

Overview

In following these tutorials, users will learn how to create a character designed for the CryENGINE. Please make sure you have all the necessary plugins installed. There are various validation tools which will warn you if something is setup incorrectly upon export but if there are still issues this page should help as well.

- [Overview](#)
- [Example Files](#)
- [In This Topic](#)
- [File Layout](#)
- [The Render Mesh \(.skin files\)](#)
- [Live Deforming Skeleton \(.chr files\)](#)
- [Phys Skeleton](#)
- [For Pre-3.5 Builds](#)
 - [Ragdoll Skeleton](#)

Example Files

Before continuing you will want to download the [Agent.rar](#). This file contains everything the topics below will discuss.

In This Topic

- [Character Authoring in 3ds Max](#) — This section covers how to create characters and export from 3ds Max.
- [Character Authoring in Maya](#)

File Layout

Naming of bipedal hierarchies does not need to match the 3ds Max organizational method. This is only necessary if you want to make use of the existing animations. There are a few other things already preset with the Bip naming conventions but can be changed in the file which you define the parameters for IK, the .chrparams file. There is some hard coded joints as well which go into detail here: [Leg and Foot Ground Alignment](#)

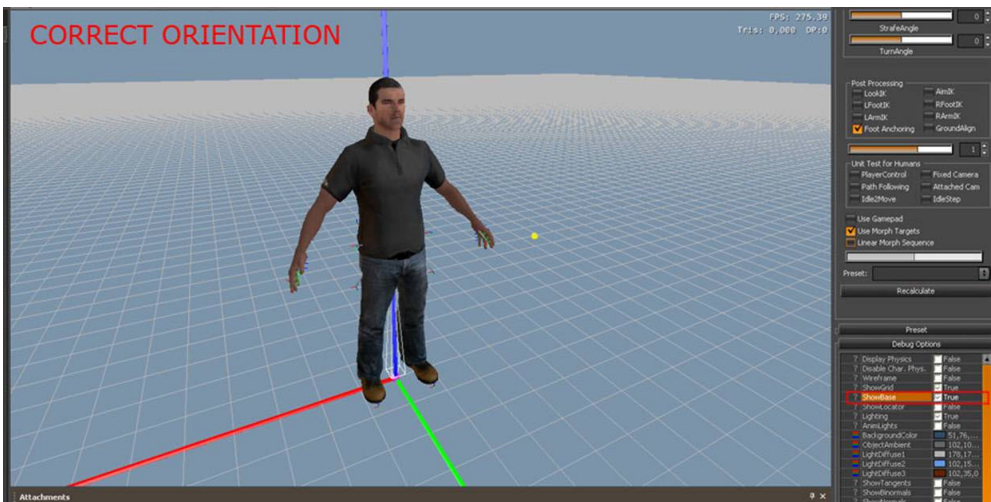
For the most part, a .chr is the main CryENGINE character file format. There is also .skin which is used for skinned meshes. Whatever skinned attachments you have will be attached to the .chr in the Character Editor and from there a .cdf file will be saved which will be used as the main character file from there. For a complete overview of the file formats used in CryENGINE, see: [Art Asset File Types](#).

The following is a brief explanation of the main terminology which will be used when creating characters. These terms are consisted of render meshes (.skin attachments), a general animated skeleton (.chr files), and the physics skeleton (included in the .chr file).

The Render Mesh (.skin files)

These are the mapped character geometries that you see in the game which will be exported as a .skin.

- These are the assets which can have LODs (Level of Detail) meshes with lower polygonal resolution that automatically fade in based on your distance from the character (in the game world).



Things to keep in mind before you skin the render mesh to a rig:

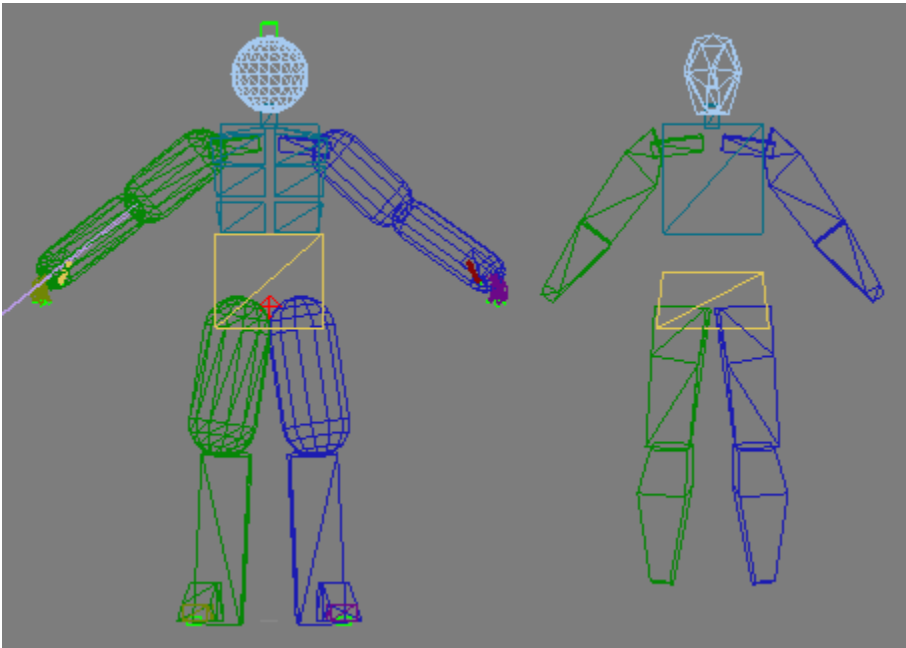
- Its pivot should be at 0,0,0.

- The character must be natively set with Z up and Y forward. Maya users working in a Y-up environment might need to create a SceneRoot node. For more information visit: [Character Authoring in Maya](#).
- It should have no transformations applied (pos, rot, scale) to the node being used to export it.
- Make sure the model matches the joints in the skeleton.
- The live deforming skeleton should be in its bind pose when you apply the Skin modifier.
- To get the best preview of skinning you see in CryENGINE in your DCC tool be sure your skin modifier is using Dual Quaternion with no more than a maximum of 4 bone influences per vertex (or 8 if you export 8 weights). In 3ds Max the Cryskin modifier already has these prepared set.

Live Deforming Skeleton (.chr files)

This is the skeleton that deforms the render meshes (animation) and also has physics geometry parented to it which is exported as a .chr file. Bones can be used for anything whether its simple animating or helpers for weapons or particle effects to emit from.

- The geometry of this skeleton is used for hit detection and physics in the live character (non-ragdoll).
- The materials applied to the geometry are used for hit detection, to tell the engine what part of the character you have hit.



Here are some general tips to follow when setting up your rig:

- There can be attachment nodes in the hierarchy such as:
 - Example: \$weapon_bone; this is where the weapon will be attached in the engine.
- Other than attachment nodes, there can be some additional nodes used for automatic foot plant and gait detection:
 - \$Bip01 L Heel and \$Bip01 L Toe0Nub on the left foot (also present on the right foot).
- If you want to have rig elements inside the hierarchy, keep them to the periphery and use the prefix "_" to comment them and their children out of export.
- The skeleton must face the +Y axis and +Z up.
- Nodes in the hierarchy should have no scale applied to them.
- The root bone pivot should be placed at 0,0,0 in the world coordinate space and must face +Y forward and +Z up without any additional offset.
- The bones of the spine and head need to match the orientations and names of the biped skeleton for lookIK to work but these can be altered.
 - The names are Bip01 Head, Bip01 Neck, Bip01 Spine3, and Bip01 Spine2.
 - If you must have them in the hierarchy, and they are exported into the engine, try not to have them be splines or shapes; convert them to polygons.
 - Keep in mind that any rig elements inside the hierarchy that are exported will be null bones in the engine and will take up memory/CPU time, so do this sparingly.
- It's a good idea to put the hit material submats first for humanoid skeletons that you will re-use. This way, the numbers will not change based on how many different texture submats you use.
- Make sure that all the models are NOT linked to anything. Having, for instance, the pants of your character linked to its hands will cause an error in the exporter, causing it to create multiple skeletons.

Phys Skeleton

This skeleton is nested in the same scene as the Live Deforming Skeleton. You can think of it as a set of switches; a node being present in this skeleton signifies that it's counterpart in the live deforming skeleton is physicalized in the engine while the character is alive and dead. In Maya this skeleton is only used for parentFrame nodes as the Live Deforming Skeleton can store all the physics parameters without it even being able to parent the physics proxies directly to the joints.

Here are some general tips to keep in mind while preparing this skeleton:

- Each node in this skeleton also stores physical properties for its corresponding bone in the deforming hierarchy; this is stored in the phys bone's IK properties. Regardless of the program you are working in to export it this skeleton is crucial for an effective and clean ragdoll simulation. For more information on ParentFrames see: [IK, Rotation, Dampening, and ParentFrames](#)
- Since the live skeleton already contains physical geometry; this skeleton will be most likely be a duplicate of those shapes.
- The pivots location and orientation will have to be identical to the joints they are associating themselves with.

For Pre-3.5 Builds

Ragdoll Skeleton

This is the skeleton that deforms the render meshes in the event of ragdoll simulation. This skeleton is exported as a LOD of the Live Deforming Skeleton. This skeleton is usually similar (in most cases identical) to the Live deforming skeleton with its associated physics proxies.

Each node in this skeleton also stores physical properties for its corresponding bone in the deforming hierarchy, this is stored in the phys bone's IK properties.