



**WEAPON .INVIEW FILE**

**BY RAVEN SOFTWARE**

## 1.0 Introduction

To find the sof2.inview file look in the folder where sof2.exe is installed (the default installation puts it in C:\Program Files\Soldier of Fortune II - Double Helix). Now look in the "base" folder there. You'll find several .pk3 files, one of which is named "therest.pk3". You can open this file with any program that handles .zip files. Inside therest.pk3 you'll see sof2.inview. To edit this for yourself, create a folder called "inview" in your base folder and extract sof2.inview. Using the default installation as an example, this would create C:\Program Files\Soldier of Fortune II - Double Helix\base\inview\sof2.inview.

Now when you run the game it will use this file instead of the one in therest.pk3. BE CAREFUL! If you accidentally introduce any errors into this new .inview file the weapons won't work in the game. If this happens you can always delete C:\Program Files\Soldier of Fortune II - Double Helix\base\inview\sof2.inview and the game will go back to using the original in therest.pk3.

Below is an excerpt from the sof2.inview file trimmed slightly for space. It defines the models, bolts, animations, and sounds for the M4 rifle with attached M203 grenade launcher.

```
weapon
{
    name          "M4"
    foreshorten    "0.6"

    viewoffset
    {
        forward    "0"
        right       "0"
        up          "0"
    }

    sounds
    {
        fire
        {
            sound1    "sound/weapons/m4/m4fire01"
            sound2    "sound/weapons/m4/m4fire02"
            sound3    "sound/weapons/m4/m4fire03"
        }
        altfire
        {
            sound1    "sound/weapons/m4/glaunch"
        }
        m4ready
        {
            sound1    "sound/weapons/m4/ready_gun"
        }
        .
        .
        .
    }

    forcefeedback
    {
        fire
```

```

        {
            force1          "fffx/weapons/m4/fire"
        }
        .
        .
    }

    weaponmodel
    {
        name      "mainwpn"
        model      "models/weapons/m4/m4.glm"
        frames     "skeletons/weapons/m4/m4.frames"

        buffer
        {
            name      "rifle buffer"
            model      "models/weapons/buffer/rifle/buffer.glm"
            bolttobone "gun"
            muzzle     "flash_m4"
        }

        hands
        {
            left
            {
                bolttobone "lhand_m4"
            }
            right
            {
                bolttobone "handle_m4"
            }
        }

        bolton
        {
            name      "m203"
            model      "models/weapons/m203/m203.glm"
            frames     "skeletons/weapons/m203/m203.frames"
            parent     "rifle buffer"
            bolttobone "m203_m4"

            rightside
            {
                surface1 "barrelfronth"
                surface2 "barrelbackh"
                surface3 "40mmgrenadeh"
                surface4 "grenadetriggerh"
                surface5 "griph"
                surface6 "latchh"
            }
        }

        rightside
        {
            surface1 "cliph"
            surface2 "gunmuzzleh"
            surface3 "frontsighth"
            surface4 "stocktubeh"
            surface5 "bodyfronth"
            surface6 "barrelh"
        }
    }

```

```

        surface7 "gunbodyh"
        surface8 "sighth"
        surface9 "gunstock"
    }
}

anim
{
    name    "idle"

    info
    {
        type    "weaponmodel"
        name    "mainwpn"
        anim    "m4idle"
        extra1  "m4fingerknuckles"
        extra2  "m4fingeradjust"
        speed   0.5

        mp_speed    0.3
    }

    info
    {
        type    "hands"
        name    "left"
        anim    "lm4idle"
        extra1  "lm4fingerknuckles"
        extra2  "lm4fingeradjust"
        speed   0.5
        lodbias 2

        mp_speed    0.3
    }

    info
    {
        type    "bolton"
        name    "m203"
        anim    "m203idle"
        speed   1

        mp_speed    0.3
    }
}

anim
{
    name    "fire"

    info
    {
        type    "weaponmodel"
        name    "mainwpn"
        anim1   "m4fire"
        anim2   "m4fire2"
        anim3   "m4fire3"
        anim4   "m4fire4"
        speed   1
    }
}

```

```

info
{
    type      "hands"
    name      "left"
    anim      "lm4fire"
    speed     1
    lodbias   2
}

info
{
    type      "bolton"
    name      "m203"
    anim      "m203idle"
    speed     1
}
}

```

The basic format of this file is the same as almost all of the external text data for SOF2 – a key followed by its corresponding value (e.g. name “M4”). Key names are case sensitive (use “name” not “Name”). A “block” refers to a group of data (key/value pairs and/or other blocks) contained within a pair of braces and given a name. The parent of a block is the block within which it is grouped. Child blocks are those that are grouped within another block. Some blocks are required for the weapon definition to be valid while some are optional.

Now let me digress long enough to provide some preliminary information on model surfaces and bolting models and effects together in SOF2.

## 1.1 Surfaces

A surface is simply part of the model. To be more specific, it is one or more polygons in the model’s mesh, or body, that can all be turned on or off simultaneously. For example, there’s a soldier model in the game that occasionally has a scarf. The scarf is its own surface (comprising several polygons) that can be turned off when we don’t want the soldier to wear a scarf and turned on when we do. Surfaces on weapon models may get turned off when that portion of the weapon isn’t in view. For example, the right side of the M4 is never in view so all of the surfaces associated with the model’s right side are turned off when the player uses it. This reduces the rendered poly count of the weapon, speeds up the game, and replenishes the ozone layer.

## 1.2 Boltons

Ghoul2 models have “bones” – nothing more than a point on the model that has a set of directions (forward, right, and up) associated with it. When we want an effect to be played in a certain place on a model – like, say, a muzzleflash on the end of a rifle barrel – we specify the model in question, a bone on the model, and the effect we want to play. The effect is given the orientation of the bone and is played wherever that bone happens to be. When we want to bolt a model onto another model we specify the base model, a bone on the model, and the model (“bolton”) we want to attach to the base. The origin of the bolton is then attached to the bone and given the orientation of the bone.

## 2.0 Blocks

### 2.1 The “weapon” block

Parent: none  
Required: **yes**  
Children: viewoffset, surfaces, sounds, forcefeedback,  
weaponmodel, anim  
How many might there be: exactly one

This is the definition of a single weapon. Within this block is all of the information pertaining to that weapon...what sounds it plays, animations it has, optional parts such as silencers and laser sights, extra models (boltons) that get attached to the main model, and miscellaneous other bits of info. Here are the keys found in the “weapon” block:

- **name** – this must match a weapon definition from the sof2.wpn file
- **foreshorten** – this is a number used to scale the long axis of the weapon model when it’s rendered so as to prevent in-view models from looking overly long and spindly. 0.6 is the standard value for all weapons with the exception of emplaced guns that the player can use. These have no foreshorten factor because the world model that the player sees isn’t foreshortened therefore the in-view weapon isn’t either.

#### 2.1.1 The “viewoffset” block

Parent: weapon  
Required: no  
Children: none  
How many might there be: zero or one

This almost doesn’t qualify as a block because it only contains three bits of information...the x, y, and z (or “forward”, “right”, and “up”) offsets that the camera will use for emplaced weapon models.

#### 2.1.2 The “surfaces” block

Parent: weapon  
Required: no  
Children: see below  
How many might there be: zero or one

This is a collection of blocks each of which is a list of surfaces that will be turned on/off as a result of a certain notetrack being hit in a weapon animation. For a proper explanation of notetracks, see the document named SoF2\_Weapons\_FramesFile. The RPG7, for example has in its surfaces block a “rocketoff” block which contains the keys “rockethead” and “rockettail”, the values of which are both “off”. This means that when an animation (it happens to be the “fire” anim) is played for the RPG7 model and the “rocketoff” notetrack is hit, the weapon model surfaces named “rockethead” and

“rockettail” will both get turned off. There’s a corresponding “rocketon” group that turns the surfaces back on when the RPG is reloaded.

```
surfaces
{
    rocketoff
    {
        rockethead    off
        rockettail    off
    }

    rocketon
    {
        rockethead    on
        rockettail    on
    }
}
```

### 2.1.3 The “sounds” block

Parent: weapon  
Required: no, but I bet you’d want one  
Children: see below  
How many might there be: zero or one

This is a collection of blocks each of which is a sound that will be played as a result of a certain notetrack being hit in a weapon animation. Looking at the “fire” sound in the above example you’ll see the name (“fire”) which is the notetrack that gets called during the weapon’s fire animation. Within the definition of the “fire” block you see three keys: sound1, sound2, sound3. The values for these keys are the filenames of sounds that can be played when this notetrack is hit. One of these three will be chosen at random every time the fire animation is played. The path for these filenames assumes that they are in the base folder and will use either mp3’s or wav’s.

### 2.1.4 The “forcefeedback” block

Parent: weapon  
Required: no  
Children: see below  
How many might there be: zero or one

Every time a sound is played via a notetrack an accompanying feedback effect, defined here, is also played. These are optional, though, since a FF mouse isn’t *required* to be able to play SoF2.

### 2.1.5 The “weaponmodel” block

Parent: weapon  
Required: **yes**  
Children: buffer, hands, bolton, rightside, leftside, front, optionalpart  
(see 2.2 for definitions)  
How many might there be: exactly one

Hoo, boy. This is where a whole bunch of magic happens. This block defines which model to use for the weapon, how to attach other models to it, what boltons to attach, various special surfaces on the models, and many other silly things. First, the keys:

- **name** – as it turns out, this field isn't terribly important for this block. It just indicates that this is the weapon model itself and not some Bolton
- **model** – this is the filename of the weapon model itself. The path for this filename assumes it's in the base folder. Only use Ghoul2 (.glm) files here.
- **frames** – this is the filename of the .frames file (see SoF2\_Weapons\_FramesFile) associated with the above .glm file. The path for this filename assumes it's in the base folder.

## 2.1.6 The “anim” block

Parent: weapon  
Required: at least the “idle” block  
Children: info (see 2.4 for definitions)  
How many might there be: one or more

Each anim block represents a different animation that the weapon might play. These blocks also define what the hand models and any boltons are doing during this anim.

- **name** – this name is used within SOF2 to play certain weapon animations at the appropriate times. For instance, when the weapon is supposed to be idle, SOF2 will attempt to play the weapon animation specified in the “idle” block. When firing a weapon it plays the anim specified in the “fire” block. In short, your weapon will only work correctly if you provide anim blocks with the correct, pre-defined names that are found in the SOF2.inview file.

## 2.2 Children of the ‘weaponmodel’ block

### 2.2.1 The “buffer” block

Parent: weaponmodel  
Required: **yes**  
Children: none  
How many might there be: exactly one

The buffer is an invisible model that attaches to the base weapon model. Yup, this is a bolton. The bones on the buffer are used as attachment points for the hand models and all other boltons used in conjunction with the weapon model.

- **name** – not necessarily unique, this identifies the type of buffer to use. All of SOF2's pistols, for instance, use the same buffer model. You can use any model you like, either an existing one from SOF2 or one you create. If you want to build a radically different pistol model such that you need new bones for attaching boltons or different effects then you'd have to build your own buffer model.
- **model** – this is the filename of the buffer model. The path for this filename assumes it's in the base folder. Only use Ghoul2 (.glm) files here.
- **bolttobone** – this is the bone on the gun model to which you want to bolt the buffer.



- **muzzle** – this is the bone on the buffer model where you want the muzzleflash to appear. Since buffers are attached to the world models used by NPCs, this is also where bullets and projectiles originate when fired by NPCs.

### 2.2.2 The “hands” block

Parent: weaponmodel  
 Required: **yes**  
 Children: left, right (although they hardly qualify as blocks)  
 How many might there be: exactly one

All we’re doing in this block is specifying where to attach the hands on the buffer model. In the “left” block the value associated with the “boltbone” key is the name of a bone on the buffer. Similarly, the “right” block also has the name of a bone. We bolt the left and right hand models, respectively, to the bones listed in these blocks.

### 2.2.3 The “rightside” block

Parent: weaponmodel  
 Required: no  
 Children: none  
 How many might there be: zero or one

This is just a list of surfaces that are turned on/off when the right side of the weapon is supposed to be turned on/off. These are provided purely for making the rendering of the weapon more efficient...you don’t have to break up your weapon models this way.

### 2.2.4 The “leftside” block

Parent: weaponmodel  
 Required: no  
 Children: none  
 How many might there be: zero or one

This is just a list of surfaces that are turned on/off when the left side of the weapon is supposed to be turned on/off. These are provided purely for making the rendering of the weapon more efficient...you don’t have to break up your weapon models this way.

### 2.2.5 The “front” block

Parent: weaponmodel  
 Required: no  
 Children: none  
 How many might there be: zero or one

This is just a list of surfaces that are turned on/off when the front side of the weapon is supposed to be turned on/off. These are provided purely for making the rendering of the weapon more efficient...you don’t have to break up your weapon models this way.

### 2.2.6 The “optionalpart” block

Parent: weaponmodel  
 Required: **yes**  
 Children: none  
 How many might there be: zero or more

Optional parts are things like the SOCOM pistol's laser sight, tactical light, and silencer. Each part represents one or more surfaces on the model that are grouped together and turned on or off at the same time.

- **name** – unique name of this part. SOF2 specially handles the names “lasersight”, “silencer”, “utl”, “slide\_back”, and “slide\_fwd”
- **bone** – the bone for a part is used to indicate a special location on the part, like the point from which a laser sight should emanate
- **surface1...surfaceX** – these are the names of all surfaces on the weapon model that comprise the part in question. An optional part of the model can have any number of surfaces associated with it.
- **muzzle** – you see this used instead of “bone” in the definition for the silencer. That's because this bone actually replaces the muzzle bone already named for the pistol model since this is where you want bullets and muzzleflashes to appear when the silencer is being used.

### 2.2.7 The “bolton” block

Parent: weaponmodel  
Required: no  
Children: leftside, rightside, front  
How many might there be: zero or more

A bolton block is just what it sounds like...a model that's bolted to the buffer model.

- **name** – uniquely identifies this bolton amongst others
- **model** -- this is the filename of the bolton model. The path for this filename assumes it's in the base folder. Only use Ghoul2 (.glm) files here.
- **frames** – this is the filename of the .frames file (see SOF2FramesFileDocs) associated with the above .glm file. The path for this filename assumes it's in the base folder.
- **parent** – name of the part of the weapon to which we're bolting this bolton. In the case of the M4 rifle, its M203 bolton is bolted to the buffer model so its parent is “rifle buffer”.
- **bolttobone** – the bone on the parent to which the bolton is attached.

## 2.3 Children of the “bolton” block

### 2.3.1 The “leftside” block

Parent: bolton  
Required: no  
Children: none  
How many might there be: zero or one

This is just a list of surfaces that are turned on/off when the left side of the bolton is supposed to be turned on/off. These are provided purely for making the rendering of the weapon more efficient...you don't have to break up your weapon models this way.

### 2.3.2 The “rightside” block

Parent: bolton  
Required: no  
Children: none

How many might there be: zero or one

This is just a list of surfaces that are turned on/off when the right side of the bolton is supposed to be turned on/off. These are provided purely for making the rendering of the weapon more efficient...you don't have to break up your weapon models this way.

### 2.3.3 The “front” block

Parent: bolton  
Required: no  
Children: none  
How many might there be: zero or one

This is just a list of surfaces that are turned on/off when the front side of the bolton is supposed to be turned on/off. These are provided purely for making the rendering of the weapon more efficient...you don't have to break up your weapon models this way.

## 2.4 Children of the “anim” block

### 2.4.1 The “info” block

Parent: anim  
Required: at least one for the main weapon model  
Children: none  
How many might there be: one or more

Within the definition of each animation we'll want to specify what each model and bolton should be doing. We do this by putting an info block in the anim definition. If there is no info block for, say, the left hand model within the idle anim then the left hand model won't be rendered for that anim. So if you don't want a model to show up at all during an animation, don't give it an info block within that anim block.

- **type** – which model are we talking about...“weaponmodel”, “hands”, or “bolton”
- **name** – further specifying which model to use. If the type is weaponmodel then the name needs to be “mainwpn”. If the type is hands then the name needs to be “left” or “right”. If the type is “bolton” then the name can be that of any of the boltons defined in the weaponmodel block.
- **anim** – this is one of the animation filenames specified in this model's .frames file (see SoF2\_Weapons\_FramesFile). It's not the entire path that you see in the .frames file, it's just the filename with the extension stripped. For example, the m4.frames file contains “s:/ani/base/models/weapons/m4/m4idle.xsi” as an idle animation. You would use “m4idle” in the anim field for the corresponding info block.
- **extra1...extraX** – these are optional idle anims that are played occasionally. Every several seconds there's a chance that, if the weapon is idling, one of these extra anims will be randomly selected. These are things like flipping the shotgun or twirling the pistol. Totally optional.
- **speed** – a number indicating how fast the animation should be played relative to the default rate of 20 frames per second (fps). If you want an idle animation to play half as fast, use a speed value of 0.5. This is also affected by the fps at which the animation was originally exported.

- **mp\_speed** – this is a different animation playback speed that will only be used in multiplayer. If this value isn't present then the weapon will animate at the same speed in multiplayer as it does in singleplayer.
- **lodbias** – since some animations require both hands to be in view (which means more polys) you can specify a number here (1 or 2...higher number means fewer polys) that will force a certain model into a lower level of detail for the duration of the animation. This renders fewer polys so your framerate won't be adversely affected when you use both hand models or wield dual weapons.
- **animNoLerp1...animNoLerpX** – when one of these is used the animation is played without lerping from frame to frame
- **extraNoLerp1...extraNoLerpX** – when one of these is used the optional idle animation is played without lerping from frame to frame.