HOWTO: Safely Change / Set Resolution of OfficeDecoder

When you add an OfficeDecoder 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 OfficeDecoder like this:

C#

```
talasoft.Imaging.Codec.RegisteredDecoders.Decoders.Add( new
Atalasoft.Imaging.Codec.Office.OfficeDecoder());
```

VB.NET

```
lobal.Atalasoft.Imaging.Codec.RegisteredDecoders.Decoders.Add( New
Global.Atalasoft.Imaging.Codec.Office.OfficeDecoder())
```

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 OfficeDecoder objects, the "extra" ones wont 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#

```
talasoft.Imaging.Codec.RegisteredDecoders.Decoders.Add( new
Atalasoft.Imaging.Codec.Office.OfficeDecoder() { Resolution = 200 } );
```

VB.NET

```
lobal.Atalasoft.Imaging.Codec.RegisteredDecoders.Decoders.Add( new
Global.Atalasoft.Imaging.Codec.Office.OfficeDecoder() 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 doesn't address item 2

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

C#

```
tatic readonly object officeLock = new object(); private static void
SafelySetOfficeDecoderResolution(int newRes) { lock (officeLock) { foreach (ImageDecoder
rawDecoder in RegisteredDecoders.Decoders) { if (rawDecoder is OfficeDecoder) {
((OfficeDecoder)rawDecoder).Resolution = newRes; return; } }
RegisteredDecoders.Decoders.Add(new OfficeDecoder() { Resolution = newRes }); } }
```

NOTE

Please note that OfficeDecoder requires a license for Atalasoft OfficeDeocoder add-on. Also, in order to use the OfficeDecoder, you need to add the correct PerceptiveFilter DLLs to your project (correct in terms of ensuring that you use the DLLs whose "bitness" matches that of your CPU Target (x86 for 32 bit applications, or x64 for 64 bit applications) Please see the OfficeDecoder section of INFO: Changes Introduced in 10.7 for more details

Original Article

Q10439 - HOWTO: Safely Change / Set Resolution of OfficeDecoder

Last Update

2026-10-09 - TD - (Code Fencing Fixes)