

FS2x XML Gauge Tutorial 2:



Vertical Speed Indicator

By Don Kuhn

The FS9 Default Beechcraft Baron 58 Vertical Speed Indicator



Figure 1

Welcome to the first fs2x XML gauge making tutorial entitled 'Vertical Speed Indicator'. In this tutorial I will be making a VSI based on the default Baron 58 VSI shown above. I chose to do the VSI first because it is a common gauge, it is functionally simple, and it requires no input by the user. The idea is to go step-by-step in explaining how I would go about making this XML gauge. Keep in mind that my way is not the only way, and maybe not even the best way, to make this gauge.

I will be referring to the various sections of the Baron, such as coding and bitmaps, as examples for clarity sake. It would be helpful if you prepared for this by unzipping the Beech_Baron.cab file located in your main FS9/Gauges folder. This will give you a folder in the same location that has all of the relevant files and bitmaps.

Don't forget to read the first XML gauge making tutorial by Nick Pike!

Now, being a simple minded person I will approach this task in a simple way by first asking 3 simple questions.

1. Functions: what does the gauge do?
2. Components: what are the components of the gauge that need to be made?
3. XML Coding: what code do I need to make the gauge work as it should?

Looking at each of these in detail:

Functions

As the name implies, the vertical speed indicator simply indicates the vertical speed of the aircraft. The default Baron VSI is a common type whereby the vertical speed is indicated by moving one end of a needle over a fixed linear value scale. The needle moves in an arc and its movement is calibrated to point to the position on the fixed scale corresponding to the aircraft vertical speed in feet per minute.

The VSI fixed scale in the Baron is partitioned in units of 100 ft/min up to a maximum of 2000 ft/min ascending or descending with large hash marks every 500 ft/min.

Components

So what components do we need? The first thing we need is the gauge 'body', or background bitmap. Since the display area and the needle are 'sunk' into the gauge body, we will need to emulate this in our gauge. We will also need a scale presented as an arc in the display area, a bit of info, and last we will need a needle.

If you look in the 640 or 1024 folder located in the ...\\Gauges\\Beech_Baron folder you will see that MS has used 2 bitmaps, Vertical_Speed_Background.bmp and Vertical_Speed_Needle.bmp, to portray this VSI. In general, it is best to put all unchanging components on the background bitmap to get the least number of bitmaps possible. In some cases, like a sophisticated MFD or GPS, the information displayed in the display area can be changed and therefore cannot be included in the background bitmap.

[**Note:** Remember to save your bitmaps without flattening the layers. This way you can easily make changes or use a particular layer for multiple gauges.]

Let's start making the VSI by making the Vertical_Speed_Background.bmp. My approach is to first make the gauge 'body' and then overlay this with the display area containing the scale, info, and needle.

Background bitmap

Step1: Background layer

The size of the Vertical_Speed_Background.bmp in the 1024 folder of the default Baron is 156 x 157 pixels, and since we are making the same VSI we will use this size for the

gauge body. We will make the size 156 x 157 and then resize to 99 x 99 pixels to put in the 640 folder. Set the background layer color to 100% black.

[Notes: Please read the section ‘Bitmaps in FS2004’ at the end of this tutorial for more information. Also, if I was making a new VSI I would go to the manufacturer’s website, find out the dimensions of the particular gauge, and then use a pic of the real panel to scale the gauge to the correct pixel size.]

Step 2: Make VSI ‘body’

Make a gray circle, closest to the default color as possible, a size such that the perimeter of the circle just touches the edges of the background layer on all 4 sides (figure 2).



Figure 2

Step 3: Make tab and shade (or blend)

Make a rectangle with rounded corners, rotate it so it fits in the lower left corner, and then remove what is not necessary (figure 3). Make a small black hole for the ‘set knob’ and position it in the center of the tab (figure 3). Now give it a 3D look by shading or blending; note the light on the upper left part of the gauge in figure 3.



Figure 3

Step 4: Display area background

Make a 98% black circle to a size that will cover all of the gray circle except what you want to leave as a ‘rim’. Next make a gray circle (darker gray than the lower gray circle) to a size that will cover ~97% of the black circle (figure 4). This will help to give a ‘sunken’ appearance.



Figure 4

Step 5: Put on scale and numbers

This step is probably the most difficult because it is tedious business. The hash marks need to be positioned correctly, and most are angled. Plus they're not all the same size.

[**Note:** You can use a real bitmap as a 'tracer template' by overlaying your bitmap with the real one, and then making the real bitmap 15-25% transparent. Just remove when finished.]

[**Note:** Usually you can use the symmetry of a gauge to cut your workload. For example, divide your gauge into 4 quadrants, make one set of hash mark for one quadrant and then copy, paste, flip, and position it in the other quadrants. Or make 2 if that is all you can. An approach to take if all hash marks are evenly distributed around the circle is to make a series of lines across the face of the gauge, like the spokes of a bicycle wheel, and then erase all but what is needed as the hash marks. Use this as a template for future gauges.]

I usually start this task by making and positioning the larger hash marks, and then doing the same with the smaller hash marks. Next put in the numbers (figure 5).



Figure 5

Step 6: Make final additions

The last additions are straightforward. Make text the appropriate size and position it where it should go. Make an arrow and position it in the up or down position as shown in figure 1. Slant the arrow, copy and paste it, then flip it vertically and move it to the correct position. Add the screws by making a small dark gray circle, blend it to give a

‘raised’ look, and then place a black x in the center to emulate the screwdriver grooves (figure 6).

At this point your background bitmap should be completed and look something similar to the one in figure 6 and the default Baron. It is easy to miss something, so double check to make sure that hash marks are angled and positioned where they should be, and that everything is sized and positioned correctly. Now save this bitmap in ...Gauges\Beech_Baron\1024 as VSI_Background_test.bmp. Next re-size this bitmap to 99 x 99 pixels and place it in theGauges\Beech_Baron\640 folder.



Figure 6

[Note: The cyan colored lines across the face of the background bitmap will be used later to center the needle in the display area when coding. You do not have to include these lines at this step; I simply left them in to refer to later.]

VSI Needle

The way I look at this approach is that I will need a white and a gray rectangle, a white triangle, several closed circles, and a small open dark circle with a very small black circle inside. Since we are making the Baron VSI, you can get the dimensions of the rectangles from the default bmp. For example, you can see that the gray rectangle covers approximately the first 34 pixels (from the left) out of 85. The pointed end of the needle, made from a rotated triangle covers about 5 pixels, making the white rectangle ~46 pixels. Notice that the needle in the bitmap does not point to the left as in the finished gauge.

Step 1: Background layer

Looking at the Vertical_Speed_Needle.bmp located in the ...Gauges\Beech_Baron\1024 folder we see that the bitmap is 85 x 11 pixels. It is 53 x 9 pixels in the 640 folder, and as pointed out above you can re-size the 85 x 11 bmp when finished.

Step 2: Make the needle

One way to make the shaft of the needle is to make one long rectangle covering the length of the background. Make it all one color and then 'paint' the appropriate fraction of the needle the other color. Or make another rectangle of the appropriate size and overlay it onto the lower rectangle of different color. Next make a white triangle of the appropriate size, rotate it 90 degrees, and position it at the right end of the needle shaft (figure 7).



Figure 7

Step 3: Add circles and screws

The next things to add are the 2 circle areas of the gray part of the needle. Make a gray circle the same color as used to make the rectangle, to the appropriate size. Position it so it cover the first ~30% of the gray rectangle. Now copy, paste, and move the new circle to the leftmost end of the needle. Next draw a dark open circle and position it top center of the right most circle (figure 8).

Save this bitmap inGauges\Beech_Baron\1024 as VSI_Needle_test. Next re-size this bitmap to 53 x 9 pixels and place it in theGauges\Beech_Baron\640 folder.



Figure 8

Your VSI needle bitmap should now be completed and look like the one in figure 8 or the one in the default Baron 1024 folder.

Coding

Please read Nick Pike's first XML tutorial on the basics to better understand the structure and meaning of the code.

Now that the 2 bitmaps are made and placed in the default Baron 640 and 1024 folders, we can move on to the coding. Since we have made the default Baron VSI, we will use the default Baron Vertical Speed Indicator.XML document located in the ...Gauges\Beech_Baron folder.

We need to make 2 changes in this document as shown below, so remember to save.

```
<Gauge Name="Vertical Speed Indicator" Version="1.0">
  <Image Name="VSI_Background_test.bmp" ImageSizes="99,99,156,157"/> (A)
  <Element>
    <Position X="51" Y="49"/>
```



```
<Image Name="VSI_Needle_test.bmp" ImageSizes="53,9,85,11"> (B)
  <Axis X="15" Y="4" PointsTo="East"/>
</Image>
```

If you recall we named our bitmaps **VSI_Background_test.bmp** and **VSI_Needle_test.bmp**. We now need to change the coding so that our bitmaps are displayed instead of the default bitmaps. You can see in the default code that we are displaying 2 bitmaps named **Vertical_Speed_Background.bmp** and **Vertical_Speed_Needle.bmp**. These need to be changed as shown above in 'gold' lettering to point the sim to our bitmaps instead of the defaults.

What we want to do with the coding is to tell the sim that we have a bitmap named **VSI_Background_test.bmp** that we want to display at a specific position (A). We also want it to display a bitmap named **VSI_Needle_test.bmp** on top of the **VSI_Background_test.bmp** at a specific position and orientation (B). We then want the sim to rotate the **VSI_Needle_test.bmp** through an arc over the fixed scale according to 1/100th of the vertical speed of the aircraft to a maximum of +/- 2000 ft/min (C).

```
<Rotate>
  <Value Minimum="-20" Maximum="20">(A:Vertical speed,feet per minute) 100
/</Value>
  <Nonlinearity>
    <Item Value="-20" X="92" Y="54"/>
    <Item Value="-5" X="17" Y="72"/>
    <Item Value="0" X="9" Y="49"/> (C)
    <Item Value="5" X="17" Y="24"/>
    <Item Value="20" X="92" Y="43"/>
  </Nonlinearity>
  <Failures>
    <GAUGE_VERTICAL_SPEED Action="Freeze"/>
    <SYSTEM_PITOT_STATIC Action="0"/>
  </Failures>
</Rotate>
```

All we had to do with the coding here is to change 2 lines. If we had to position the **VSI_Background_test.bmp** ourselves we would need to know where the center of the bitmap was. In cases like this I find it useful to make some cyan cross lines to pinpoint the center location as shown in **Figure 6**.

Bitmaps in FS2004

FS2004 does not require bitmap size information. For example, the line
 <Image Name="VSI_Needle_test.bmp" ImageSizes="53,9,85,11">
 could become,
 <Image Name="VSI_Needle_test.bmp">

The bitmap sizes are given in this tutorial because they are given in the default gauge. This is a bit of a leftover from previous versions of FS where sizes were required. Giving the two sizes for 640 and 1024 folders is actually theoretically correct. The idea of the 640 bitmaps was to make gauges look reasonable on small computer monitors using lower resolutions (probably 640x480). When previous versions of FS were released, monitor sizes were smaller. I believe that the monitor size that the smaller bitmaps catered for was 15" and less. These days, the vast majority of monitors are larger than this using higher resolutions, so the bitmap sizes used in the 1024 folder are satisfactory. Without bitmap size information, FS would take the bitmap in this line

<Image Name="VSI_Background_test.bmp"/> and use it to set up the gauge size, and use it as a sort of painting canvas for applying further bitmaps, like the needle for example. Any positional information, for the needle, would be taken from the top left corner of the background bitmap and the X/Y co-ordinates are the number of pixels on this bitmap. Please note that any added bitmaps (needle) have to be made to the correct relative size.

There is a slight complication you need to understand in the default gauge coding. The positional and axis values in this default gauge are given relative to the smaller bitmap sizes, and FS automatically changes these values for the larger bitmap if using higher monitor resolutions. If bitmap size information is not given, then the positional and axis values would have to be relative to the bitmap sizes used.

Well, that's it! If all went well your 'new' VSI is displayed in the default Baron 58 and is working properly. My finished product is shown below in figure 9.



Figure 9

Don Kuhn
June, 2005
fs2x.com