



# **WEAPON CREATION- ANIMATOR'S POINT OF VIEW**

**BY RAVEN SOFTWARE**

SoF2 inview weapons comprise three separate components: the weapon itself (“gun”), hands, and a special “buffer” object that set up where hands and accessories “bolt on” to the gun. Here’s how they are connected with a more in-depth description

The **gun** is the main parent object, which includes the weapon models and bones. The **buffer** bolts on to the gun and includes reference bones or points that are in the correct position and rotation for things like accessories and hands to attach to. Finally, there are the **hands**, which include models and bones. In this same hand category, we include accessories, which are attached at the same level as the **hands**, to the **buffer**. Now that the hierarchy is explained, we will look at the M4 weapon and I will try and explain all of its components in depth.

## **M4**

### **Gun**

At the top-most part of the gun hierarchy, as in all .glm’s, we have the **model\_root**. This null must be located at 0,0,0 in 3d space. Underneath that, its children are the **mesh\_root** and **skeleton\_root**, at the same branch level. These also, must be located at 0,0,0 in 3d space. **Mesh\_root** is where we keep all of the geometry and Level Of Detail information. The highest LOD always is under one model object (i.e. **bodyfront**) and has an original, unique name. To create subsequent LODs, you must retain the exact same hierarchy as the highest; only add a suffix “\_x” with “x” being the next LoD number. In SoF2, we’ve created three levels of detail, and using the bodyfront object as an example, are named as: **bodyfront** (lod0) **bodyfront\_1** (lod1), and **bodyfront\_2** (lod2), each with their own hierarchies, under the **mesh\_root**. Below is a complete layout of what the M4 hierarchy looks like (tabbing denotes parent/child relationship):

```
model_root
  mesh_root
    bodyfront
      barrel
        barrelh
      bodyfronth
      clip
        cliph
      frontsight
        frontsighth
      gunbody
        gunbodyh
      gunmuzzle
        gunmuzzleh
      handle
      sight
        sighth
      stocktube
        gunstock
```

```

        stocktubeh
bodyfront_1
    barrel_1
        barrelh_1
    bodyfronth_1
    clip_1
        cliph_1
    front sight_1
        front sight h_1
    gunbody_1
        gunbodyh_1
    gunmuzzle_1
        gunmuzzleh_1
    handle_1
    sight_1
        sight h_1
    stocktube_1
        gunstock_1
        stocktubeh_1
bodyfront_2
    barrel_2
        barrelh_2
    bodyfronth_2
    clip_2
        cliph_2
    front sight_2
        front sight h_2
    gunbody_2
        gunbodyh_2
    gunmuzzle_2
        gunmuzzleh_2
    handle_2
    sight_2
        sight h_2
    stocktube_2
        gunstock_2
        stocktubeh_2
skeleton_root
    gun    (bone buffer attaches to)
        bolt1
        bolt2
        clip1
        clip2
        door
        option1
        option2

```

option3  
option4  
option5  
option6  
option7  
option8  
stock  
trigger

In the **model\_root** hierarchy, I tried to give each object unique names that would help me know what they were. In the case of the M4, I gave one object the name; **stock**, while I named its child, which I separated into the hidden right side that is never seen inview, as **stockh**. I wanted to know which objects should be hidden and which should be visible when inview so I added an “h”, but they can be named anything you want.

Now we come to the **skeleton\_root**. All **skeleton\_root** models must have its first and only child be named “**gun**” if it’s to be an inview weapon. This bone must have at least one object in the **mesh\_root** hierarchy point weighted to it. For every single weapon in SoF2, for simplicity sake, they share the same names in all skeleton bones, as written above. Now, not every weapon uses all 16 bones under **skeleton\_root**, but carcass will keep only the ones that have model objects point weighted to them.

## ***Buffer***

The buffer bone was created to make organization and sharing of animations easier. Basically, the buffer is just a group of bones located throughout 3d space in positions where things like hands, accessories, and effects, are attached too. In SoF2, there are 6 buffers to start off with, which are separated into different categories. They are:

Acc (accessories) –buffer for knife and binoculars

At (antitank) –buffer for mm1, and rpg7

Gren (grenade) –buffer for all grenades

Mg (machine gun) –buffer for m60

Pistol –buffer for ussocom, micro Uzi, and m1911

Rifle –buffer for m4, ak74, msg90a1, usas12, oicw, m590, and m3a1

The idea with the buffer is that, with it, we have the possibility of sharing animations. With many weapons, in some animations, all the hands are doing is holding onto the gun. For instance, the hand would hold a gun in the same position for the ready, put away and idle animations. Rather than duplicate these files, I only exported two frames of the hand in its “gun hold position”. I then moved a bone in the buffer, designated to be the position where the right hand is attached. Since the buffer is attached to the gun (at the **gun** bone under **skeleton\_root**) the right hand will move with its parent, the weapon. So now, it doesn’t matter if the gun is moving through space while being taken out, put away, or just sitting there, because the hand will be in it’s “gun hold position” and be attached to the gun, thereby saving on animation space.

As a modder, you have the choice of creating your own buffer for each weapon you choose, or using the existing buffers. I only recommend using existing buffers if your changes are, for the most part, just texture tweaks, because they will tell you where your hands, muzzle flashes, and ejection shells will be located.

Here is a list of what is included in the M4 portion of the Rifle buffer (tabbing denotes parent/child relationship):

**model\_root** (need to be at 0,0,0 3d space)

**mesh\_root**

**stupidtriangle** (in order to be carcass-able, there needs to be at least one piece of geometry point weighted to the first bone. At the time, I thought this was a pretty stupid rule and thus, the name.)

**skeleton\_root** (need to be at 0,0,0 3d space)

**handle\_m4\_always\_**

**ejection\_m4\_always\_** (the “\_always\_” means that carcass will not throw away this bone, even though there isn’t any geometry point weighted to it) –this bone is where ejection shells come from.

**flashm203\_m4\_always\_** -this is where the muzzle flashe effect is attached to.

**lhand\_m4\_always\_** -bone where lhand is attached to

**m203\_m4\_always\_** -bone where m203 is attached to

**rhand\_m4\_always\_** -bone where rhand is attached to

## Hands

Hands are separated into the left and the right models and are attached to the buffer bones. Again, I’ve separated the hand models into separate pieces for hiding and saving on polygons. The model\_root for these must also be at 0,0,0 3d space. Since there is only one hand model being used throughout the entire game, all existing animations will be required in order add and carcass new ones.

## External files

Once the models and animations are carcassed and a .glm is created (see “**SoF2\_Weapons\_Overview.doc**” for more info), it’s on to getting the game to recognize them.

## SOF2.inview

The **SOF2.inview** file can be edited with notepad and is where one can add new sounds, and piece together a new weapon. This file will tell the game where to find all of its assets and also set things like animation playback speed. This is also the file that foreshortens the gun to a “.6” scale on the forward/backward axis. By scaling the gun,

we eliminate much of the distortion that the field of view creates when a weapon is turned to the side. (see “**SoF2\_Weapons\_InviewFile.doc**” for more info)

### **[weapon].frames**

There is one “**.frames**” file for every weapon. In this file, note tracks can be set up to play specific sounds, listed in the sof2.inview file, on specific frames. There is also the ability to turn off surfaces on certain frames as well to efficiently keep polygon count down. (see “**SoF2\_Weapons\_FramesFile.doc**” for more info)

### **SOF2.wpn**

This file details weapon type, category, and effects for each weapon. (see “**SoF2\_Weapons\_WpnFile.doc**” for more info)