations (like moving or re-sizing parts of the entity) in this class, instead you do that in the render() method of the model class. You CAN do texture animations here though if you want by changing textures as needed.  
  
Example:

public class RenderSerpent extends RenderLiving

{

protected ResourceLocation serpentTexture;

public RenderSerpent(ModelBase par1ModelBase, float parShadowSize)

{

super(par1ModelBase, parShadowSize);

setEntityTexture();

}

@Override

protected void preRenderCallback(EntityLivingBase entity, float f)

{

preRenderCallbackSerpent((EntitySerpent) entity, f);

}

protected void preRenderCallbackSerpent(EntitySerpent entity, float f)

{

// some people do some G11 transformations or blends here, like you can do

// GL11.glScalef(2F, 2F, 2F); to scale up the entity

// which is used for Slime entities. I suggest having the entity cast to

// your custom type to make it easier to access fields from your

// custom entity, eg. GL11.glScalef(entity.scaleFactor, entity.scaleFactor,

// entity.scaleFactor);

}

protected void setEntityTexture()

{

serpentTexture = new ResourceLocation(WildAnimals.MODID+":textures/entity/serpents/python.png");

}

/\*\*

\* Returns the location of an entity's texture. Doesn't seem to be called

\* unless you call Render.bindEntityTexture.

\*/

@Override

protected ResourceLocation getEntityTexture(Entity par1Entity)

{

return serpentTexture;

}

}

**Register The Renderer**

Rendering only happens on the client side, so the registration should occur in the ClientProxy class' preInit() event handling method. You can learn more about why we use proxies here: [*TheGreyGhost*'s Tutorial On *Forge* Intialization](http://greyminecraftcoder.blogspot.com/2013/11/how-forge-starts-up-your-code.html).

For example:

RenderingRegistry.registerEntityRenderingHandler(EntitySerpent.class,

new RenderSerpent(new ModelSerpent(), 0.5F));

**Organize Entity Texture Assets**

It is important to put the texture in the correct location. For 1.7.2 it should be in the filing path of: <your mod folder>/src/main/resources/assets/<your modid>/textures/entity/<folder for your entity>.  
  
But for *Eclipse* to recognize it as a resource, you should create the package first (which will create the right folder for you) and import the file into that package. So within *Eclipse* you need to make a new package called assets.textures.entity.<your entity folder> in the src/main/resources folder of your project. Then import your actual texture (PNG formatted graphics file) into that package.  
  
**Note**: I've seen mentioned that the name of the texture should be all lowercase, so I always do that, but I'm not certain that is actually necessary.  
  
**Tip**: Creating a good texture for your model is often time consuming and something I suggest you perfect at the end of the model creation process, so before I have a proper texture file I drop a file that is filled with single easy-to-see color into the assets folder during development. This is preferable to no texture file (which will cause model to render with a default pink-and-black checker pattern) because it ensures you've got the asset organized and called correctly in your code.  
  
Once you have your texture asset properly imported into proper location, you can reference this file from your code as a resourceLocation with a string value of your modid + ":" + the remaining file path starting with textures folder. For example, in the entity's custom Renderer class you set the texture as follows:

protected void setEntityTexture()

{

birdOfPreyTexture = new ResourceLocation("wildanimals:textures/entity/birdsofprey/eagle.png");

}

**Organize Entity Sound Assets**

Sound files should be OGG format, properly placed in the assets directory, and indexed in the sounds.json file. See my supplementary tutorial: [Jabelar's Sound File Tutorial](http://jabelarminecraft.blogspot.com/p/minecraft-forge-1721710-organizing-and.html).

**Update Lang File For Entity**

See my tips tutorial on .lang files here: [Jabelar's Lang File Tutorial](http://jabelarminecraft.blogspot.com/p/minecraft-forge-1721710-modding-tips_22.html)

**Confirm the Bounding Box**

If you created a custom model, it is important to check that the bounding box (a.k.a. the hitbox, a.k.a. collision box) is aligned properly to the model.

**Tip**: You can press F3+B to display the bounding boxes of all entities.

If you find that your bounding box is the wrong size, you can adjust the setSize() method in your custom entity class' constructor.

If you find that your bounding box is misaligned, you need to go into your model and adjust all your rotation points to move them into alignment. (Alternatively you can use a translation transformation in the render() method of your entity's model class).

**Register The Entity**

Previously people registered their entity using the global ID system. However, the problem (potentially) is that that is limited to 256 entities which could cause difficulty if you ran your mods concurrently with others that also added entities. Therefore it is considered better practice to use the registerModEntity() method. which uses the name string as a key.

The problem though is that the built-in spawn egg system doesn't work easily with the registerModEntity() method, so you also have to make your own custom spawn egg (which I show you how to do below).

Of course if you don't want a spawn egg, then you can skip that part. I wanted a spawn egg so created a custom registry method that both registers the custom entity and also associates a spawn egg (with colors) similar to the built-in method. For example:

registerModEntityWithEgg(EntitySerpent.class, "Python", 0x3F5505, 0x4E6414);

which calls:

public void registerModEntityWithEgg(Class parEntityClass, String parEntityName,

int parEggColor, int parEggSpotsColor)

{

EntityRegistry.registerModEntity(parEntityClass, parEntityName, ++modEntityID,

WildAnimals.instance, 80, 3, false);

registerSpawnEgg(parEntityName, parEggColor, parEggSpotsColor);

}

See further below for the spawn egg.  
  
**Tip:** Fast moving entities may need "tracking" enabled. In the registerModEntity() method, the last three parameters are related to motion "tracking". Basically in any multiplayer game there is an issue where you can't rely on the server to sent every position update every tick because if it misses a tick it will cause jerky movement and lag. So most systems let the client side help with the "interpolation" (calculating the movement between server syncs). In this example above I have tracking off (false) but if you have a fast moving entity like a thrown object or projectile you probably want to turn it on (true). The only thing is that you still need to specify how far and how frequently the tracking is synced. If you're interested in the values used by vanilla entities, check out the code for the EntityTracker.addEntityToTracker() method.

**Create Entity Extended Properties**

For most custom entities, you'll add fields to hold information that is specific to your entity. This information is not automatically saved or synced, so you need to make sure that happens.

There is an interface provided called IExtendedEntityProperties that helps handle these properties. See my tutorial at: [Jabelar's Extended Entity Properties Tutorial](http://jabelarminecraft.blogspot.com/p/minecraft-17x.html).

**Register Natural Spawns If Desired**

EntityRegistry.addSpawn(EntityLion.class, 6, 1, 5, EnumCreatureType.creature,

BiomeGenBase.savanna); //change the values to vary the spawn rarity, biome, etc.

**Create Custom Spawn Egg**

You may want to have a spawn egg for your custom entity. If you use the global ID method for registering entities, it comes with a spawn egg alternate constructor. However, as mentioned above, the mod entity ID method for registering entities is considered more proper as it has no potential for running out of ID numbers.

In that case, if you want a spawn egg, see: [Jabelar's Custom Spawn Egg Tutorial](http://jabelarminecraft.blogspot.com/p/minecraft-forge-1721710-creating-custom.html)

**Create Packet System to Keep Your Client and Server In Sync**

Your server may update class fields in ways that you want to reflect graphically, either in the render or gui on the client side. For example, if you want to do a specific animation when a certain AI task is being followed. In that case you need to send a packet from server to client.

For a tutorial on using packets for syncing entities, see [Jabelar's Tips For Custom Packets](http://jabelarminecraft.blogspot.com/p/minecraft-forge.html).  
You can feel free to use other packet systems, as there are several good tutorials that take similar approaches. The point is you need to be able to send a packet from your server to your client to update custom fields in your entity class on the client.

**Ensure Client Entity Is Sync'ed To Server**

Once you have a packet system, any time a custom field in your entity is updated on the server, for example with AI, you need to send a packet to the client. Since I recommend creating "setter" methods in your entity class to control the changes of any custom fields, you can simply send the sync packet from the setter method. This ensures that the sync happens every time there is a change, but prevents unnecessary packets (i.e. don't send sync packets every tick).

**Test Your Entity**

Now, finally, you should have a fairly complete custom entity. When you run *Minecraft* you should start a creative mode game and test it out. There are four ways you can get your entity active in the game:

1. If you registered natural spawns, you may find that it has generated naturally.
2. If you created a spawn egg, you could use that.
3. You can use the /summon command plus the entity name in the form: <modid>+"."+<entityLocalizedName>. For example: "/summon wildanimals.Python"
4. For fun create your own summoning command. See: [Jabelar's Custom Server Command Tutorial](http://jabelarminecraft.blogspot.com/p/minecraft-forge-172.html).

**Conclusion**

In summary, there is actually quite a lot of things you have to do to create a custom entity. Hopefully the above information helps you remember all those steps.

I hope this inspires you to create custom entities. Please feel free to write with suggestions for corrections or clarifications!

**6 comments:**

[Ростислав Могила](https://www.blogger.com/profile/12055765723812578501)[May 15, 2015 at 10:33 AM](http://jabelarminecraft.blogspot.com/p/creating-custom-entities.html?m=1)

Hello! When i try to make my mob Ownable, he just fall trough the ground and the console gives me an GL error 1284.  
Can u help me pls? i tryed all i can... i dont know what to do now.

[Reply](javascript:;)<https://www.blogger.com/delete-comment.g?blogID=8871285205929815684&postID=772454021449458027>

[Replies](javascript:;)

[Julian Abelar](https://www.blogger.com/profile/03003230541420264945)[May 18, 2015 at 8:29 PM](http://jabelarminecraft.blogspot.com/p/creating-custom-entities.html?m=1)

GL errors usually come from the render class. The 1284 error I think means that you're not popping or pushing the matrix properly (you need to have one pop for every push otherwise the memory stack will overflow). I suspect that in your render class, when the entity is owned it probably tries to render a collar or some other change. So look in your render class whereever it checks to see if the entity is owned. There you're probably missing a matrix push/pop.

<https://www.blogger.com/delete-comment.g?blogID=8871285205929815684&postID=249648863932142766>

[Reply](javascript:;)

[main](https://www.blogger.com/profile/11777510638358093011)[June 5, 2015 at 10:27 PM](http://jabelarminecraft.blogspot.com/p/creating-custom-entities.html?m=1)

[Sims 3 PC Game - Compressed Games Free Download](http://www.muhammadsakil.com/sims-3-pc-game/)  
[the sims 4 download free game - Compressed Games Free](http://www.muhammadsakil.com/the-sims-4-download-free-game/)  
[minecraft download free game - Compressed Games Free](http://www.muhammadsakil.com/minecraft-download-free-game/)  
[Garrys Mod Free Download Pc Game Compressed Full](http://www.muhammadsakil.com/garrys-mod-free-download-pc-game-compressed-full/)  
[Spore Game free Download full version - Compressed](http://www.muhammadsakil.com/spore-game-free-download-full-version/)  
[dayz standalone download free game - Compressed Games](http://www.muhammadsakil.com/dayz-standalone-download-free-game/)  
[the sims 2 download free game - Compressed Games Free](http://www.muhammadsakil.com/the-sims-2-download-free-game/)  
[2014 FIFA World Cup Brazil Game - Compressed Games](http://www.muhammadsakil.com/2014-fifa-world-cup-brazil-game/)

[Reply](javascript:;)<https://www.blogger.com/delete-comment.g?blogID=8871285205929815684&postID=4723354563715787560>

[Shark bladez](https://www.blogger.com/profile/07904310696640181042)[July 14, 2015 at 9:38 AM](http://jabelarminecraft.blogspot.com/p/creating-custom-entities.html?m=1)

What would i do if i wanted my mob to fly somewhat like a ghast?

[Reply](javascript:;)<https://www.blogger.com/delete-comment.g?blogID=8871285205929815684&postID=1922861612495909564>

[Replies](javascript:;)

[Julian Abelar](https://www.blogger.com/profile/03003230541420264945)[July 14, 2015 at 10:39 AM](http://jabelarminecraft.blogspot.com/p/creating-custom-entities.html?m=1)

The easiest way is to have your entity class extend EntityFlying.

<https://www.blogger.com/delete-comment.g?blogID=8871285205929815684&postID=6541493805516012897>

[Reply](javascript:;)

[sky01](https://www.blogger.com/profile/05959290210439689943)[September 12, 2015 at 9:42 PM](http://jabelarminecraft.blogspot.com/p/creating-custom-entities.html?m=1)

When registering entities with EntityRegistry.registerModEntity, you suggest using "++modEntityId" -- what should this be initialized to? Zero?

[Reply](javascript:;)<https://www.blogger.com/delete-comment.g?blogID=8871285205929815684&postID=8157290205409712301>

<javascript:;>

[Load more...](javascript:;)

<https://www.blogger.com/comment-iframe.g?blogID=8871285205929815684&pageID=6649015753121698594&m=1&blogspotRpcToken=9619423>

[Home](http://jabelarminecraft.blogspot.com/?m=1)

[View web version](http://jabelarminecraft.blogspot.com/p/creating-custom-entities.html?m=0)

Powered by [Blogger](https://www.blogger.com).

[//www.blogger.com/rearrange?blogID=8871285205929815684&widgetType=Attribution&widgetId=Attribution1&action=editWidget&sectionId=footer-3//www.blogger.com/rearrange?blogID=8871285205929815684&widgetType=Attribution&widgetId=Attribution1&action=editWidget&sectionId=footer-3](//www.blogger.com/rearrange?blogID=8871285205929815684&widgetType=Attribution&widgetId=Attribution1&action=editWidget&sectionId=footer-3)