HOWTO: Efficiently Process Images as they are Saved Into a PDF

Sometimes you are trying to create a PDF from many images and you need to process them before they are saved. PdfEncoder is written so that it can keep only one Image at a time in memory as it is building up the PDF Stream. This article tells you how you can hook into the process to change the image if you need to.

To be able to process images you need to use the variant of PdfEncoder.Save() that uses an ImageSource. ImageSource is designed for you to subclass, and it gives you the hooks you need to do this. Here is the main function to read in list of files and put all of their frames into a PDF using a custom ImageSource and a SetEncoderCompression event to set the compression.

C#

```
ublic void SaveToPdf(string[] imgs, string outputFile) { ImageSource pdfpages = new
ProcessImageFSImageSource(imgs, true); using (FileStream fs = new FileStream(outputFile,
FileMode.Create, FileAccess.Write)) { PdfEncoder pdfEncoder = new PdfEncoder(); // this lets
me set the compression type of the image pdfEncoder.SetEncoderCompression += new
EncoderCompressionEventHandler(pdf_SetEncoderCompression); pdfEncoder.Save(fs, pdfpages,
null); } } ProcessImageFSImageSource is a new ImageSource -- in this case, I'm going to base
it on FileSystemImageSource. Here it is public class ProcessImageFSImageSource :
FileSystemImageSource { public ProcessImageFSImageSource(string[] files, bool doAllFrames) :
base(files, doAllFrames) { } // override the function that is called to get images and change
them if necessary protected override ImageSourceNode LowLevelAcquire(int index) {
ImageSourceNode node = base.LowLevelAcquire(index); if (node != null && node.Image != null) {
// here you can alter node.Image to be a new // image using any DotImage function } return
node; } } And here is the event handler we added to SetEncoderCompression // (called by
PdfEncoder's SetEncoderCompression event) // this lets me set the compression type of the
image void pdf_SetEncoderCompression(object sender, EncoderCompressionEventArgs e) { // e
will have information about the Image that // you can use to decide the right compression //
here we just always use CcittGroup4 (which is only for 1bit images) e.Compression = new
PdfCodecCompression(PdfCompressionType.CcittGroup4); }
```

VB.NET

```
ublic Class ProcessImageFSImageSource Inherits Atalasoft.Imaging.FileSystemImageSource Public
Sub New(ByVal files As String(), ByVal doAllFrames As Boolean) MyBase.New(files, doAllFrames)
End Sub ' override the function that is called to get images and change them if necessary
Protected Overloads Overrides Function LowLevelAcquire(ByVal index As Integer) As
Atalasoft.Imaging.ImageSourceNode Dim node As Atalasoft.Imaging.ImageSourceNode =
MyBase.LowLevelAcquire(index) If node IsNot Nothing AndAlso node.Image IsNot Nothing Then '
here you can alter node.Image to be a new ' image using any DotImage function End If Return
node End Function End Class Public Sub SaveToPdf(ByVal imgs As String(), ByVal outputFile As
String) Dim pdfpages As Atalasoft.Imaging.ImageSource = New ProcessImageFSImageSource(imgs,
True) Using fs As New System.IO.FileStream(outputFile, System.IO.FileMode.Create,
```

HOWTO: Efficiently Process Images as they are Saved Into a PDF

```
System.IO.FileAccess.Write) Dim pdfEncoder As New Atalasoft.Imaging.Codec.Pdf.PdfEncoder() '
this lets me set the compression type of the image AddHandler
pdfEncoder.SetEncoderCompression, _ AddressOf pdf_SetEncoderCompression pdfEncoder.Save(fs,
pdfpages, Nothing) End Using End Sub Private Sub pdf_SetEncoderCompression(ByVal sender As
Object, ByVal e As Atalasoft.Imaging.Codec.EncoderCompressionEventArgs) ' e will have
information about the Image that ' you can use to decide the right compression ' here we just
always use CcittGroup4 (which is only for 1bit images) e.Compression = New
Atalasoft.Imaging.Codec.Pdf.PdfCodecCompression(Atalasoft.Imaging.Codec.Pdf.PdfCompressionType.CcittGroup4)
End Sub
```

Original Article:

Q10193 - HOWTO: How to efficiently process images as they are saved into a PDF

Atalasoft Knowledge Base

https://www.atalasoft.com/kb2/KB/50264/HOWTO-Efficiently-Process-Images-as-...