HOWTO: Burn Annotations into a Bitmap

These code snippets show how to burn a one or more annotations into a Bitmap. Note that the Bitmap object should be 24bpp and should have a defined resolution.

C#

```
rivate void BurnAnnotationsIntoBitmap(AnnotationData data, Bitmap bmp, PointF resolution, AnnotationUnit units) { LayerData ld = data as LayerData; if (ld != null) { foreach (AnnotationData item in ld.Items) { BurnAnnotationsIntoBitmap(item, bmp, resolution, units); } } else { IAnnotationRenderer renderer = Atalasoft.Annotate.Renderer.AnnotationRenderers.Get(data.GetType()); BurnAnnotationIntoBitmap(data, renderer, bmp, resolution, units); } } private void BurnAnnotationIntoBitmap(AnnotationData data, IAnnotationRenderer renderer, Bitmap bmp, PointF resolution, AnnotationUnit units) { RenderEnvironment re = new RenderEnvironment(new AnnotationImage(bmp), resolution, units, null); renderer.RenderAnnotation(data, re); } // Note: the above methods do the basic work of rendering onto a bitmap, but there's a bit more prep work needed to get the expected results. Specifically, the offset of the annoation from the top left corner of the viewer is not taken into account when dealing with single annotations. The following method will return a system.drawing.bitmap that represents a single annotation at the desired resolution private Bitmap AnnotationToBitmap(AnnotationUI targetAnnotation, Dpi resolution) { // get the annotationData object from the desired annotation object // if we don't clone it, we'll be altering the annotation in the viewer... not good. AnnotationData annoData = (AnnotationData)targetAnnotation.Data.Clone(); // We need to calculate the difference between location and bounds and track it down for later use PointF delta = new PointF((annoData.Location.X - annoData.Bounds.X), (annoData.Location.Y - annoData.Bounds.Y)); // Setting the location to this will ensure that the WHOLE annotation will be in "view" of the renderer annoData.Location = delta; // Since AtalaImage needs ints, but annotations could be floats, we're going to round up // to the nearest int and provide the desired sizes, padded with the delta amount int desiredWidth = (int)Math.Round(annoData.Bounds.Size.Width, 0) + (int)Math.Round(delta.X, 0); int desiredHeight = (int)Math.Round(annoData.Bounds.Size.Height, 0) + (int)Math.Round(delta.X, 0); // Directly make a bmp from an AtalaIamge so we can set width, height, pixel format and background color // Could do this directly with a new System.Drawing.Bitmap, but that doesn't allow background color to be set Bitmap bmp = new AtalaImage(desiredWidth, desiredHeight, PixelFormat.Pixel24bppBgr, Color.White).ToBitmap(); // Convert incoming Atalasoft.Imaging.Dpi resolution into PointF for use in burning PointF res = new PointF((float)resolution.X, (float)resolution.Y); // A good default to start with ... we need to fully qualify Atalasoft.Annotate.AnnotationUnit resUnit = Atalasoft.Annotate.AnnotationUnit.Pixel; // Directly modifies the passed in bitmap (bmp) by overlaying the graphics BurnAnnotationsIntoBitmap(annoData, bmp, res, resUnit); return bmp; }
```

Original Article:

Q10169 - HOWTO: Burning Annotations into a Bitmap

Atalasoft Knowledge Base
https://www.atalasoft.com/kb2/KB/50285/HOWTO-Burn-Annotations-into-a-Bitmap