I'm opening an image that doesn't require thumbnails to be shown and is smaller than the window. This causes 3/4 divide by ZERO errors for various reasons. I appreciate that this is kind of the opposite to how the control is supposed to be used though.

- In RasterThumbsCtrl::CalcScale:

```
int RasterThumbsCtrl::CalcScale(int imScale, int rasterWidth, int maxPageHeight)
{
 // calculates scale factor based on max thumb width
 // slightly smaller than ctrl width
 int maxScaledWidth = iscale(GetSize().cx, THUMBS_HSIZE_MUL, THUMBS_HSIZE_DIV);
 return iscale(maxScaledWidth, 1000, rasterWidth);

} // END RasterThumbsCtrl::CalcScale()
```

With no thumbnails GetSize().cx is 0 since you have previously set the splitter pos to 0. imageScale is then set to 0 and causes a DIVIDE_BY_ZERO. Adding the following after the call to CalcSize in RasterBaseCtrl::Layout

```
 if (!imageScale) {
  inside = false;
  return;
 }
```

fixes it.

- Also RasterBaseCtrl::Layout. You need !0 checks for vScollMax and hScrollMax on the two lines that do:
hScrollBar.Set(iscale(hScrollPos, scaledRasterWidth, hScrollMax)); and the vertical equivalent. I don't really understand what's going on here, but I can't see how this could ever work since vScrollMax should always start as 0.

- RasterBaseCtrl::PaintCache. Add a check for imageScale == 0 at the start of the function.

- ImageCache::Fill, caused when you resize the window down to nothing. Change first line to:
 if(imageBuffer.IsEmpty() || !imageBuffer.GetSize().cx || !imageBuffer.GetSize().cy)

Then everything works very nicely .

You also may want to investigate Upp::Scroller. Not sure if this'll help in your case but it's worth a look.

James