

HOWTO: Safely Change / Set Resolution of PdfDecoder

When you add a PdfDecoder to your RegisteredDecoders.Decoders collection, you are modifying the static collection where all of your DotImage components look when they want to know how to render a given image type. If you simply add a default PdfDecoder like this:

C#

```
Atalasoft.Imaging.Codec.RegisteredDecoders.Decoders.Add(new  
Atalasoft.Imaging.Codec.Pdf.PdfDecoder());
```

VB.NET

```
Atalasoft.Imaging.Codec.RegisteredDecoders.Decoders.Add(new  
Atalasoft.Imaging.Codec.Pdf.PdfDecoder())
```

There are two things you should be aware of:

- 1) The default resolution is that of your system - which is usually 96 DPI - this is generally far too low quality for printing and editing.
- 2) You're "blindly" adding a decoder to the collection without checking if there's already one there. If you add multiple PdfDecoder objects, the "extra" ones won't get picked up because the way our decoders work, it stops once it finds the first instance of a decoder that works.

So, what we're left with is that if you want to set the resolution when you add (issue 1) you can do it easily:

C#

```
Atalasoft.Imaging.Codec.RegisteredDecoders.Decoders.Add(new  
Atalasoft.Imaging.Codec.Pdf.PdfDecoder() {Resolution = 200 } );
```

VB.NET

```
Atalasoft.Imaging.Codec.RegisteredDecoders.Decoders.Add(new  
Global.Atalasoft.Imaging.Codec.Pdf.PdfDecoder() With { .Resolution = 200 } )
```

This is great, and it gets you a desired higher resolution (suggested values are 200 or 300... any higher and you significantly increase memory and/or file size without really gaining much in quality, any lower and the quality is generally noticeably poor) .. however, it

HOWTO: Safely Change / Set Resolution of PdfDecoder

doesn't address item 2

So, to properly, safely add a PdfDecoder (only if needed) or modify the already present PdfDecoder to use a new resolution, you can use this approach:

C#

```
staticreadonlyobject pdfLock = newobject();

privatestaticvoid SafelySetPdfDecoderResolution(int newRes)
{
    lock (pdfLock)
    {
        foreach (ImageDecoder rawDecoder
inRegisteredDecoders.Decoders)
        {
            if (rawDecoder isPdfDecoder)
            {
                ((PdfDecoder)rawDecoder).Resolution = newRes;
                return;
            }
        }

        RegisteredDecoders.Decoders.Add(newPdfDecoder() { Resolution
= newRes });
    }
}
```

VB.NET

HOWTO: Safely Change / Set Resolution of PdfDecoder

'TODO: lock is not supported at this time

For Each rawDecoder As ImageDecoder In RegisteredDecoders.Decoders

 If (TypeOf rawDecoder Is PdfDecoder) Then

 CType(rawDecoder, PdfDecoder).Resolution = newRes

 Return

 End If

Next

RegisteredDecoders.Decoders.Add(New PdfDecoder() With { .Resolution = newRes })

End Sub

NOTE: the return statement inside the loop properly exists the whole safelySetPdfDecoderResolution method after having updated the decoder.. it does not ever hit the final RegisteredDecoders.Add statement unless no PdfDecoder was found and modified.

This is the reason we suggest this method.. because it safely adds ONLY IF needed otherwise it modifies the existing decoder

Original Article:

Q10433 - HOWTO: Safely Change / Set Resolution of PdfDecoder

Atalasoftware Knowledge Base

<https://www.atalasoftware.com/kb2/KB/50067/HOWTO-Safely-Change-Set-Resolution-o...>