Custom Animations Pt.01 (Raw Structures)

by **DemonicSandwich** » Nov 2009, Mon 16, 8:04 pm

Custom Animations Pt.01
(Raw Structures)

This topic is just for reference. I intend on making a topic on animations and explaining the raw structures *there*would clutter the topic.
But please, post your questions/ideas/whatever.

Things to note:

- When I am referring to a meta value, this value is always in the Animations reflexive unless otherwise stated.
- Not all values are mapped out or mapped out correctly. But there is enough to create a functioning animation.

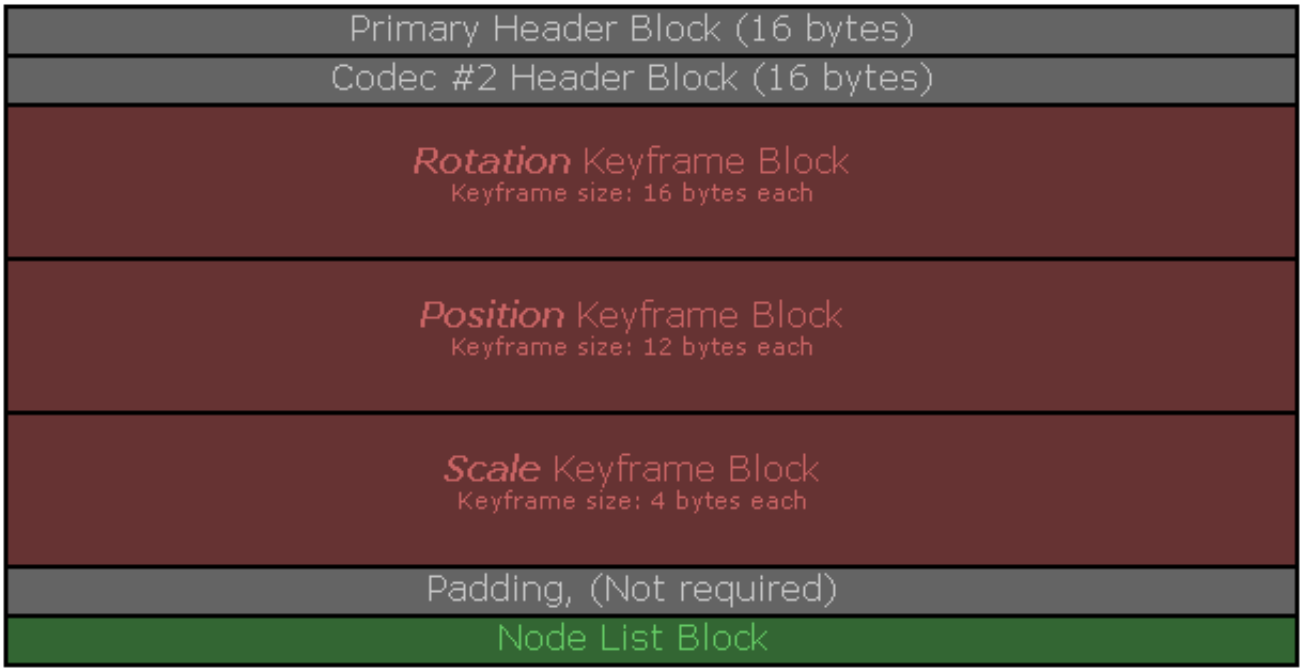
Compression Types: (Mapped out, Unknown)

- ***CODEC #0*** no_compression_codec AVERAGE KEY RATIOS Rot:1.00, Trs:1.00, Scl:1.00
- ***CODEC #1*** uncompressed_static_data_codec AVERAGE KEY RATIOS Rot:1.00, Trs:1.00, Scl:1.00
- ***CODEC #2*** uncompressed_animated_data_codec AVERAGE KEY RATIOS Rot:1.00, Trs:1.00, Scl:1.00
- ***CODEC #3*** 8byte_quantized_rotation_only_codec AVERAGE KEY RATIOS Rot:1.00, Trs:1.00, Scl:1.00
- ***CODEC #4*** byte_keyframe_lightly_quantized AVERAGE KEY RATIOS Rot:0.22, Trs:0.26, Scl:1.00
- ***CODEC #5*** word_keyframe_lightly_quantized AVERAGE KEY RATIOS Rot:1.00, Trs:1.00, Scl:1.00
- ***CODEC #6*** reverse_byte_keyframe_lightly_quantized AVERAGE KEY RATIOS Rot:0.27, Trs:0.64, Scl:0.12
- ***CODEC #7*** reverse_word_keyframe_lightly_quantized AVERAGE KEY RATIOS Rot:0.07, Trs:0.14, Scl:1.00
- ***CODEC #8*** blend_screen_codec AVERAGE KEY RATIOS Rot:1.00, Trs:1.00, Scl:1.00

==========

==========

Compression Format #2



Format 2 is identical to format 3 with the only difference being that Format 2's Rotation keyframes aren't compressed.
Format 2 also shares the same Pros/Cons of format 3.
See format 3 for more information.

Primary Header:

Same in all formats. See format 3.

Codec #2 Header:

- **Data End [unsigned integer]:**
  Defines the where the Keyframe data ends. This value is also in the meta and should be updated as this one is.
  This value is also just user reference as the animation will work without it being updated. However the value in the meta is used.

- **Rotated Keyframe Block Size [unsigned integer]:**
  Defines the total size off the Rotation Keyframe Block in bytes. *Size is [Keyframe count * Rotation Node count * 16]*

- **Position Keyframe Block Size [unsigned integer]:**
  Defines the total size off the Position Keyframe Block in bytes. *Size is [Keyframe count * Position Node count * 12]*

- **Scale Keyframe Block Size [unsigned integer]:**
  Defines the total size off the Scale Keyframe Block in bytes. *Size is [Keyframe count * Scale Node count * 4]*

Rotation Keyframe Block:

- **Keyframe:**

- I rotation **[float]**
- J rotation **[float]**
- K rotation **[float]**
- W scale **[float]**

Position Keyframe Block:

See format 3.

Scale Keyframe Block:

See format 3.

Padding:

See format 3.

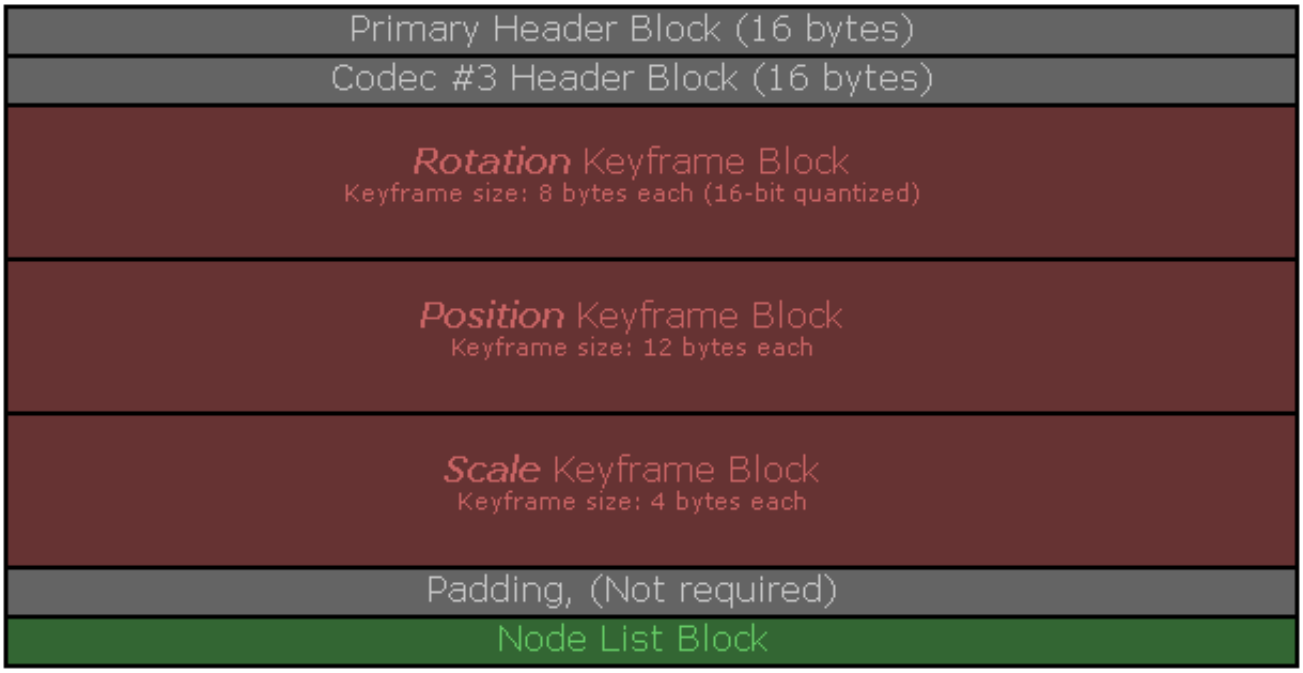
Node List Block:

See foramt 3.

==========

==========

Compression Format #3



Format 3 is one of 3 simple formats that have the bare minimum information to have a working animation as shown below.

Advantages:

This is a simple format, easy to work with for beginners.

Disadvantages:

It requires a key frame for every render frame so animating a single node for 1 second requires 30 keyframes. Since machines have a play speed modifier, you can create a simple machine animation with few keyframes.

Primary Header:

- **Codec [byte]:**
  Defines what compression format is used on the animation
- **Rotation Node Count [byte]:**
  Defines the number of nodes affected by Rotation. Purely user reference, has no affect on actual node counts.
- **Possition Node Count [byte]:**
  Defines the number of nodes affected by Possition. Purely user reference, has no affect on actual node counts.
- **Scale Node Count [byte]:**
  Defines the number of nodes affected by Scale. Purely user reference, has no affect on actual node counts.
- **Unknown [integer]:**
  Purpose still unknown. Leave it as is or make it zeros.
- **Playback Rate [float]:**
  What the name suggests. Haven't seen any significant affect when changed though. Leave it as is or make it 1.
- **Data Start (usually)[unsigned integer]:**
  It appears to define where the data starts. But will often reference the end of the data before the Node List.
  Not sure *what* it truly stands for.

Codec #3 Header:

- **Data End [unsigned integer]:**
  Defines the where the Keyframe data ends. This value is also in the meta and should be updated as this one is.
  This value is also just user reference as the animation will work without it being updated. However the value in the meta is used.
- **Rotated Keyframe Block Size [unsigned integer]:**
  Defines the total size off the Rotation Keyframe Block in bytes. *Size is [Keyframe count * Rotation Node count * 8]*
- **Position Keyframe Block Size [unsigned integer]:**
  Defines the total size off the Position Keyframe Block in bytes. *Size is [Keyframe count * Position Node count * 12]*
- **Scale Keyframe Block Size [unsigned integer]:**
  Defines the total size off the Scale Keyframe Block in bytes. *Size is [Keyframe count * Scale Node count * 4]*

Rotation Keyframe Block:

- **Keyframe:**
  - **I rotation [short]**
  - **J rotation [short]**
  - **K rotation [short]**
  - **W scale [short]**

The rotations in this format are 16-bit quantized. For the common man, you do the following:
To Compress: (uncompressed float * 32,767 = compressed short)
To Decompress: (compressed short / 32,767 = uncompressed float)

An interesting little aspect of the rotations, Animations in Halo 2 actually use the inverse values. So when you're using a quats calculator to get your quaternions, make sure to reverse their signs before inserting them into your raw.

Position Keyframe Block:

- **Keyframe:**
  - **X position [float]**
  - **Y position [float]**
  - **Z position [float]**

Scale Keyframe Block:

- **Keyframe:**
  - **Scale [float]**

I've only seen this used on first person animations, only on the camera node, and always 1.
When I attempted to use this on an object, it self deleted.

There can be multiple blocks of one type depending on the number of nodes ticked.
Each block of animations will be assigned to the selected nodes in numerical order.
So if you have Rotation Nodes 2, 7, and 9 selected, there will be 3 blocks or Rotation Keyframes and will be applied to the nodes in numerical order with the first block affecting 2, the next affecting 7, and the next 9.

Then the Position animation blocks after those, also in numerical order and so on.

Padding:

This is optional. I've seen several animations with in raw padding, yet none of them fucking needed it.
This padding will always come imediatly before the Node List Block. It's size is determined by the meta value**Internal Padding Size** [byte, offset: 60].
Setting this value to 0 makes this padding block disappear so do that.

Node List Block:

Just a block of flags. Tick a flag and the respective node will be affected by animation.
The size of this block should always be a multiple of 12. It's size is determined my the meta value Node List Size [byte, offset: 61]
When set to 12, the list can handle objects with a node count up to 32 (0-31).

When set to 24, the object can have 64 nodes and so on.
The block is split into 3 groups of flags.
The first group determines what nodes are affected by rotation, the second group for position, and the third group for scale.

If looking at it in hex, the Node ticks for each group will be in order as follows:
[7][6][5][4][3][2][1][0] - [15][14][13][12][11][10][9][8] - etc.
Each flag corresponding to a Node index.

==========

==========

Compression Format #6

Format 6 uses Keyframes the way there are supposed to be used. As *KEY* frames, with the majority of the frames being Tweens.

This format adds two extra blocks of data to it's structure.
One block that determines how many Keyframes each Node will have, and an offset to to where it's Markers start from.

Advantages:

This format requires far fewer Keyframes to make a complex animation.
You do not need a keyframe for each render frame.

Disadvantages:

It's a bit more complex to work with.
Since it's Keyrame Markers are only 1 byte in size, you're animation cannot be longer than 256 frames, or ~8.5 seconds for non-machine objects.
Machines are still limited to 256 frames but the frames can be stretched out.

--- will add structure later ---

Model Customization Pt.01 | Model Customization Pt.02 | Bipd Attachments | True Marker Rotations
"I'm the h4x man! Skibby Dibby Dib YoDahDubDub, YoDahDubDub"

DemonicSandwich
Trollwich

Posts: 1620
Joined: Dec 2007, Sat 08, 2:47 pm
Location: I...huh...I don't really know. x.x