



RasterEdge.XDoc.DICOM for .NET SDK Developer's Guide

RasterEdge.DocImagingSDK 9.8.7

2016-07-29

Table of Contents

| | |
|--|----|
| Getting Started | 1 |
| System Requirements for .NET | 1 |
| Supported Operating System | 1 |
| Development Environments | 1 |
| .NET Framework versions supported | 1 |
| Reference RasterEdge.XDoc.DICOM in .NET project | 2 |
| Necessary Libraries | 2 |
| Add References | 2 |
| FAQ | 4 |
| Errors On Visual Studio | 4 |
| Errors On IIS | 4 |
| Feature List | 6 |
| DICOM Generator | 6 |
| DICOM Document | 6 |
| DICOM Page | 6 |
| Annotations | 6 |
| Save | 6 |
| Programmer Guide | 7 |
| DICOM Object Generator | 7 |
| Create DICOM Document object from file path/data | 7 |
| Create DICOM Document object from images | 8 |
| Working with Document | 10 |
| Get Page Count | 10 |
| Get Document Type | 10 |
| Extract DICOM Page(s) to DICOM file/stream | 11 |
| Get a Particular Page | 12 |
| Convert to Images | 13 |
| Convert To Document | 15 |
| Working with Page | 18 |
| Get Frame Width/Height | 18 |
| Convert to Image | 19 |
| Working with Annotation | 24 |
| Add annotation | 24 |
| Add Image Annotation | 26 |
| Save | 27 |

Getting Started

System Requirements for .NET

Supported Operating System

The following Microsoft Windows operating systems are supported:

- Microsoft Windows XP Home Edition
- Microsoft Windows XP Professional Edition
- Microsoft Windows XP Professional x64 Edition
- Microsoft Windows 2003 Server
- Microsoft Windows 2008 Server R2
- Microsoft Windows Vista
- Microsoft Windows Vista x64 Edition
- Microsoft Windows 7
- Microsoft Windows 7 Enterprise x64 Edition
- Microsoft Windows 7 Professional x64 Edition
- Microsoft Windows 2012 Server x64 Edition

Development Environments

You can use RasterEdge.XDoc.DICOM for .NET to develop applications in any development environment that targets the .NET platform, but the following environments are explicitly supported:

- Microsoft Visual Studio 2005
- Microsoft Visual Studio 2008
- Microsoft Visual Studio 2010
- Microsoft Visual Studio 2011
- Microsoft Visual Studio 2012
- Microsoft Visual Studio 2013
- Microsoft Visual Studio 2015

.NET Framework versions supported

The following .NET Framework versions are supported:

- .NET Framework 2.0
- .NET Framework 3.0
- .NET Framework 3.5

.NET Framework 4.0
.NET Framework 4.5
.NET Framework 4.5.1
.NET Framework 4.5.2
.NET Framework 4.6

Reference RasterEdge.XDoc.DICOM in .NET project

Necessary Libraries

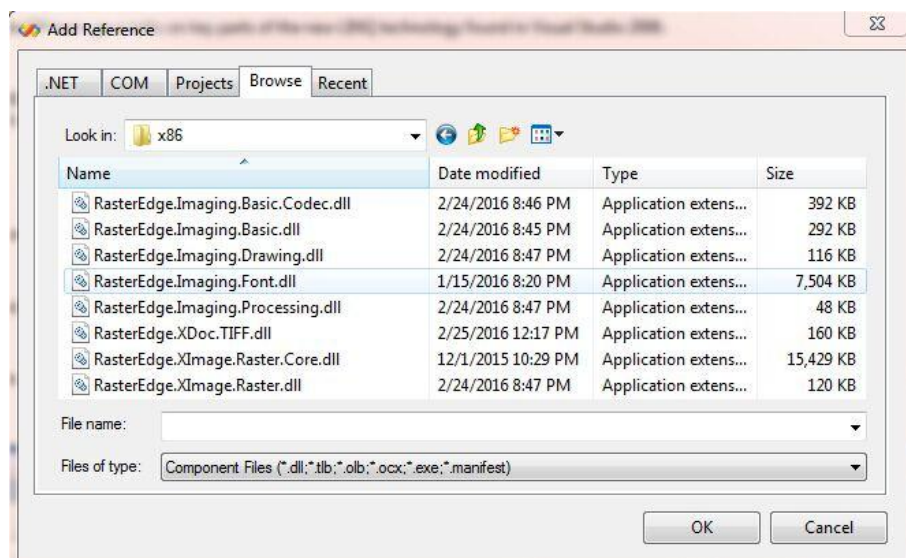
To use RasterEdge.XDoc.DICOM library successfully, the following libraries are necessary:

RasterEdge.Imaging.Basic.dll
RasterEdge.Imaging.Basic.Codec.dll
RasterEdge.Imaging.JPEG2000.dll
RasterEdge.Imaging.Processing.dll
RasterEdge.XImage.Raster.Core.dll
RasterEdge.XImage.Raster.dll
RasterEdge.XDoc.DICOM.dll

Add References

The following steps will show you how to use in Visual Studio .NET:

1. In the Solution Explorer, expand the project node you want to add a reference to.
2. Right-click the project's **References** node and select **Add Reference**.
3. In the Add Reference dialog box, Click **Browse** and Navigate to the specified folder.
4. Select the dlls as listed in the following screenshot, Click **OK**.



5. The RasterEdge.XDoc.DICOM for .NET reference appears under the project's **References** node.

If you want to know how to select dlls according to your specific development environment, please refer to the **Readme.txt** file in the **/Bin** directory.

FAQ

Errors On Visual Studio

If you get the error as follows:

"Could not load file or assembly 'RasterEdge.XDoc.DICOM' or one of its dependencies. An attempt was made to load a program with an incorrect format."

Please check your project configures as following ways:

1. If you are using the .NET Framework 4.0 dlls, please confirm that:

Right-click the project -> Properties ->

a. Application -> Target framework: .NET Framework 4 or higher

b. Build -> Platform target: x86 if using x86 dlls, x64 if using x64.

2. If you are using the .NET Framework 2.0 dlls, please confirm that:

Right-click the project -> Properties ->

c. Application -> Target framework: .NET Framework 3.0 or 3.5

d. Build -> Platform target: x86 if using x86 dlls, x64 if using x64.

Errors On IIS

If you configure IIS to run and 500.19 error occurs, then it may be caused by:

1. Not registered the .net framework to the iis. (One of reasons: install a .net framework before the installation of iis.)
2. The site configured in IIS has no sufficient authority to operate. (Modify permission)

There are some solutions:

1. cd to C:\Windows\Microsoft.NET\Framework64\v2.0.50727, Command to re-register net framework to the iis: aspnet_regiis-i.
2. Right-click the correspond site -> Edit Permissions -> Security -> Group or user names -> Edit -> Add -> Add Everyone users given Full Control permissions.

If you get the error as follows:

"Could not load file or assembly 'RasterEdge.Imaging.Basic' or any other one assembly or one of its dependencies. An attempt was made to load a program with an incorrect format."

Please check your IIS configure as following ways:

- a. If you are using the .NET framework 4.0 or higher dlls, confirm that Web.config is using the content in **Web(for .net4.0 or higher).Config file**.
- b. After checking first step, if you are still facing the issue, confirm that:

If you are using x64 dlls, "Application Pools" -> "Set Application Pool Defaults..." -> "Enable 32-Bit

Applications" should be false.

If you are using **x86** dlls, "Application Pools" -> "Set Application Pool Defaults..." -> "Enable 32-Bit Applications" should be **true**.

Feature List

DICOM Generator

- [Create DICOM from byte array](#)
- [Create DICOM from file path](#)
- [Create DICOM from stream](#)
- [Create DICOM from bitmaps](#)

DICOM Document

- [Get DICOM Document properties](#)
- [Get a particular DICOM page](#)
- [Convert To Images](#)
- [Convert To Document](#)
- [Extract DICOM pages](#)

DICOM Page

- [Get DICOM page properties](#)
- [Convert To Image](#)

Annotations

- [Add Annotation](#)
- [Add Image Annotation](#)

Save

- [Save DICOM to file path](#)
- [Save DICOM to byte array](#)
- [Save DICOM to stream](#)

Programmer Guide

DICOM Object Generator

Our RasterEdge.XDoc.DICOM dll allows developer to create TIFF file from Bitmap and Dicom file.

Create DICOM Document object from file path/data

You can easily create a dicom document object just follow the code below:

C#:

```
//load a dicom file object from input file path
```

```
String inputPath = @"F:\input.dcm";
```

```
DCMDocument dcmDoc = new DCMDocument(inputPath);
```

Related API(s) (**DCMDocument.cs**):

```
public DCMDocument(string filePath);
```

Descriptions:

Create a dicom document object from file path.

Parameters:

| Name | Description | Valid Value |
|----------|-----------------------|---|
| filePath | the dicom file's path | there must be a valid dicom file at this location |

```
public DCMDocument(byte[] fileData);
```

Descriptions:

Create a dicom document object from byte array

Parameters:

| Name | Description | Valid Value |
|----------|---|---------------|
| fileData | a byte array contains the whole dicom file data | must be valid |

```
public DCMDocument(Stream stream);
```

Descriptions:

Create a dicom document object from a stream.

Parameters:

| Name | Description | Valid Value |
|--------|--|---------------|
| stream | the stream must contains a whole dicom file data | must be valid |

Create DICOM Document object from images.

It allows developers create dicom document object from a single bitmap or a bitmap array with optional informations.

Create Dicom document object without options.

C#:

```
//create a dicom file object from bitmap  
Bitmap bmp = new Bitmap(@"F:\sample.jpg");  
DCMDocument dcmDoc = new DCMDocument(bmp);  
dcmDoc.Save(@"F:\output.dcm");
```

Related API(s) (**DCMDocument.cs**):

```
public DCMDocument(Bitmap bmp);
```

Description:

Create a dicom document object from bitmap with no format limited, that is, the bitmap can be a .jpeg image or a .png image or some else single frame image.

Parameters:

| Name | Description | Valid Value |
|------|--|---------------|
| bmp | the image for creating dicom document object | must be valid |

```
public DCMDocument(Bitmap[] bmps);
```

Descriptions:

Create a dicome document object from a bitmap array. These images must has the same size and color space.

Parameters:

| Name | Description | Valid Value |
|------|--|---------------------------------|
| bmps | a series of bitmaps for creating dicom document object | every image must be a valid one |

Create Dicom document object with specified information.

It allows create dicom document object from a single bitmap or a bitmap array with specified information, therefore, the patient name, patient id, study date can be defined by yourself.

C#:

```
//create a dicom file object from bitmap with specified information
```

```
Bitmap bmp = new Bitmap(@"F:\sample.jpg");
```

```
Infomation infos = new Infomation();
```

```
infos.PatientName = "SpongeBob"
```

```
infos.PatientSex = "male";
```

```
//format: YYYYMMDD    YYYY:year/MM:month/DD:day
```

```
infos.StudyDate = "20160729";
```

```
//format: HHMMSS    HH:hour/MM:minutes/SS:second
```

```
infos.StudyTime = "153120";
```

```
DCMDDocument dcmDoc = new DCMDDocument(bmp,infos);
```

```
dcmDoc.Save(@"F:\output.dcm");
```

Related API(s) (**DCMDDocument.cs**):

```
public DCMDDocument(Bitmap bmp, Infomation info);
```

Description:

Create dicom document object from bitmap with specified information.

Parameters:

| Name | Description | Valid Value |
|------|--|-----------------------|
| bmp | image for dicom document object creating | must be a valid image |
| info | a description for the dicom file. | - |

```
public DCMDDocument(Bitmap[] bmps, Infomation info);
```

Descriptions:

Create dicom document object from a bitmap array with specified information.

Parameters:

| Name | Description | Valid Value |
|------|---|---------------------------|
| bmps | images for dicom document object creating | every image must be valid |
| info | a description for the dicom file | - |

Working with Document

Get Page Count

The following demo code will show you how to get the total page number of dicom file.

C#:

```
String inputPath = @"F:\input.dcm";  
DCMDocument dcmDoc = new DCMDocument(inputPath);  
int pageCount = dcmDoc.GetPageCount();
```

Related API(s) (**DCMDocument.cs**):

```
public override int GetPageCount();
```

Return: 0 if failed.

Get Document Type

You can do follow to get the input document's type:

C#:

```
String inputPath = @"F:\input.dcm";  
DCMDocument dcmDoc = new DCMDocument(inputPath);  
DocumentType type = dcmDoc.GetDocumentType();
```

Related API(s) (**DCMDocument.cs**):

```
public override DocumentType GetDocumentType();
```

Description:

Get the document type of the input file.

Return:

DICOM, Invalid or other format if failed

Extract DICOM Page(s) to DICOM file/stream

To extract DICOM page(s) to a file or stream, the following steps will be helpful:

1. Open an existing DICOM file.
2. Define the page indexes will be extracted.
3. Call the method ExtractPages to save the specified pages as a new dicome file.

C#:

```
String inputPath = @"F:\input.dcm";  
String outputPath = @"F:\output.dcm";  
DCMDocument dcmDoc = new DCMDocument(inputPath);  
int[] pageIndex = new int[3] { 0, 1, 4 };  
dcmDoc.ExtractPages(pageIndex, outputPath);
```

Related API(s) (**DCMDocument.cs**):

```
public override void ExtractPages(int[] extractIds, string filePath);
```

Descriptions:

Extract the specifed pages and save them as a new dicom file. The compression type of the image in the new file is lossy jpeg.

Parameters:

| Name | Description | Valid Value |
|------------|------------------|---------------------|
| extractIds | page index | 0 to page count - 1 |
| filePath | output file path | valid file path |

```
public override void ExtractPages(int[] extractIds, Stream stream);
```

Description:

Extract the specified pages and save them as a new dicom file into the stream.

Parameters:

| Name | Description | Valid Value |
|------------|---------------|------------------------------------|
| extractIds | page index | 0 to page count – 1 |
| stream | output stream | valid file stream or memory stream |

Get a Particular Page

Follow the code below to get a specified page from the dicom file.

C#:

```
String inputPath = @"F:\input.dcm";  
DCMDocument dcmDoc = new DCMDocument(inputPath);  
DCMPage page = (DCMPage)dcmDoc.GetPage(0);
```

Related API(s) (**DCMDocument.cs**):

```
public override BasePage GetPage(int pageIndex);
```

Descriptions:

Get a specified page from the dicom document object.

Parameters:

| Name | Description | Valid Value |
|-----------|----------------|---------------------|
| pageIndex | the page index | 0 to page count - 1 |

Convert to Images

This section will tell you how to convert the whole dicom file to images. it's quite easy just get a dicom document object and call the method **ConvertToImages**. The following demo code will show you how to complete the conversion.

```
C#
//open a Dicom file
String inputPath = @"F:\7Frames.dcm";
//set the output directory;
String outputDir = @"F:\Dcm2jpeg\";
DCMDocument dcmDoc = new DCMDocument(inputPath);
//convert all frames to jpeg images
//it will create jpeg images in the directory whose name is "demo_"
//note:you must create the output folder on your disk.
dcmDoc.ConvertToImages(ImageType.JPEG, outputDir, "demo_");
```

Related API(s) (**DCMDocument.cs**):

```
public override void ConvertToImages(ImageType targetType, string directory, string
fileName);
```

Descriptions:

Convert all frames of dicom file to image with specified format and save them to the given folder.

Parameters:

| Name | Description | Valid Value |
|------------|------------------------------------|-------------|
| targetType | format of output image | - |
| directory | output directory | must exist |
| fileName | output image's name without suffix | a string |

```
public override void ConvertToImages(ImageType targetType, Stream[] streams);
```

Descriptions:

Convert all frames of dicom file to image with specified format and save them to memory stream or file stream.

Parameters:

| Name | Description | Valid Value |
|------------|----------------------------|-------------|
| targetType | format of the output image | - |
| streams | output stream | - |

```
public override void ConvertToImages(ImageType targetType, float zoomValue, Stream[]
streams);
```

Descriptions:

Convert all frames of dicom file to target format images with specified zoom value and save them to memory stream or file stream.

Parameters:

| Name | Description | Valid Value |
|------------|---------------------------------------|-------------|
| targetType | format of the output image | - |
| zoomValue | the magnification of the output image | >0 |
| streams | output stream | - |

```
public override void ConvertToImages(ImageType targetType, int resolution, Stream[] streams);
```

Descriptions:

Convert all frames of dicom file to the target format images with specified resolution and save them to memory stream or file stream.

Parameters:

| Name | Description | Valid Value |
|------------|--------------------------------|-------------|
| targetType | format of the output image | - |
| resolution | the resolution of output image | >0 |
| streams | output stream | |

```
public override void ConvertToImages(ImageType targetType, float zoomValue, string directory, string fileName);
```

Descriptions:

Convert all frames of dicom file to the target format images with specified zoom value and save them to the given folder.

Parameters:

| Name | Description | Valid Value |
|------------|---------------------------------------|-------------|
| targetType | format of the output image | - |
| zoomValue | the magnification of the output image | >0 |
| directory | output directory | must exist |
| fileName | output image's name without suffix | a string |

```
public override void ConvertToImages(ImageType targetType, int resolution, string directory, string fileName);
```

Descriptions:

Convert all frames of dicom file to the target format images with specified resolution and save them to the given folder.

Parameters:

| Name | Description | Valid Value |
|------------|------------------------------------|-------------|
| targetType | format of the output image | - |
| resolution | resolution of the output image | >0 |
| directory | output directory | must exist |
| fileName | output image's name without suffix | a string |

Convert To Document

Add extra references:

If you want to Convert DICOM document to PDF(TIFF) document, you need add following dll:

RasterEdge.XDoc.PDF(TIFF).dll

To achieve the conversion, please do as follows:

- 1: Open an existing DICOM file.
- 2: Call the method **ConvertToDocument** to complete the conversion.

The following demo code will show you how to convert DICOM file to TIFF file:

```
C#
//open a DICOM file
String inputPath = @"F:\input.dcm";
String outputPath = @"F:\output.tif";
DCMDocument dcmDoc = new DCMDocument(inputPath);
//convert DICOM file to TIFF file.
dcmDoc.ConvertToDocument(DocumentType.TIFF, outputPath);
```

Related API(s) (**DCMDocument.cs**):

public override void ConvertToDocument(**DocumentType** targetType, **Stream** stream);

Descriptions:

Convert dicom file to tiff document or pdf document and save it to stream.

Parameters:

| Name | Description | Valid Value |
|------------|-----------------------|------------------------------------|
| targetType | format of output file | - |
| stream | output stream | valid memory stream or file stream |

public override void ConvertToDocument(**DocumentType** targetType, **string** filePath);

Descriptions:

Convert dicom file to tiff document or pdf document and save it to the given path.

Parameters:

| Name | Description | Valid Value |
|------------|-----------------------|-------------|
| targetType | format of output file | - |
| filePath | output file path | a string |

public override void ConvertToDocument(**DocumentType** targetType, **float** zoomValue, **Stream** desStream);

Descriptions:

Convert dicom file to tiff document or pdf document with specified magnification and save it to stream.

Parameters:

| Name | Description | Valid Value |
|------------|------------------------------|------------------------------|
| targetType | format of output file | - |
| zoomValue | magnification of output file | >0 |
| desStream | output stream | memory stream or file stream |

```
public override void ConvertToDocument(DocumentType targetType, float zoomValue,
string filePath);
```

Descriptions:

Convert dicom file to tiff document or pdf document with specified magnification and save it to the given path.

Parameters:

| Name | Description | Valid Value |
|------------|------------------------------|-------------|
| targetType | format of output file | - |
| zoomValue | magnification of output file | >0 |
| filePath | output file path | - |

```
public override void ConvertToDocument(DocumentType targetType, int resolution, Stream
desStream);
```

Descriptions:

Convert dicom file to tiff document or pdf document with specified resolution and save it to stream

Parameters:

| Name | Description | Valid Value |
|------------|---------------------------|--------------------------------------|
| targetType | format of output file | - |
| resolution | resolution of output file | >0 |
| desStream | output stream | a valid memory stream or file stream |

```
public override void ConvertToDocument(DocumentType targetType, int resolution, string
filePath);
```

Descriptions:

Convert dicom file to tiff document or pdf document with specified resolution and save it to the given path.

Parameters:

| Name | Description | Valid Value |
|------------|---------------------------|-------------|
| targetType | format of output file | - |
| resolution | resolution of output file | >0 |
| filePath | output file path | - |

```
public override void ConvertToDocument(DocumentType targetType, Stream desStream,
ImageOutputOption options);
```

Descriptions:

Convert dicom file to tiff document or pdf document with optional settings and save it to stream

Parameters:

| Name | Description | Valid Value |
|------------|--------------------------|--------------------------------------|
| targetType | format of output file | - |
| desStream | output stream | a valid memory stream or file stream |
| options | settings for output file | - |

```
public override void ConvertToDocument(DocumentType targetType, string filePath,
ImageOutputOption options);
```

Descriptions:

Convert dicom file to tiff document or pdf document with optional settings and save it to the given path.

Parameters:

| Name | Description | Valid Value |
|------------|--------------------------|-------------|
| targetType | format of output file | - |
| filePath | output file path | - |
| options | settings for output file | - |

Working with Page

Get Frame Width/Height

```
C#  
//load DICOM file Document object from file path  
String inputPath = @"F:\input.dcm";  
DCMDocument dcmDoc = new DCMDocument(inputPath);  
//get the first frame  
DCMPage page = (DCMPage)(dcmDoc.GetPage(0));  
//the width of the frame  
int width = (int)page.GetWidth();  
int height = (int)page.GetHeight();
```

Related API(s) (**DCMPage.cs**):

```
public override float GetHeight();
```

Descriptions:

Get the height of the frame.

Return:

0 if failed

```
public override float GetWidth();
```

Description:

Get the width of the frame

Return:

0 if failed

Convert to Image

To convert the specified frame of DICOM file to image, you just need several steps as follows:

- 1: Open an existing DICOM file through DICOM document object.
- 2: Call Method `GetPage` to get a frame of DICOM file.
- 3: Call method `ConvertToImage` to convert the frame to image and save it to file path, byte array or stream.

The following demo code will show the conversion in details:

```
C#
//open a DICOM file
String inputFilePath = @"F:\7Frames.dcm";
String outputFilePath = @"F:\output.png";
DCMDocument dcmDoc = new DCMDocument(inputFilePath);
//get the first frame of the file.
int pageIndex = 0;
DCMPage page = (DCMPage)dcmDoc.GetPage(pageIndex);
//convert Dicome frame to image
Bitmap bmp = page.ConvertToImage();
//save the bitmap to file path.
bmp.Save(outputFilePath);
```

Related API(s) (**DCMPage.cs**):

```
public override Bitmap ConvertToImage();
```

Descriptions:

Convert the specified frame of dicom file to image.

Return:

Null if failed.

```
public override Bitmap ConvertToImage(float zoomValue);
```

Descriptions:

Convert a frame of dicom file to image with specified magnification.

Parameters:

| Name | Description | Valid Value |
|-----------|-----------------------------------|-------------|
| zoomValue | the magnification of output image | >0 |

Return:

Null if failed.

```
public override Bitmap ConvertToImage(int targetResolution);
```

Descriptions:

Convert a frame of dicom file to image with specified resolution.

Parameters:

| Name | Description | Valid Value |
|------------------|----------------------------|-------------|
| targetResolution | resolution of output image | >0 |

Return: Null if failed.

public override Bitmap ConvertToImage(**Size** targetSize);

Descriptions:

Convert a frame of dicom file to image and scale it to the target size.

Parameters:

| Name | Description | Valid Value |
|------------|--------------------------|---------------------|
| targetSize | the size of output image | width>0 && height>0 |

Return:

Null if failed.

public override void ConvertToImage(**ImageType** targetType, **string** filePath);

Descriptions:

Convert a frame of dicom file to image with specified format and save it to the given path.

Parameters:

| Name | Description | Valid Value |
|------------|------------------------|-------------|
| targetType | format of output image | - |
| filePath | output path | - |

public override void ConvertToImage(**ImageType** targetType, **float** zoomValue, **string** filePath);

Descriptions:

Convert a frame of dicom file to image with specified format, and save it to the given path with target magnification.

Parameters:

| Name | Description | Valid Value |
|------------|-------------------------------|-------------|
| targetType | format of output image | - |
| zoomValue | magnification of output image | >0 |
| filePath | output path | - |

public override void ConvertToImage(**ImageType** targetType, **int** resolution, **string** filePath);

Descriptions:

Convert a frame of dicom file to image with specified format and save it to the given path with target resolution.

Parameters:

| Name | Description | Valid Value |
|------------|----------------------------|-------------|
| targetType | format of output image | - |
| resolution | resolution of output image | >0 |
| filePath | output file path | - |

public override byte[] ConvertToImageBytes(**ImageType** targetType);

Descriptions:

Convert a frame of dicom file to image with specified format and return it as a byte array.

Parameters:

| Name | Description | Valid Value |
|------------|------------------------|-------------|
| targetType | format of output image | - |

Return:

Null if failed.

```
public override byte[] ConvertToImageBytes(ImageType targetType, float zoomValue);
```

Descriptions:

Convert a frame of dicom file to image with specified format and magnification, then return it as a byte array.

Parameters:

| Name | Description | Valid Value |
|------------|-------------------------------|-------------|
| targetType | format of output image | - |
| zoomValue | magnification of output image | >0 |

```
public override byte[] ConvertToImageBytes(ImageType targetType, int targetResolution);
```

Descriptions:

Convert a frame of dicom file to image with specified format and resolution, then return it as a byte array.

Parameters:

| Name | Description | Valid Value |
|------------------|----------------------------|-------------|
| targetType | format of output image | - |
| targetResolution | resolution of output image | >0 |

```
public override Bitmap ConvertToImageFitHeight(int height);
```

Descriptions:

Convert a frame of dicom file to image with specified height.

Parameters:

| Name | Description | Valid Value |
|--------|------------------------|-------------|
| height | height of output image | >0 |

Return:

Null if failed.

```
public override Bitmap ConvertToImageFitWidth(int width);
```

Descriptions:

Convert a frame of dicom file to image with specified width.

Parameters:

| Name | Description | Valid Value |
|-------|-----------------------|-------------|
| width | width of output image | >0 |

Return:

Null if failed.

```
public override void ConvertToImageStream(ImageType targetType, Stream stream);
```

Descriptions:

Convert a frame of dicom file with specified format and save it to a memory stream or a file stream.

Parameters:

| Name | Description | Valid Value |
|------------|------------------------|----------------|
| targetType | format of output image | - |
| stream | output stream | a valid stream |

```
public override void ConvertToImageStream(ImageType targetType, float zoomValue, Stream stream);
```

Descriptions:

Convert a frame of dicom file with specified format and mangnification and save it to a memory stream or a file stream.

Parameters:

| Name | Description | Valid Value |
|------------|--------------------------------|----------------|
| targetType | format of output image | - |
| zoomValue | mangnification of output image | >0 |
| stream | output stream | a valid stream |

```
public override void ConvertToImageStream(ImageType targetType, int resolution, Stream stream);
```

Descriptions:

Convert a frame of dicom file with specified format and resoluition and save it to a memory stream or a file stream.

Parameters:

| Name | Description | Valid Value |
|------------|----------------------------|----------------|
| targetType | format of output image | - |
| resolution | resolution of output image | >0 |
| stream | output stream | a valid stream |

```
public override Bitmap CropImage(Rectangle sourceRegion, Size targetSize);
```

Descriptions:

Take a sub region out from the frame and scale it to target size.

Parameters:

| Name | Description | Valid Value |
|--------------|-------------------------------|-------------------|
| sourceRegion | take this sub region out | a valid rectangle |
| targetSize | target size of the sub region | a valid Size |

Return:

Null if failed.

```
public override Bitmap GetBitmap();
```

Descriptions:

Convert a frame of dicom file to bitmap.

Return:

Null if failed.

`public override Bitmap GetBitmap(float zoomValue);`

Descriptions:

Convert a frame of dicom file to bitmap with specified magnification.

Parameters:

| Name | Description | Valid Value |
|-----------|------------------------------|-------------|
| zoomValue | magnification of output file | >0 |

Return:

Null if failed.

`public override Bitmap GetBitmap(int resolution);`

Descriptions:

Convert a frame of dicom file to bitmap with specified resolution.

Parameters:

| Name | Description | Valid Value |
|------------|---------------------------|-------------|
| resolution | resolution of output file | >0 |

Return:

Null if failed.

`public override Bitmap GetBitmap(Rectangle sourceRectangle, Size targetSize);`

Descriptions:

Take a specified sub region out from the frame and scale it to target size.

Parameters:

| Name | Description | Valid Value |
|-----------------|-------------------------------|-------------------|
| sourceRectangle | take this sub region out | a valid rectangle |
| targetSize | target size of the sub region | a valid Size |

Return:

Null if failed.

Working with Annotation

Add annotation

In order to add annotation to dicom files, you need to add extra reference:

- RasterEdge.Imaging.Annotation.dll

Three steps to add an annotation on the frame:

- 1: Add RasterEdge.Imaging.Annotation dll to your reference.
- 2: Create a annotation through AnnotationGenerator
- 3: Add the annotation on the frame.

There are 9 types annotation, and you can create them by calling the method below:

| Type | Method |
|-------------|---|
| Line | AnnotationGenerator.CreateLineAnnotation() |
| Lines | AnnotationGenerator.CreateLinesAnnotation() |
| Image | AnnotationGenerator.CreateEmbeddedImageAnnotation() |
| FreeHand | AnnotationGenerator.CreateFreeHandLineAnnotation() |
| Ellipse | AnnotationGenerator.CreateEllipseAnnotation() |
| Rectangle | AnnotationGenerator.CreateRectangleAnnotation() |
| Polygon | AnnotationGenerator.CreatePolygonAnnotation() |
| Text | AnnotationGenerator.CreateTextAnnotation() |
| RubberStamp | AnnotationGenerator.CreateRubberStampAnnotation() |

Table 1-1

The following demo code will show you how to add a text annotation to dicom file:

```
C#
//open a DICOM file
DCMDocument dcmDoc = new DCMDocument(@"F:\input.dcm");
DCMPage page = (DCMPage)dcmDoc.GetPage(0);
//create a text annotation
TextAnnotation annotation = AnnotationGenerator.CreateTextAnnotation(10, 100F, 200F,
100F, "www.RasterEdge.com", new System.Drawing.Font("Arial", 12F));
//add it on the frame
page.AddAnnotation(annotation);
dcmDoc.Save(@"F:\output.dcm");
```

Related API(s) (DCMPage.cs):

```
public override void AddAnnotation(AnnotationHandler annoHandler);
```

Descriptions:

Add annotation on the frame.

Parameters:

| Name | Description | Valid Value |
|-------------|----------------------|---|
| annoHandler | An annotation object | Created by the Method in the Table 1-1 |

```
public void AddAnnotation(AnnotationHandler annoHandler, float zoomValue);
```

Descriptions:

Add annotation on the frame with specified magnification.

Parameters:

| Name | Description | Valid Value |
|-------------|--------------------------------|---|
| annoHandler | An annotation object | Created by the Method in the Table 1-1 |
| zoomValue | magnification of annotation | >0 |

Add Image Annotation

The following demo code will show you how to add an image annotation to dicom file:

```
C#
//open a DICOM file
DCMDocument dcmDoc = new DCMDocument(@"F:\input.dcm");
//get specified DICOM page
DCMPage page = (DCMPage)dcmDoc.GetPage(0);
//load an image
BaselImage image = new REImage(@"F:\logo.png");
//set the image annotation location
PointF position = new PointF(200f, 200f);
//add image annotation on the frame at specified location.
page.AddImage(image, position);
dcmDoc.Save(@"F:\output.dcm");
```

Related API(s) (**DCMPage.cs**):

```
public override void AddImage(BaselImage image, PointF point);
```

Descriptions:

Add image on frame at the specified position

Parameters:

| Name | Description | Valid Value |
|-------|------------------------------|---|
| image | the image to add on the page | can't be null |
| point | location of the image | x>=0 && x<=page.width y>=0 && y<=page.height |

Save

RasterEdge.XDoc.DICOM dll allows developer to save the DICOM document object to file path, stream and byte array.

```
C#  
//load DICOM file Document object from file path  
String inputPath = @"F:\input.dcm";  
DCMDocument dcmDoc = new DCMDocument(inputPath);  
//save the DICOM Document object to the file path  
dcmDoc.Save(@"F:\output.dcm");
```

Related API(s) (**DCMDocument.cs**):

public override void Save(string filePath);

Descriptions:

Save dicom document object to the given file path.

Parameters:

| Name | Description | Valid Value |
|----------|------------------|-------------|
| filePath | output file path | - |

public override byte[] SaveToBytes();

Descriptions:

Save dicom document object to byte array.

Return:

Null if failed.

public override void SaveToStream(Stream stream)

Descriptions:

Save dicom document object to stream

Parameters:

| Name | Description | Valid Value |
|--------|---------------|--------------------------------------|
| stream | output stream | a valid memory stream or file stream |