RasterEdge.XImage.Raster

Developer Guide

Contents

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.XImage.Raster 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.XImage.Raster in .NET project

Necessary Libraries

To use RasterEdge.XDoc.PDF library successfully, the following libraries are necessary:
- RasterEdge.Imaging.Basic.dll
- RasterEdge.XImage.Raster.Core.dll
- RasterEdge.XImage.Raster.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**.
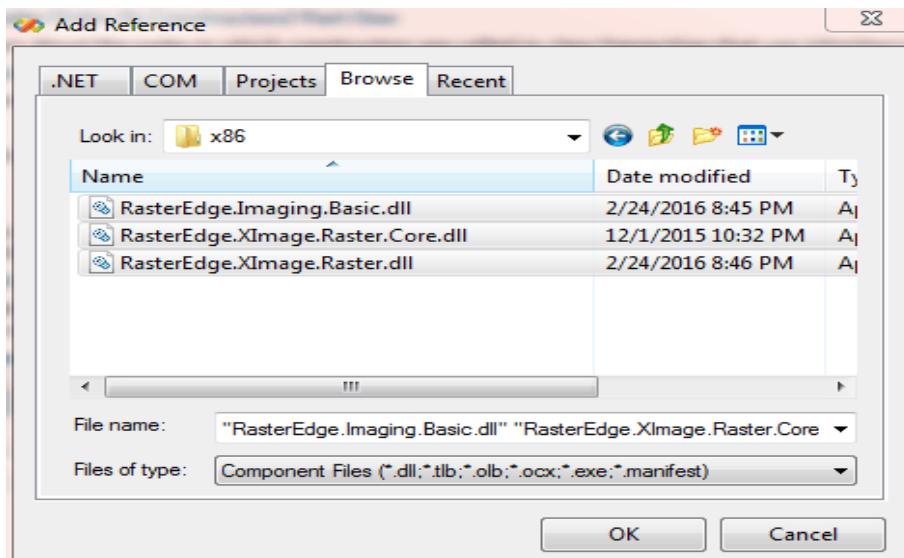
5. The RasterEdge.XImage.Raster for .NET reference appears under the project's **References** node.

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.XImage.Raster 'or one of its dependenci es. 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 usersgiven 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.

Load

Base Load

Load **RasterImage** object from file path, stream, bitmap, Hbitmap, byte array and so on.

C#

```
//load a RasterImage object from file path.
RasterImage tifDoc = new RasterImage(@"F:\input.png");
```

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

public RasterImage(string filePath);

Description:

Load **RasterImage** object from file path

Parameters:

Name	Description	Valid Value
filePath	input file path	-

public RasterImage(Bitmap bmp);

Description:

Load **RasterImage** object from bitmap.

Parameters:

Name	Description	Valid Value
bmp	input bitmap	can't be null

public RasterImage(Bitmap[] bmps);

Description:

Load **RasterImage** object from bitmap array.

Parameters:

Name	Description	Valid Value
bmps	array of bitmap	can't be null

public RasterImage(byte[] imageDataBytes);

Description:

Load **RasterImage** object from file's data.

Parameters:

Name	Description	Valid Value
imageDataBytes	the whole data of the input file	can't be null

public RasterImage(Stream imageStream);

Description:

Load **RasterImage** object from the input stream.

Parameters:

Name	Description	Valid Value
imageStream	input stream	can't be null

public RasterImage(int width, int height, Color background);

Description:

Load a pure RasterImage object with specified width and height.

Parameters:

Name	Description	Valid Value
width	width of RasterImage object	>0
height	height of RasterImage object	>0
background	color of RasterImage object	-

public RasterImage(IntPtr Hbitmap);

Description:

Load a **RasterImage** object from Hbitmap

Parameters:

Name	Description	Valid Value
Hbitmap	the input Hbitmap	can't be null

public RasterImage(IntPtr Hbitmap, IntPtr Hpalette);

Description:

Load a **RasterImage** object from Hbitmap with palette

Parameters:

Name	Description	Valid Value
Hbitmap	the input Hbitmap	can't be null
Hpalette	input palette	can't be null

public int LoadImageFromGraphicScreen();

Description:

Load a **RasterImage** object from the screen.

Return:

0 if successed.

Load with option

There are a series of settings for loading, so you can specify the size, color, rotation, and resolution when load the input image as a **RasterImage** object, and you can also decide which area of the input image can be load, or if use the embedded color management. It's very easy to use, just three steps needed:

- Create a LoadOption object
- Set the values of the LoadOption
- Pass the LoadOption to RasterImage' constructor

C#

```
//Create a load option
LoadOption option = new LoadOption();
//set the rotate angle
option.RotateAngle = RotateAngle.Rotate90;
//load a RasterImage object with load option    from file path.
RasterImage image = new RasterImage(@"F:\input.png",option);
```

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

public RasterImage(string filePath, LoadOption loadOption);

Description:

Load a **RasterImage** object with load options.

Parameters:

Name	Description	Valid Value
filePath	input file path	-
loadOption	a series settings for loading	-

public RasterImage(Btimap bmp, LoadOption loadOption);

Description

Load **RasterImage** object from bitmap with specified load option.

Parameters:

Name	Description	Valid Value
bmp	input bitmap	can' t be null
loadOption	a series settings for loading	-

public RasterImage(Stream imageStream, LoadOption loadOption);

Description:

Load **RasterImage** object from stream with specified load option.

Parameters:

Name	Description	Valid Value
imageStream	input stream	can't be null
loadOption	a series settings for loading	-

6

public RasterImage(byte[] imageDataBytes, LoadOption loadOption);

Description:

Load **RasterImage** object from file data with specified load option.

Parameters:

Name	Description	Valid Value
imageDataBytes	the whole data of input file	can't be null
loadOption	a series settings for loading	-

public RasterImage(IntPtr Hbitmap, LoadOption loadOption);

Description:

Load **RasterImage** object from Hbitmap with specified load option.

Parameters:

Name	Description	Valid Value
Hbitmap	the input HBitmap	can't be null
loadOption	a series settings for loading	-

public RasterImage(IntPtr Hbitmap, IntPtr Hpalette, LoadOption loadOption);

Description:

Load **RasterImage** object from Hbitmap with specified palette and load option.

Parameters:

Name	Description	Valid Value
Hbitmap	the input Hbitmap	can't be null
Hpalette	input palette	can't be null
loadOption	a series settings for loading	-

The values of load options

Load Option		
Name	**Description**	**Valid Value**
CropRectangle	only load this specified area of input image as RasterImage object	a rectangle, can't be null
Jpeg	this is a class special for jpeg format. If smooth the input image	true: smooth the jpeg file false：not
LoadAlphaChannel	If load the alpha channel, if set false, image's transparency will be abandoned.	true: load alpha channel false: not load
LoadResizeAntiAlias	if turned of Anti-alias	true: turn on false: not
MaintainAspectRatio	if maintain the aspect ratio when resize the input image.	true: maintain it false: not
NegateColor	if negate the color of input image	true: negate the color false: not
Resize	set the target size for loading	a rectangle, can't be null
RotateAngle	set the rotate angle for loading	choose from enum **RotateAngle**
Rresolution	set the resolution for loading	>0
RenderIntent	set the render intent for loading	choose form enum **RenderingIntents**
ThumbnailSize	load the input image as thumbnail	choose from enum **ThumbnailSize**
UseEmbeddedColorManagement	if use embedded color management.	true: use it false: not use
TargetProfileName	use the specified color file, should set flag UseEmbeddedColorManagement to false	-

Save

Base Save

Save **RasterImage** object to file path, stream and byte array.

```csharp
C#
//load a RasterImage object
RasterImage image = new RasterImage(@"F:\input.png");
//save it to the file path.
image.Save(@"F:\output.png");
```

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

public void Save(string filePath);

Description:

Save **RasterImage** object to the given file path.

Parameters:

Name	Description	Valid Value
filePath	output file path	-

public byte[] SaveToBytes(ImageFormat format);

Description:

Save **RasterImage** object to byte array with specified format

Parameters:

Name	Description	Valid Value
format	output format	choose from enum **ImageFormat**

Return:

Null if failed.

public void SaveToStream(Stream stream, ImageFormat format);

Description:

Save **RasterImage** object to stream with specified format.

Parameters:

Name	Description	Valid Value
stream	output stream	-
format	output format	choose from enum **ImageFormat**

Save With Option

There are a series of settings for saving. Every output format has their own special save options. The table at the end of this section will explain these parameters in detail. To save **RasterImage** object with option just need three steps:

- Create a SaveOption object.
- Set the corresponding arguments
- Pass the object to the **Save** method.

```csharp
C#
//load a RasterImage object
RasterImage image = new RasterImage(@"F:\input.png");
//create a SaveOption object.
SaveOption saveOption = new SaveOption();
//image format must be set
saveOption.ImageFormat = RasterEdge.XImage.Raster.ImageFormat.PNG;
//assign a filter to png
saveOption.Png.Filter = PNGFilter.PAETH;
//turn on the interlace
saveOption.Png.Interlaced = true;
//set compress level to 7. LEVEL0:least but fast, LEVEL9:best but slowest.
saveOption.Png.Level = PNGCompressLevel.LEVEL9;
//save it to the path.
image.Save(@"F:\output.png", saveOption);
```

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

public void Save(string filePath, SaveOption saveOption);

Description:

Save **RasterImage** object to the given path with specified save option.

Parameter:

Name	Description	Valid Value
filePath	output file path	-
saveOption	a series of setting for saving	-

public byte[] SaveToBytes(SaveOption saveOption);

Description:

Save **RasterImage** object to byte array with specified save option.

Parameters:

Name	Description	Valid Value
saveOption	a series of setting for saving	-

Return:

Ouput file data.

public void SaveToStream(Stream stream, SaveOption saveOption);

Description:

Save **RasterImage** object to stream with specified save option.

Parameters:

Name	Description	Valid Value
stream	output stream	-
saveOption	a series of setting for saving	-

Values of **SaveOption** objet:

SaveOption		
Name	**Description**	**Valid Value**
ImageFormat	format for output file	choose from enum **ImageFormat**
ThumbnailSize	the size of the output, set this option will save the RasterImage object as a thumbnail	choose from enum **ThumbnailSize**
UseEmbeddedColorManagement	if use the embedded color management	true: use false: not use
RenderIntent	set the render intent for output file	choose from enum **RenderIntent**
TargetProfileName	the path of color profile	-
Tiff	through this class to set tiff file save option	-
Bmp	through this class to set bitmap file save option	-
Png	through this class to set png file save option	-
Jpeg	through this class to set jpeg file save option	-
Gif	through this class to set gif file save option	-
Jp2	through this class to set jp2 file save option	-

Values of SaveOption.Png:

SaveOption.Png		
Name	**Description**	**Valid Value**
Filter	filtering the data before compression	choose from enum PNGFilter
Interlaced	if save the png file in interlaced format. Turn on it if there is a slow internet connection.	true: save in interlaced format false: not
Level	compression level	0 to 9. 0:least but fast compression 9:best but slowest compression
Strategy	compression strategy	0 to 4. 0:default compression strategy 1:filtered compression strategy 2:Huffman_only compression strategy 3:RLE compression strategy 4:fixed ZLIB compression strategy.
TransparencyColor	pick a color form the image, and make it transparent	-
TransparencyMatch	colors within this distance are considered equal.	>0

Values of SaveOption.Tif:

SaveOption.Tiff		
Name	**Description**	**Valid Value**
Compression	set compression for output tif file	choose from enum Compression
ColorSpace	set color space for output tif file	choose from enum **ColorSpace**
MultiPage	if save to a multipage tif file, otherwise only the first page of the file will be saved.	true: save in a multipage format false: just save the first page
RowsPerStip	set the number of rows per strip	>0

Values of SaveOption.Bmp:

SaveOption.Bmp		
Name	**Description**	**Valid Value**
Compression	set the compression for output bmp file	choose from enum **Compression**

Values of SaveOption.Jpeg:

SaveOption.Jpeg		
Name	**Description**	**Valid Value**
Progressive	if save jpeg file in progressed format. Turn on it if there is a slow internet connection.	true: save in progressive format false: not
OptimizedHuffman	if use the optimized Huffman codes during encoding instead of using default Huffman codes	true: use optimized Huffman codes false: not use
SaveToGray	if save the output jpeg file to gray	true: save to gray false: not
ColorSpace	set the output jpeg file's color space	choose from enum **ColorSpace**
Quality	set compression level	0 to 99 default:75
SubSampleType	specify sub sample type for jpeg encoder, use default compress quality 85	choose from enum **SubSampleType**
SmoothImage	if smooth the output jpeg file	true: smooth it false: not

Values of SaveOption.Gif:

SaveOption.Gif		
Name	**Description**	**Valid Value**
Interlaced	if save gif file in interlaced format	true: save it in interlaced format false: not
TransparencyColor	pick a color from color palette and define it as a transparent color	-
TransparencyMatch	colors within this distance are considered equal	>0
Version	set output gif's version note:87a not support animate delays nor transparent background colors	choose from enum **GIFVersion**

Values of SaveOption.Jp2:

SaveOption.Jp2		
Name	**Description**	**Valid Value**
SaveToGray	if save the jp2 file to gray scale	true: save to gray false: not
PSNR	set the Peak Signal to Noise Ratio(DB)	>0
CompressRate	set jp2 file's compression rate	0 to 1
ProgressOrder	set jp2 file's progress order	choose from enum **ProgressOrder**
TileWidth	set jp2 file's tile width	>0
TileHeigth	set jp2 file's tile height	>0
Lossless	if save jp2 file in lossless format	true: save in lossless format false: not

Properties

Color space

Get color space of the input file:

```
C#
RasterImage image = new RasterImage(@"F:\input.png");
//get color space.
ColorSpace colorSpace = image.ColorSpace;
```

Change color space of the input file:

```
C#
RasterImage image = new RasterImage(@"F:\input.png");
//assign a processer to the image
ImageProcess process = new ImageProcess(image);
//change the image's color space to gray
process.TransformColorspace(ColorSpace.GRAY);
//save it
image.Save(@"F:\output.png");
```

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

public void TransformColorspace(ColorSpace colorSpace);

Description:

Change the input image's color space.

Parameters:

Name	Description	Valid Value
colorSpace	the target color place	-

public void TransformColorspace(ColorSpace colorSpace, int pageIndex);

Description:

Change the specified page's color space.

Parameters:

Name	Description	Valid Value
colorSpace	the target color place	-
pageIndex	page index	0 to page count - 1

Compression

Get compression type of the input file:

```csharp
//load an image
RasterImage image = new RasterImage(@"F:\input.png");
//assign a processor to the image
ImageProcess process = new ImageProcess(image);
//get the image's compression type, default the first page.
Compression compress = process.GetCompressionType();
```

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

public Compression GetCompressionType();

Description:

Get the input file's compression type

Return:

The file's compress type.

public Compression GetCompressionType(int pageIndex);

Description:

Get the specified page's compression type.

Parameters:

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

Return:

The page's compression type.

Change the compression type of the input file

```csharp
//load an image
RasterImage image = new RasterImage(@"F:\input.png");
//assign a processor to the image
ImageProcess process = new ImageProcess(image);
//change the image's compression type to jpeg
process.TransformCompression(Compression.JPEG);
//save it
image.Save(@"F:\output.jpg");
```

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

public void TransformCompression(Compression compression);

Description:

Change all pages compression type of **RasterImage** object.

Parameters:

Name	Description	Valid Value
compression	the target compression type	-

public void TransformCompression(Compression compression, int pageIndex);

Description:

Change the specified page's compression type of **RasterImage** object.

Parameters:

Name	Description	Valid Value
compression	the target compression type	-
pageIndex	page index	0 to page count - 1

Resolution

Page count

Get input file's page count

Page Width/Height

Get input file's width and height:

```csharp
//load an image
RasterImage image = new RasterImage(@"F:\input.gif");
//default get the first page's width and height in pixel
float width = image.Width;
float height = image.Height;
```

Get page width and height by page index:

```csharp
//load an image
RasterImage image = new RasterImage(@"F:\input.gif");
//assign a processor to the input image
ImageProcess process = new ImageProcess(image);
//get page's width and height in pixel through page index
//get the second page's width and height.
int width = process.GetImageWidth(1);
int height = process.GetImageHeight(1)
```

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

```csharp
public int GetImageWidth();
```

Description:

Get width of the first page.

Return:

-1 if failed.

```csharp
public int GetImageWidth(int pageIndex);
```

Description:

Get the specified page's width through the page index.

Parameters:

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

Return:

-1 if failed.

Clone

Get a copy of the input file.

```
C#

//load an image
RasterImage image = new RasterImage(@"F:\input.png");
//get a copy
RasterImage copy = image.Clone();
```

GetDiskSize

Get the file size on disk.

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

public int GetDiskSize ();

Description:

Get the file size on disk.

Return:

-1 if failed.

public int GetDiskSize(int pageIndex);

Description:

Get the specified page's size on disk.

Parameters:

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

Return:

-1 if failed.

Convert

Convert To Bitmap

There are two ways to convert **RasterImage** object to bitmap(s), two different ways get the same result. The first way is through the object itself, and the other is calling the static method of class **ConvertHandler** to complete the conversion.

The first way:

```
C#
//load a RasterImage object
RasterImage image = new RasterImage(@"F:\input.png");
//Convert the first page to bitmap
Bitmap bmp = image.ToBitmap();
```

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

public Bitmap ToBitmap();

Description:

Convert the first page of **RasterImage** object to bitmap.

Return:

Null if failed.

public Bitmap ToBitmap(int pageIndex);

Description:

Convert the specified page of the **RasterImage** object to bitmap.

Parameters:

Name	Description	Valid Value
pageIndex	page index	0 to page count − 1

Return:

Null if failed.

public IntPtr ToHbitmap();

Description:

Convert the first page of RasterImage object to HBitmap.

Return:

0 if failed.

public IntPtr ToHbitmap(int pageIndex);

Description:

Convert the sepcified page of **RasterImage** object to Hbitmap.

Parameters:

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

Return:

0 if failed.

The second way is by calling static method of class **ConvertHandler.**

C#

//load a RasterImage object
RasterImage image = new RasterImage(@"F:\input.png");
//convert the first page of the object to bitmap
Bitmap bmp = ConvertHandler.ConvertToBitmap(image);

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

public static Bitmap ConvertToBitmap(RasterImage image)

Description:

Convert the first page of **RasterImage** object to bitmap.

Parameters:

Name	Description	Valid Value
image	the input **RasterImage** object	can't be null

Return:

Null if failed.

public static IntPtr ConvertToHBitmap(RasterImage image, int pageIndex)

Description:

Convert the specifed page of the **RasterImage** object to HBitmap.

Parameters:

Name	Description	Valid Value
image	the input **RasterImage** object	can't be null
pageIndex	page index	0 to page count - 1

Return:

0 if failed.

public static IntPtr ConvertToHBitmap(RasterImage image)

Description:

Convert the first page of **RasterImage** object to HBitmap.

Parameters:

Name	Description	Valid Value
image	the input **RasterImage** object	can't be null

Return:

0 if failed.

Convert To Other Format

This section will explain how developers complete the conversion form one format to another. To complete the conversion, just call the static method 'Convert' through class ConvertHandler and set the input file path and output file path like code below:

C#

```
// This method only convert the first page, if the multipage conversion is needed please go
to the section 'Save'.
ConvertHandler.Convert(@"F:\input.tif", @"F:\output.png");
```

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

public static int Convert(string inputFilePath, string outputFilePath);

Description:

Convert the input file to specified file format

Parameters:

Name	Description	Valid Value
inputFilePath	input file path	-
outputFilePath	output file path	-

Return:

-1 if failed.

public static int Convert(Stream inputStream, string outputFilePath);

Description:

Convert the input stream to specified file format

Parameters:

Name	Description	Valid Value
inputStream	input file's stream	-
outputFilePath	output file path	-

Return:

-1 if failed.

public static int Convert(byte[] inputBytes, string outputFilePath);

Description:

Convert byte array to specified file format.

Parameters:

Name	Description	Valid Value
inputBytes	all data of input file	can't be null
outputFilePath	output file path	-

Return:

-1 if failed.

public static Bitmap ConvertEMFtoBitmap(Stream emfStream, string path);
public static Bitmap ConvertEMFtoBitmap(Stream emfStream, string path, int width, int

height);
public static byte[] ConvertToBytes(string inputFilePath, string ext);

Description:

Convert input file to specified file format, and return it as byte array.

Parameters:

Name	Description	Valid Value
inputFilePath	input file path	-
ext	extension of target file	e.g. .tif .bmp .png...

Return:

The data of output file.

public static byte[] ConvertToBytes(Stream inputStream, string ext);

Description:

Convert input stream to specified format, and return it as byte array.

Parameters:

Name	Description	Valid Value
inputStream	input file's stream	-
ext	extension of target file	e.g. .tif .bmp .png...

Return:

The data of output file.

public static byte[] ConvertToBytes(byte[] inputBytes, string ext);

Description:

Convert input file's data to specified format, and return it as byte array.

Parameters:

Name	Description	Valid Value
inputBytes	the whole data of input file	can't be null
ext	extension of target file	e.g. .tif .bmp .png...

Return:

The data of output file.

public static int ConvertToIcon(string inputFilePath, string outputPath, Size targetSize);

Description:

Convert input file to icon with specified size.

Parameters:

Name	Description	Valid Value
inputFilePath	input file path	-
outputPath	output file path	-
targetSize	the size of output file	max size: 256 * 256

Return:

-1 if failed.

public static int ConvertToIcon(string inputFilePath, string outputPath, Size targetSize, int colorsNum);

Description:

Convert input file to icon with specified size and colors.

Parameters:

Name	Description	Valid Value
inputFilePath	input file path	-
outputPath	output file path	-
targetSize	icon's size	max size: 256 * 256
colorsNum	number of colors the output icon contain	>0

Return:

-1 if failed.

public static Stream ConvertToStream(string inputFilePath, string ext);

Description:

Convert input file to specified format and return it as a stream.

Parameters:

Name	Description	Valid Value
inputFilePath	input file path	-
ext	the extension of output file	e.g. .tif .bmp .png...

Return:

The stream of output file.

public static Stream ConvertToStream(Stream inputStream, string ext);

Description:

Convert input stream to specified format, and return it as a stream.

Parameters:

Name	Description	Valid Value
inputStream	input file's stream	-
ext	the extension of output file	e.g. .tif .bmp .png...

Return :

The stream of output file.

public static Stream ConvertToStream(byte[] inputBytes, string ext);

Description:

Convert input file's data array to specified format, and return it as a stream.

Parameters:

Name	Description	Valid Value
inputBytes	the whole data of input file	can't be null
ext	the extension of output file	e.g. .tif .bmp .png...

Return:

The stream of output file.

Multipage Handle

Combine document

To complete the combination between two files, just follow the code below:

```csharp
C#

//load first file.
RasterImage image_1 = new RasterImage(@"F:\input_1.png");
//load second file.
RasterImage image_2 = new RasterImage(@"F:\input_2.png");
//set the output file path.
//ensure the output format support multipage, like .tif or .gif and so on.
String savePath = @"F:\output.tif";
MultiPageImageProcess.CombineDocument(image_1, image_2, savePath);
```

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

public static void CombineDocument(RasterImage sourceImage, RasterImage combineWith, string savePath);

Description:

Combine two files into one.

Parameters:

Name	Description	Valid Value
sourceImage	the first file	a valid RasterImage object
combineWith	the second file	a valid RasterImage object
savePath	output file path	-

Split Document

Split a multipage file into two, and save them to the given file path.

```csharp
C#

RasterImage tifDoc = new RasterImage(@"F:\7Pages.tif");
//sepcify the first output file's save path
String firstDocSavePath = @"F:\first.tif";
//specify the second output file's save path
String secondDocSavePath = @"F:\second.tif";
//boundary = 4
//first output file is made    up of first 5 pages, and rest of the pages will be second output file's page.
MultiPageImageProcess.SplitDocument(tifDoc, 4, firstDocSavePath, secondDocSavePath);
```

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

public static void SplitDocument(RasterImage sourceImage, int boundry, string firstDocSavePath, string lastDocSavePath);

Description:

Split the input mulipage file into two with specified index boundary and save them to the given file path.

Parameters:

Name	Description	Valid Value
sourceImage	the input multipage file	a valid RasterImage object
boundry	page index boundary for split	0 to page count - 1
firstDocSavePath	first output file save path	-
lastDoc	second output file save path	-

Delete pages

Remove pages from multipage document.

```
C#
//load a multipage tif file
RasterImage tifDoc = new RasterImage(@"F:\7Pages.tif");
//specify which page to delete from the file.
//actually the page1, page4, page5 will be removed from the tif file.
int[] index = new int[] { 0, 3, 4 };
//set the output file path
String savePath = @"F:\output.tif";
MultiPageImageProcess.DeletePages(tifDoc, index, savePath);
```

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

public static void DeletePages(RasterImage sourceImage, int[] pageIndedx, string savePath);

Description:

Remove specified pages from the input multipage document and save it to the given path.

Parameters：

Name	Description	Valid Value
sourceImage	the input multipage document	a valid RasterImage object
pageIndedx	page index of deleting page	0 to page count − 1
savePath	output file path	-

Extract pages

Extract pages from the input multipage file.

C#

```csharp
//load a multipage tif file
RasterImage tifDoc = new RasterImage(@"F:\7Pages.tif");
//specify which page will be extracted.
int[] pageIdnex = new int[]{0,3,5};
//extract the pages and they will be returned as a RasterImage object.
//actually, the page1, page4 and page6 will be extracted.
RasterImage output = MultiPageImageProcess.ExtractPages(tifDoc, pageIdnex);
```

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

```csharp
public static RasterImage ExtractPages(RasterImage sourceImage, int[] pageIndexs);
```

Description:

Extract specified pages from input file and returned them as a **RasterImage** object.

Parameters:

Name	Description	Valid Value
sourceImage	the input file	a valid RasterImage object
pageIndexs	page index of extracting pages	0 to page count - 1

Return:

RasterImage object made up of the extracted pages.

Reorder pages

Modify page orders of multipage file.

C#

```csharp
//input a multipage tif file
RasterImage tifDoc = new RasterImage(@"F:\4Pages.tif");
//specify the new order
int[] newOrders = new int[] { 0, 2, 3, 1 };
//ensure the output format support multipage
String savePath = @"F:\output.tif";
MultiPageImageProcess.ReOrderPages(tifDoc, newOrders, savePath);
```

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

```csharp
public static void ReOrderPages(RasterImage sourceImage, int[] pageIndex, string savePath);
```

Description:

Modify the page orders of multipage file.

Parameters:

Name	Description	Valid Value
sourceImage	input multipage file	a valid RasterImage object
pageIndex	new order of the output file	0 to pagecount - 1
savePath	output file path	-

Insert page

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

public static void InsetPage(RasterImage sourceImage, RasterImage newPage, int inserPageIndex, string savePath);

Description:

Insert an image or multipage document to the input file with specified location.

Parameters:

Name	Description	Valid Value
sourceImage	input file	a valid RasterImage object
newPage	file for inserting	a valid RasterImage object
inserPageIndex	location of the insert page	0 to sourceImage's page count - 1
savePath	output file path	-

Special Effect

Adaptive blur

```csharp
C#
//load a RasterImage object
RasterImage image = new RasterImage(@"F:\input.png");
//assign a processor to the image
ImageProcess processor = new ImageProcess(image);
//set Gaussian radius to 0, standard deviation of the Laplacian to 0.5
processor.AdaptiveBlurImage(0, 0.5);
```

Related API(s)(**ImageProcess.cs**)

public void AdaptiveBlurImage(double radius, double sigma);

Description:

Adaptive all pages of the **RasterImage** object

Parameters:

Name	Description	Valid Value
radius	the radius of the Gaussian, in pixels.The higher the value, the smoother the image	>=0
sigma	the standard deviation of the Laplacian, in pixels.	default: 0

public void AdaptiveBlurImage(double radius, double sigma, int pageIndex);

Descirption:

Adaptive blur the specified page of the **RasterImage** object.

Parameters:

Name	Description	Valid Value
radius	the radius of the Gaussian, in pixels.The higher the value, the smoother the image	>=0
sigma	the standard deviation of the Laplacian, in pixels.	default: 0
pageIndex	page index	0 to page count - 1

Adaptive resize

Fast resize image. When differs between target width/height and original width/height less than 50 pixles.

Related API(s)(**ImageProcess.cs**)

public void AdaptiveResizeImage(int width, int height);

Description:

Fast resize all pages of the **RasterImage** object.

Parameters:

Name	Description	Valid Value
width	output file width	>0
height	output file height	>0

public void AdaptiveResizeImage(int width, int height, int pageIndex);

Description:

Fast resize the specified page of the **RasterImage** object.

Parameters:

Name	Description	Valid Value
width	output file width	>0
height	output file height	>0
pageIndex	page index	0 to page count -1

Adaptive threshold

Threshold image. Evaulating the mean of a the specified region (by width and height),and using the mean as the thresholding value.In order to remove residual noise from the background, the threshold may be adjusted by subtracting a constant offset (default zero) from the mean to compute the threshold.

Related API(s)(**ImageProcess.cs**)

public void AdaptiveThresholdImage(int width, int height, int offset);

Description:

Adaptive threshold all pages of the **RasterImage** object.

Parameters:

Name	Description	Valid Value
width	the specified region's width	>0
height	the specified region's height	>0
offset	for adjust the threshold value	default: 0

public void AdaptiveThresholdImage(int width, int height, int offset, int pageIndex);

Description:

Adaptive threshold the specified page of **RasterImage** object.

Parameters:

Name	**Description**	**Valid Value**
width	the specified region's width	>0
height	the specified region's height	>0
offset	for adjust the threshold value	default: 0
pageIndex	page index	0 to page count - 1

Add noise

Add specified type of noise to the image.

```csharp
C#
//load a RasterImage object
RasterImage image = new RasterImage(@"F:\input.png");
//assign a processor to the image
ImageProcess processor = new ImageProcess(image);
//Adaptive Gaussian noise to the image.
processor.AddNoiseImage(NoiseType.GaussianNoise);
```

Related API(s)(**ImageProcess.cs**)

public void AddNoiseImage(NoiseType noise);

Description:

Add specified type of noise on all pages of the **RasterImage** object.

Parameters:

Name	**Description**	**Valid Value**
noise	noise's type	choose from enum **NoiseType**

public void AddNoiseImage(NoiseType noise, int pageIndex);

Description:

Add nosie on the specified page of the **RasterImage** object.

Parameters:

Name	Description	Valid Value
noise	noise's type	choose from enum **NoiseType**
pageIndex	page index	0 to page count - 1

Auto level

Scale the minimum and maximum color values to the full quantum range.

C#

```csharp
//load a RasterImage object
RasterImage image = new RasterImage(@"F:\1.png");
//assign a processor to the image
ImageProcess processor = new ImageProcess(image);
//Auto level the image
processor.AntoLevelImage();
```

Related API(s)(**ImageProcess.cs**)

public void AntoLevelImage();

Description:

Auto level all pages of the **RasterImage** object.

public void AntoLevelImage(int pageIndex);

Description:

Auto level the specified page of **RasterImage** object.

Parameters:

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

Append

Connect all pages of a multipage document from top to bottom(left to right) together.

C#

```csharp
//load a multipage file as RasterImage object
RasterImage image = new RasterImage(@"F:\input.tif");
//assign a processor to the RasterImage object
ImageProcess processor = new ImageProcess(image);
//append all pages.
RasterImage output = processor.AppendImages(1);
output.Save(@"F:\output.png");
```

Related API(s)(**ImageProcess.cs**)

public RasterImage AppendImages(int direction);

Description:

Connect a multipage file's pages together.

Parameters:

Name	Description	Valid Value
direction	direction of the connection	0: left to right 1: top to bottom

Return:

Null if failed.

```C#
//load a RasterImage object
RasterImage tifDoc = new RasterImage(@"F:\input.tif");
//input a second image
RasterImage image = new RasterImage(@"F:\input.png");
//assign a processor to the RasterImage object
ImageProcess processor = new ImageProcess(tifDoc);
//connect tifDoc's pages and image's pages all together.
RasterImage output = processor.AppendImages(image, 1);
output.Save(@"F:\output.png");
```

Related API(s)(**ImageProcess.cs**)

public RasterImage AppendImages(RasterImage raster, int direction);

Description:

Connect **RasterImage** object itself's pages and the input object's pages together.

Parameters:

Name	Description	Valid Value
raster	the input RasterImage object	can't be null
direction	direction of the connection	0: left to right 1: top to bottom

Return:

Null if failed

public RasterImage AppendImages(RasterImage[] rasters, int direction)

Description:

Connect **RasterImage** object itself's pages and the input objects' pages together.

Parameters:

Name	Description	Valid Value
rasters	the input RasterImage objects	can't be null
direction	direction of the connection	0: left to right 1: top to bottom

Return:

Null if failed

Auto gamma

Gamma correction, use the mean of the image to adjust the image.

C#

```
//load a RasterImage object
RasterImage image = new RasterImage(@"F:\input.png");
//assign a processor to the RasterImage object
ImageProcess processor = new ImageProcess(image);
//Gamma correction
processor.AutoGammaImage();
image.Save(@"F:\output.png");
```

Related API(s)(**ImageProcess.cs**)

public void AutoGammaImage();

Description:

Auto gamma correction for all pages of **RasterImage** object.

public void AutoGammaImage(int pageIndex);

Description:

Auto gamma correction for specified page of **RasterImage** object.

Parameters:

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

Auto orient

Adjust the orientation of the image for best viewing.

C#

```
//load a RasterImage object
RasterImage image = new RasterImage(@"F:\input.png");
//assign a processor to the RasterImage object
ImageProcess processor = new ImageProcess(image);
//orientation correction
processor.AutoOrientImage();
image.Save(@"F:\output.png");
```

Related API(s)(**ImageProcess.cs**)

public void AutoOrientImage();

Description:

Adjust all pages' orientation of **RasterImage** object for best viewing.

public void AutoOrientImage(int pageIndex);

Description:

Adjust the specified page's orientation of **RasterImage** object for best viewing.

Parameters:

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

Apply median filter to image

Bilevel

```csharp
C#
//load a RasterImage object
RasterImage image = new RasterImage(@"F:\input.png");
//assign a processor to the RasterImage object
ImageProcess processor = new ImageProcess(image);
//bilevel
processor.BilevelImage(0.5);
image.Save(@"F:\output.png");
```

Related API(s)(**ImageProcess.cs**)

public void BilevelImage(double threshold);

Description:

Image's bilevel all pages of the **RasterImage** object.

Parameters:

Name	Description	Valid Value
threshold	the color value below threshold will be set to 0 and above it will be set to 255	0 to 255

public void BilevelImage(double threshold, int pageIndex)

Description:

Image's bilevel for the specified page of the **RasterImage** object.

Parameters:

Name	Description	Valid Value
threshold	the color value below threshold will be set to 0 and above it will be set to 255	0 to 255
pageIndex	page index	0 to page count - 1

Blueshift

Simulate a scene at nighttime in the moonlight.

```
//load a RasterImage object
RasterImage image = new RasterImage(@"F:\input.png");
//assign a processor to the RasterImage object
ImageProcess processor = new ImageProcess(image);
//blue shift
processor.BlueShiftImage(1.5f);
image.Save(@"F:\output.png");
```

Related API(s)(**ImageProcess.cs**)

public void BlueShiftImage(double factor);

Description:

Simulate a scene at nighttime in the moonlight for all pages of **RasterImagae** object

Parameters:

Name	Description	Valid Value
factor	blue shift factor	default:1.5f

public void BlueShiftImage(double factor, int pageIndex);

Description:

Simulate a scene at nighttime in the moonlight for the specified page of the **RasterImage** object..

Parameters:

Name	Description	Valid Value
factor	blue shift factor	default:1.5f
pageIndex	page index	0 to page count - 1

Blur

Blur the image

C#

```
//load a RasterImage object
RasterImage image = new RasterImage(@"F:\input.png");
//assign a processor to the RasterImage object
ImageProcess processor = new ImageProcess(image);
//image blur set the radius to 0(default),set sigma to 1.
processor.BlurImage(0,1);
image.Save(@"F:\output.png");
```

Related API(s)(**ImageProcess.cs**)

public void BlurImage(double radius, double sigma);

Description:

Apply to all pages of **RasterImage** object.

Parameters:

Name	Description	Valid Value
radius	the radius of the Gaussian, in pixels.The higher the value, the smoother the image	>=0
sigma	the standard deviation of the Laplacian, in pixels	default:0

public void BlurImage(double radius, double sigma, int pageIndex)

Description:

Blur the specifed page of **RasterImage** object.

Parameters:

Name	Description	Valid Value
radius	the radius of the Gaussian, in pixels.The higher the value, the smoother the image	>=0
sigma	the standard deviation of the Laplacian, in pixels	default: 0
pageIndex	page index	0 to page count - 1

Buttonize

darken or lighten the edge of the image to simulate a 3d botton effect

```csharp
C#
//load a RasterImage object
RasterImage image = new RasterImage(@"F:\input.png");
//assign a processor to the RasterImage object
ImageProcess processor = new ImageProcess(image);
//set the button edge's width and height to 10 pixels.
processor.ButtonizeImage(10, 10, false);
image.Save(@"F:\output.png");
```

Related API(s)(**ImageProcess.cs**)

public void ButtonizeImage(int width, int height, bool flag);

Description:

A button effect for all the pages of the **RasterImage** object.

Parameters:

Name	Description	Valid Value
width	edge's width	>0
height	edge's height	>0
flag	lighten or darken the edge	true: lighten false: darken

public void ButtonizeImage(int width, int height, bool flag, int pageIndex)

Description:

A button effect for the specified page of the **RasterImage** object.

Parameters:

Name	Description	Valid Value
width	edge's width	>0
height	edge's height	>0
flag	lighten or darken the edge	true: lighten false: darken
pageIndex	page index	0 to page count - 1

CannyEdge

Detect the edges in image, produces a very strong (binary) single pixel wide lines at all sharp edges, with very little noise interferance.

C#

```
//load a RasterImage object
RasterImage image = new RasterImage(@"F:\input.png");
//assign a processor to the RasterImage object
ImageProcess processor = new ImageProcess(image);
//detect the edge in image.
processor.CannyEdgeImage(0,1,0.1,0.3);
image.Save(@"F:\output.png");
```

Related API(s)(**ImageProcess.cs**)

public void CannyEdgeImage(double radius, double sigma, double lowerPercent, double upperPercent)

Description:

Detect the edges in all pages of **RasterImage** object

Parameters:

Name	Description	Valid Value
radius	the radius of the Gaussian, in pixels. The higher the value, the smoother the image	>=0
sigma	the standard deviation of the Laplacian, in pixels	default:0
lowerPercent	Percentage of edge pixels in the lower threshold	0.0 to 1.0
upperPercent	Percentage of edge pixels in the upper threshold	0.0 to 1.0

public void CannyEdgeImage(double radius, double sigma, double lowerPercent, double upperPercent, int pageIndex);

Description:

Detect the edges in first page of the **RasterImage** object.

Parameters:

Name	Description	Valid Value
radius	the radius of the Gaussian, in pixels.The higher the value, the smoother the image	>=0
sigma	the standard deviation of the Laplacian, in pixels	default:0
lowerPercent	Percentage of edge pixels in the lower threshold	0.0 to 1.0
upperPercent	Percentage of edge pixels in the upper threshold	0.0 to 1.0
pageIndex	page index	0 to page count - 1

Charcoa

Simulate a charcoal sketch

```csharp
C#
//load a RasterImage object
RasterImage image = new RasterImage(@"F:\input.png");
//assign a processor to the RasterImage object
ImageProcess processor = new ImageProcess(image);
//simulate a charcoal sketch
processor.CharcoaImage(0, 1);
image.Save(@"F:\output.png");
```

Related API(s)(**ImageProcess.cs**)

public void CharcoaImage(double radius, double sigma);

Description:

Simulate all the pages of the **RasterImage** object to the charcoal sketch.

Parameters:

Name	Description	Valid Value
radius	the radius of the Gaussian, in pixels.The higher the value, the smoother the image	>=0
sigma	the standard deviation of the Laplacian, in pixels	default: 0

public void CharcoaImage(double radius, double sigma, int pageIndex);

Description:

Simulate the first page of the **RasterImage** object to the charcoal sketch.

Parameters:

Name	Description	Valid Value
radius	the radius of the Gaussian, in pixels.The higher the value, the smoother the image	>=0
sigma	the standard deviation of the Laplacian, in pixels	default:0
pageIndex	page index	0 to page count - 1

Change image's contrast

Enhances or reduce the intensity differences between the lighter and darker elements of the image

```csharp
//load a RasterImage object
RasterImage image = new RasterImage(@"F:\input.png");
//assign a processor to the RasterImage object
ImageProcess processor = new ImageProcess(image);
//reduce contrast of the image
processor.ContrastImage(false);
image.Save(@"F:\output.png");
```

Related API(s)(**ImageProcess.cs**)

public void ContrastImage(bool sharpen)

Description:

Enhance or reduce the contrast of all the pages.

Parameters:

Name	Description	Valid Value
sharpen	enhance or reduce the contrast of the image	false: reduce true: enhance

public void ContrastImage(bool sharpen, int pageIndex);

Description:

Enhance or reduce the specified page of the **RasterImage** object.

Parameters:

Name	Description	Valid Value
sharpen	enhance or reduce the contrast of the image	false: reduce true: enhance
pageIndex	page index	0 to page count - 1

Change image's brightness and contract

Change the image's brightness and contract.

```csharp
//load a RasterImage object
RasterImage image = new RasterImage(@"F:\input.png");
//assign a processor to the RasterImage object
ImageProcess processor = new ImageProcess(image);
processor.BrightnessContrastImage(50,50);
image.Save(@"F:\output.png");
```

Related API(s)(**ImageProcess.cs**)

public void BrightnessContrastImage(double brightbess, double contrast)

Description:

Apply to all pages of **RasterImage** object

Parameters:

Name	Description	Valid Value
brightbess	image's brightness	-100 to 100 0:no change <0:decrease >0:increase
contrast	image's contrast	-100 to 100 0:no change <0:decrease >0:increase

public void BrightnessContrastImage(double brightbess, double contrast, int pageIndex)

Description:

Apply to the specified page of **RasterImage** object

Parameters:

Name	Description	Valid Value
brightbess	image's brightness	-100 to 100 0:no change <0:decrease >0:increase
contrast	image's contrast	-100 to 100 0:no change <0:decrease >0:increase
pageIndex	page index	0 to page count - 1

Chrominance sub sample

Chroma Downsampling, Specify the SubSample Type For Jpeg Encoder, Use Default Compress Quality 85

```csharp
C#
//load a RasterImage object
RasterImage image = new RasterImage(@"F:\input.jpg");
//assign a processor to the RasterImage object
ImageProcess processor = new ImageProcess(image);
//set jpeg file's sub sample type
processor.ChrominanceSubSample(SubSampleType.SubSample411);
image.Save(@"F:\output.jpg")
```

Related API(s)(**ImageProcess.cs**)

public void ChrominanceSubSample(SubSampleType subSampleType);

Description:

Jpeg file Chroma Downsampling.

Parameters:

Name	Description	Valid Value
subSampleType	type of sub sample	choose from the enum SubSampleType. e.g. NoneSubSample = 4:4:4　SubSample422 = 4:2:2

Colorize

Compare

Compare two images. The different part will be marked to red.

```
C#
//load a RasterImage object
RasterImage image = new RasterImage(@"F:\input.png");
RasterImage compareWith = new RasterImage(@"F:\input_1.png");
//assign a processor to the RasterImage object
ImageProcess processor = new ImageProcess(image);
//compare two images.
processor.CompareImage(compareWith);
image.Save(@"F:\output.png");
```

Related API(s)(**ImageProcess.cs**)

public void CompareImage(RasterImage compareWith);

Description:

Compare all pages of **RasterImage** object with the input one.

Parameters:

Name	Description	Valid Value
compareWith	the image to compare with	can't be null

public void CompareImage(RasterImage compareWith, int pageIndex);

Description:

Get the specified page of the **RasterImage** object, and compare with the input one.

Parameters:

Name	Description	Valid Value
compareWith	the image to compare with	can't be null
pageIndex	page index	0 to page count - 1

Cropping

Image cropping

C#

```
//load a RasterImage object
RasterImage image = new RasterImage(@"F:\input.png");
//assign a processor to the RasterImage object
ImageProcess processor = new ImageProcess(image);
//set the cropping area.
Rectangle cropArea = new Rectangle(0, 0, 50, 50);
processor.CropImage(cropArea);
image.Save(@"F:\output.png");
```

Related API(s)(**ImageProcess.cs**)

```
public void CropImage(Rectangle rect);
```

Description:

Crop all pages of **RasterImage** object with the specified rectangle.

Parameters:

Name	Description	Valid Value
rect	the cropping area	can't be null

```
public void CropImage(Rectangle rect, int pageindex);
```

Description:

Crop the specified page of **RasterImage** object with the specified rectangle.

Parameters:

Name	Description	Valid Value
rect	the cropping area	can't be null
pageindex	page index	0 to page count - 1

Cycle colormap

Displace image colormap by amount.

C#

```
//load a RasterImage object
RasterImage image = new RasterImage(@"F:\input.png");
//assign a processor to the RasterImage object
ImageProcess processor = new ImageProcess(image);
//set the crop area.
processor.CycleColormapImage(1);
image.Save(@"F:\output.png");
```

Related API(s)(**ImageProcess.cs**)

```
public void CycleColormapImage(int amount);
```
Description:

Displace all pages' colormap of **RasterImage** object with specifed amount.

Parameters:

Name	Description	Valid Value
amount	the number of positions each colormap entry is shifted	>0

```
public void CycleColormapImage(int amount, int pageIndex);
```
Description:

Displace the sepcified page's colormap of **RasterImage** object with given amount.

Parameters:

Name	Description	Valid Value
amount	the number of positions each colormap entry is shifted	>0
pageIndex	page index	0 to page count - 1

Decipher

Decipher image with the pass phrase, only be used in concert with API EnCipherImage. That is you should use the same phrase to decipher and encipher it.

C#

```
//load a RasterImage object
RasterImage image = new RasterImage(@"F:\input.png");
//assign a processor to the RasterImage object
ImageProcess processor = new ImageProcess(image);
//Encipher the image with the String
processor.DecipherImage ("Hello");
image.Save(@"F:\output.png");
```

Related API(s)(**ImageProcess.cs**)

```
public void DecipherImage(string passphrase);
```
Description:

Decipher all pages of **RasterImage** object with the pass phrase.

Parameters:

Name	Description	Valid Value
passphrase	a string to decipher the image	must be the same one when encipher the image.

```
public void DecipherImage(string passphrase, int pageIndex);
```
Description:

Decipher the specified page of **RasterImage** object with the pass phrase.

Parameters:

Name	Description	Valid Value
passphrase	a string to decipher the image	must be the same one when encipher the image.
pageIndex	page index	0 to page count - 1

Deskew

Straighten the image

```csharp
//load a RasterImage object
RasterImage image = new RasterImage(@"F:\input.png");
//assign a processor to the RasterImage object
ImageProcess processor = new ImageProcess(image);
//deskew the image, recommend value:0.4
processor.DeskewImage(0.4);
image.Save(@"F:\output.png");
```

Related API(s)(**ImageProcess.cs**)

public void DeskewImage(double threshold);

Description:

Deskew all pages of **RasterImage** object with specified threshold.

Parameters:

Name	Description	Valid Value
threshold	threshold value	recommend value: 0.4

public void DeskewImage(double threshold, int pageIndex);

Description:

Deskew the specified page of **RasterImage** object with specified threshold.

Parameters:

Name	Description	Valid Value
threshold	threshold value	recommend value: 0.4
pageIndex	page index	0 to page count - 1

Distort

Effect3D

All these is needed are two pictures are separated by a small horizontal distance, roughly equal to the distance between our eyes. Changing a camera's location is easy, you just take a picture, move the camera and take another picture.

```
//load a RasterImage object
RasterImage left = new RasterImage(@"F:\left.png");
RasterImage right = new RasterImage(@"F:\right.png");
//assign a processor to the RasterImage object
ImageProcess processor = new ImageProcess(left);
processor.Effect3D(right);
left.Save(@"F:\output.png");
```

Related API(s)(**ImageProcess.cs**)

public void Effect3D(RasterImage rightImage);

Description:

Produce three-dimensional image.

Parameters:

Name	Description	Valid Value
rightImage	the same image with RasterImage object's page, just has a small horizontal displacement.	can't be null

public void Effect3D(RasterImage rightImage, int pageIndex);

Description:

Produce three-dimensional image with the specified page of the **RasterImage** object

Parameters:

Name	Description	Valid Value
rightImage	the same image with RasterImage object's specified page, just has a small horizontal displacement.	can't be null
pageIndex	page index	0 to page count - 1

Emboss

Hilight edges within the image.

C#

```
//load a RasterImage object
RasterImage image = new RasterImage(@"F:\input.png");
//assign a processor to the RasterImage object
ImageProcess processor = new ImageProcess(image);
processor.EmbossImage(0,1);
image.Save(@"F:\output.png");
```

Related API(s)(**ImageProcess.cs**)

public void EmbossImage(double radius, double sigma);

Description:

Hilight edges within all the pages of **RasterImage** object.

Parameters:

Name	Description	Valid Value
radius	the radius of the Gaussian, in pixels.The higher the value, the smoother the image	>=0
sigma	the standard deviation of the Laplacian, in pixels	default: 0

public void EmbossImage(double radius, double sigma, int pageIndex);

Description:

Hilight edges within the specified page of **RasterImage** object.

Parameters:

Name	Description	Valid Value
radius	the radius of the Gaussian, in pixels.The higher the value, the smoother the image	>=0
sigma	the standard deviation of the Laplacian, in pixels	default: 0
pageIndex	page index	0 to page count - 1

EnCipher

Encipher the image with phrase

```
C#
//load a RasterImage object
RasterImage image = new RasterImage(@"F:\input.png");
//assign a processor to the RasterImage object
ImageProcess processor = new ImageProcess(image);
//encipher the image with string "Hello"
processor.EnCipherImage("Hello");
image.Save(@"F:\output.png");
```

Related API(s)(**ImageProcess.cs**)

public void EnCipherImage(string passphrase);

Description:

Encipher all the pages of **RasterImage** object with the same phrase

Parameters:

Name	Description	Valid Value
passphrase	the string to encipher the image	can't be null

public void EnCipherImage(string passphrase, int pageIndex);

Description:

Encipher the specified page of **RasterImage** object with the given phrase

Parameters:

Name	Description	Valid Value

passphrase	the string to encipher the image	can't be null
pageIndex	page index	0 to page count - 1

Enhance

use a digital filter to minimize the noise

```csharp
C#

    //load a RasterImage object
RasterImage image = new RasterImage(@"F:\input.png");
//assign a processor to the RasterImage object
ImageProcess processor = new ImageProcess(image);
processor.EnhanceImage();
image.Save(@"F:\output.png");
```

Related API(s)(**ImageProcess.cs**)

public void EnhanceImage();

Description:

Enhance all the pages of **RasterImage** object.

public void EnhanceImage(int pageIndex);

Description:

Enhance the specified page of **RasterImage** object.

Parameters:

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

Equalize

Adjusts contrast without changing the brightness range of the image, thus more detail can be seen in areas that have very similar value.

```csharp
C#

//load a RasterImage object
RasterImage image = new RasterImage(@"F:\input.png");
//assign a processor to the RasterImage object
ImageProcess processor = new ImageProcess(image);
processor.EqualizeImage();
image.Save(@"F:\output.png");
```

Related API(s)(**ImageProcess.cs**)

public void EqualizeImage();

Description:

Equalize all the pages of **RasterImage** object.

```
public void EqualizeImage(int pageIndex);
```

Description:

Equalize the specified page of **RasterImage** object

Parameters:

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

Erase

EraseImage Image to Current BackGround Color, but you should set the property
"BackgroundColor", otherwise the image will be erase to white

C#

```
//load a RasterImage object
RasterImage image = new RasterImage(@"F:\input.png");
//assign a processor to the RasterImage object
ImageProcess processor = new ImageProcess(image);
processor.EraseImage();
image.Save(@"F:\output.png");
```

Related API(s)(**ImageProcess.cs**)

```
public void EraseImage();
```

Description:

Erase all pages of **RasterImage** object to background color.

```
public void EraseImage(int pageIndex);
```

Description:

Erase the specified page of **RasterImage** object to background color.

Parameters:

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

Fill with opaque color

Detect color which matches the "newColor" and replace it with "newColor"

C#

```
//load a RasterImage object
RasterImage image = new RasterImage(@"F:\input.png");
//assign a processor to the RasterImage object
ImageProcess processor = new ImageProcess(image);
processor.FillWithopaqueColor(0,0,Color.FromArgb(0,255,0,0),1,false);
image.Save(@"F:\output.png");
```

Related API(s)(**ImageProcess.cs**)

public void FillWithopaqueColor(int xOffset, int yOffset, Color newColor, double distance, bool invert);

Description:

Replace the color within all pages of **RasterImage** object.

Parameters:

Name	Description	Valid Value
xOffset	x coordinate of start location	>=0 && <= page's width
yOffset	y coordinate of start location	>=0 && <= page's height
newColor	the specified color	can't be null
distance	colors within this distance are considered equal.	>=0
invert	replace color match the given color or not match.	true:replace color which not matched the 'newColor' with 'newColor' false:replace color which matches the 'newColor' with 'newColor'

public void FillWithopaqueColor(int xOffset, int yOffset, Color newColor, double distance, int pageIndex, bool invert);

Description:

Replace the color within the specified page of **RasterImage** object.

Parameters:

Name	Description	Valid Value
xOffset	x coordinate of start location	>=0 && <= page's width
yOffset	y coordinate of start location	>=0 && <= page's height
newColor	the specified color	can't be null
distance	colors within this distance are considered equal.	>=0
pageIndex	page index	0 to page count - 1
invert	replace color match the given color or not match.	true:replace color which not matched the 'newColor' with 'newColor' false:replace color which matches the 'newColor' with 'newColor'

Fill with transparent color

Detect color which matches the "newColor" and replace it with transparent color.

Related API(s)(**ImageProcess.cs**)

public void FillWithTransparentColor(int xOffset, int yOffset, Color newColor, double distance, bool invert)

Description:

Replace all pages' color.

Parameters:

Name	Description	Valid Value
xOffset	x coordinate of start location	>=0 && <=page's width
yOffset	y coordinate of start location	>=0 && <=page's height
newColor	the specified color	can't be null
distance	colors within this distance are considered equal.	>=0
invert	replace the color matches the given color or not matches.	true: replace color which not matches the given color. false: replace color which matches the given color

public void FillWithTransparentColor(int xOffset, int yOffset, Color newColor, double distance, int pageIndex, bool invert)

Replace the specified pages' color.

Parameters:

Name	Description	Valid Value
xOffset	x coordinate of start location	>=0 && <=page's width
yOffset	y coordinate of start location	>=0 && <=page's height
newColor	the specified color	can't be null
distance	colors within this distance are considered equal.	>=0
pageIndex	page index	0 to page count - 1
invert	replace the color matches the given color or not matches.	true: replace color which not matches the given color. false: replace color which matches the given color

Flip

Get the image upside down.

```
C#

//load a RasterImage object
RasterImage image = new RasterImage(@"F:\input.png");
//assign a processor to the RasterImage object
ImageProcess processor = new ImageProcess(image);
processor.FlipImage();
image.Save(@"F:\output.png");
```

Related API(s)(**ImageProcess.cs**)

public void FlipImage();

Description:

Upsidedown all pages of **RasterImage** object.

public void FlipImage(int pageIndex);

Description:

Upsidedown the specified page of **RasterImage** object.

Parameters:

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

Flop

Mirror the image.

```
C#

//load a RasterImage object
RasterImage image = new RasterImage(@"F:\input.png");
//assign a processor to the RasterImage object
ImageProcess processor = new ImageProcess(image);
processor.FlopImage();
image.Save(@"F:\output.png");
```

Related API(s)(**ImageProcess.cs**)

public void FlopImage();

Description:

Mirror all pages of **RasterImage** object.

public void FlopImage(int pageIndex);

Description:

Mirror the specified page of **RasterImage** object.

Parameters:

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

Frame

Add decorative frame around image

C#

```
//load a RasterImage object
RasterImage image = new RasterImage(@"F:\input.png");
//assign a processor to the RasterImage object
ImageProcess processor = new ImageProcess(image);
processor.FrameImage(10,10,0,10);
image.Save(@"F:\output.png");
```

Related API(s)(**ImageProcess.cs**)

public void FrameImage(int width, int height, int innerBevel, int outerBevel);

Description:

Decorate all pages of **RasterImage** object.

Parameters:

Name	Description	Valid Value
width	frame's width	>0
height	frame's height	>0
innerBevel	frame's inner bevel	>=0
outerBevel	frame's outer bevel	>=0

public void FrameImage(int width, int height, int innerBevel, int outerBevel, int pageIndex);

Description:

Decorate the specified page of **RasterImage** object.

Parameters:

Name	Description	Valid Value
width	frame's width	>0
height	frame's height	>0
innerBevel	frame's inner bevel	>=0
outerBevel	frame's outer bevel	>=0
pageIndex	page index	0 to page count - 1

Gamma

Gamma correction: adjust color difference displayed on differnt display monitor

Related API(s)(**ImageProcess.cs**)

public void GammaImage(double gamma);

Description:

Corrcet all pages of **RasterImage** object.

Parameters:

Name	Description	Valid Value
gamma	gamma factor	<1.0: darken the image >=1.0: lighten the image

public void GammaImage(double gamma, int pageIndex);

Description:

Corrcet the specified of **RasterImage** object.

Parameters:

Name	Description	Valid Value
gamma	gamma factor	<1.0: darken the image >=1.0: lighten the image
pageIndex	page index	0 to page count - 1

Gaussian blur

Blur the image.

Related API(s)(**ImageProcess.cs**)

public void GaussianBlurImage(double radius, double gamma);

Description:

Blur all pages of **RasterImage** object.

Parameters:

Name	Description	Valid Value
radius	number of neighbor pixels to be included in the convolution mask	the standard is 3
gamma	standard deviation of the gaussian bell curve	<1.0: darken the image >=1.0: lighten the image

public void GaussianBlurImage(double radius, double gamma, int pageIndex);

Description:

Blur the specified page of **RasterImage** object.

Parameters:

Name	Description	Valid Value
radius	number of neighbor pixels to be included in the convolution mask	the standard is 3
gamma	standard deviation of the gaussian bell curve	<1.0: darken the image >=1.0: lighten the image
pageIndex	page index	0 to page index

Grayscale

Convert image to gray scale.

```
C#
//load a RasterImage object
RasterImage image = new RasterImage(@"F:\input.png");
//assign a processor to the RasterImage object
ImageProcess processor = new ImageProcess(image);
processor.GrayscaleImage(PixelIntensityMethod.AveragePixelIntensityMethod);
image.Save(@"F:\output.png");
```

Related API(s)(**ImageProcess.cs**)

public void GrayscaleImage(PixelIntensityMethod method);

Description:

Convert all pages of **RasterImage** object to gray scale.

Parameters:

Name	Description	Valid Value
method	the method to conver it	choose from the enum**PixelIntensityMethod**

public void GrayscaleImage(PixelIntensityMethod method, int pageIndex);

Description:

Convert the specified page of **RasterImage** object to gray scale.

Parameters:

Name	Description	Valid Value
method	the method to conver it	choose from the enum **PixelIntensityMethod**
pageIndex	page index	0 to page count - 1

HoughLine

Implode

Simulate a implode effect.

```csharp
//load a RasterImage object
RasterImage image = new RasterImage(@"F:\input.png");
//assign a processor to the RasterImage object
ImageProcess processor = new ImageProcess(image);
processor.ImplodeImage(0.5);
image.Save(@"F:\output.png");
```

Related API(s)(**ImageProcess.cs**)

public void ImplodeImage(double factor);

Description:

Apply to all the page of **RasterImage** object.

Parameters:

Name	Description	Valid Value
factor	the implode factor	>=0

public void ImplodeImage(double factor, int pageIndex);

Description:

Apply to the specified page of **RasterImage** object.

Parameters:

Name	Description	Valid Value
factor	the implode factor	>=0
pageIndex	page index	0 to page count - 1

JPEG smoothing

Smooth the JPEG file.

C#

```csharp
//load a RasterImage object
RasterImage image = new RasterImage(@"F:\input.jpg");
//assign a processor to the RasterImage object
ImageProcess processor = new ImageProcess(image);
processor.JPEGSmoothing();
image.Save(@"F:\output.jpg");
```

Related API(s)(**ImageProcess.cs**)

public void JPEGSmoothing();

Kuwahara

Apply an edge preserving noise filter to the image.

C#

```csharp
//load a RasterImage object
RasterImage image = new RasterImage(@"F:\input.png");
//assign a processor to the RasterImage object
ImageProcess processor = new ImageProcess(image);
processor.KuwaharaImage(0,1);
image.Save(@"F:\output.png");
```

Related API(s)(**ImageProcess.cs**)

public void KuwaharaImage(double radius, double sigma);

Description:

Apply an edge preserving noise filter to all pages of the **RasterImage** object

Parameters:

Name	Description	Valid Value
radius	the radius of the Gaussian, in pixels.The higher the value, the smoother the image	>=0
sigma	the standard deviation of the Laplacian, in pixels	default: 0

public void KuwaharaImage(double radius, double sigma, int pageIndex);

Description:

Apply an edge preserving noise filter to the specified page of **RasterImage** object

Parameters:

Name	Description	Valid Value
radius	the radius of the Gaussian, in pixels.The higher the value, the smoother the image	>=0
sigma	the standard deviation of the Laplacian, in pixels	default: 0
pageIndex	page index	0 to page count - 1

Layer merge

Merge multi layers into one with specified merge method.

```csharp
C#
//load a RasterImage object
RasterImage image = new RasterImage(@"F:\input.png");
RasterImage inputLayer = new RasterImage(@"F:\input_1.png");
//assign a processor to the RasterImage object
ImageProcess processor = new ImageProcess(image);
processor.LayerMerge(inputLayer, LayerMergeMethod.CompositeLayer);
image.Save(@"F:\output.png");
```

Related API(s)(**ImageProcess.cs**)

public void LayerMerge(RasterImage layer, LayerMergeMethod method);

Description:

All pages of **RasterImage** object will merge with the input layer.

Parameters:

Name	Description	Valid Value
layer	a RasterImage object	can not be null
method	merge method	choose from the eum **LayerMergeMethod**

public void LayerMerge(RasterImage[] layers, LayerMergeMethod method);

Description:

Specify a page of **RasterImage** object to merge with the input layers

Parameters:

Name	Description	Valid Value
layers	RasterImage objects	can not be null
method	merge method	choose from the eum **LayerMergeMethod**

Level

Adjust the levels of the image by scaling the colors falling between specified white and black points to the full available quantum range

Related API(s)(**ImageProcess.cs**)

public void LevelImage(double blackPoint, double whitePoint, double gamma);

Description:

Level all pages of **RasterImage** object.

Parameters:

Name	Description	Valid Value
blackPoint	black-out at most blackPoint % pixels	0 to 1
whitePoint	white-out at most 100% minus whitePoint % pixels.	0 to 1
gamma	gamma correction	<1.0: darken the image >=1.0: lighten the image

public void LevelImage(double blackPoint, double whitePoint, double gamma, int pageIndex);

Description:

Level the spciefied page of **RasterImage** object.

Parameters:

Name	Description	Valid Value
blackPoint	black-out at most blackPoint % pixels	0 to 1
whitePoint	white-out at most 100% minus whitePoint % pixels.	0 to 1
gamma	gamma correction	<1.0: darken the image >=1.0: lighten the image
pageIndex	page index	0 to page count - 1

Linear stretch

Work by counting pixels from the ends of the histogram to get the graylevels to stretch. Discards any pixels below the black point and above the white point, and levels the remaining pixels.

Related API(s)(**ImageProcess.cs**)

public void LinearStretchImage(double blackPoint, double whitePoint);

Description:

Liner stretch all pages of **RasterImage** object.

Parameters:

Name	Description	Valid Value
blackPoint	black-out at most blackPoint % pixels	0 to 1
whitePoint	white-out at most 100% minus whitePoint % pixels.	0 to 1

public void LinearStretchImage(double blackPoint, double whitePoint, int pageIndex);

Description:

Liner stretch the specified page of **RasterImage** object.

Parameters:

Name	Description	Valid Value
blackPoint	black-out at most blackPoint % pixels	0 to 1
whitePoint	white-out at most 100% minus whitePoint % pixels.	0 to 1
pageIndex	page index	0 to page count - 1

Magnify

Double image size

Related API(s)(**ImageProcess.cs**)

public void MagnifyImage();

public void MagnifyImage(int pageIndex);

Description:

Double the specified page's size of **RasterImage** object.

Parameters:

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

Merge

Merge the two images into one with specified merge method.

```
C#
//load a RasterImage object
RasterImage image = new RasterImage(@"F:\input.png");
RasterImage child = new RasterImage(@"F:\child.png");
//assign a processor to the RasterImage object
ImageProcess processor = new ImageProcess(image);
processor.MergeImage(child, MergeType.Atop, 0, 0);
image.Save(@"F:\output.png");
```

Related API(s)(**ImageProcess.cs**)

public void MergeImage(RasterImage childImage, MergeType mergeType, int xOffset, int yOffset);

Description:

All pages of **RasterImage** object will merge with the input one.

Parameters:

Name	Description	Valid Value
childImage	the image to merge with	can't be null
mergeType	the merge method	choose from the enum **MergeType**
xOffset	x coordinate on the background image	>=0 && <= page's width
yOffset	y coordinate on the background image	>=0 && <=page's height

Modulate

Modulate percent hue, saturation, and brightness of an image

```
C#
//load a RasterImage object
RasterImage image = new RasterImage(@"F:\input.png");
//assign a processor to the RasterImage object
ImageProcess processor = new ImageProcess(image);
processor.ModulateImage(150,120,180);
image.Save(@"F:\output.png");
```

Related API(s)(**ImageProcess.cs**)

public void ModulateImage(double brightness, double saturation, double hue);

Description:

Apply to all pages of **RasterImage** object.

Parameters:

Name	Description	Valid Value
brightness	brightness of the image	0 to 200 100 for no change
saturation	saturation of the image	0 to 200 100 for no change
hue	hue of the image	0 to 200 100 for no change

public void ModulateImage(double brightness, double saturation, double hue, int pageIndex);

Description:

Apply to the specified page of **RasterImage** object.

Parameters:

Name	Description	Valid Value
brightness	brightness of the image	0 to 200 100 for no change
saturation	saturation of the image	0 to 200 100 for no change
hue	hue of the image	0 to 200 100 for no change
pageIndex	page index	0 to page count -1

Monochrome

Convert the image to bitonal(black & white)

```csharp
C#

//load a RasterImage object
RasterImage image = new RasterImage(@"F:\input.png");
//assign a processor to the RasterImage object
ImageProcess processor = new ImageProcess(image);
processor.MonochromeImage();
image.Save(@"F:\output.png");
```

Related API(s)(**ImageProcess.cs**)

public void MonochromeImage();

Description:

Convert all pages of **RasterImage** object to bitnal.

public void MonochromeImage(int pageIndex);

Description:

Convert the specified page of **RasterImage** object to bitnal.

Parameters:

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

Morphology

Motion blur

Motion blur the image.

```
C#
//load a RasterImage object
RasterImage image = new RasterImage(@"F:\input.png");
//assign a processor to the RasterImage object
ImageProcess processor = new ImageProcess(image);
processor.MotionBlurImage(0,1,0);
image.Save(@"F:\output.png");
```

Related API(s)(**ImageProcess.cs**)

public void MotionBlurImage(double radius, double sigma, double angle);

Description:

Motion blur all pages of **RasterImage** object.

Parameters:

Name	Description	Valid Value
radius	radius of the Gaussian, in pixels.the higher the value, the smoother the image.	>=0
sigma	standard deviation of the Laplacian, in pixels.	default:1
angle	object appears to be comming from	0: from right

public void MotionBlurImage(double radius, double sigma, double angle, int pageIndex);

Description:

Motion blur the specified page of **RasterImage** object.

Parameters:

Name	Description	Valid Value
radius	radius of the Gaussian, in pixels.the higher the value, the smoother the image.	>=0
sigma	standard deviation of the Laplacian, in pixels.	default:1
angle	object appears to be comming from	0: from right
pageIndex	page index	0 to page count - 1

Negate

Replace image's pixels with its complementary color (white becomes black, yellow becomes blue, etc.)

```csharp
C#
//load a RasterImage object
RasterImage image = new RasterImage(@"F:\input.png");
//assign a processor to the RasterImage object
ImageProcess processor = new ImageProcess(image);
processor.NegateImage(false);
image.Save(@"F:\output.png");
```

Related API(s)(**ImageProcess.cs**)

```csharp
public void NegateImage(bool grayScale);
```

Description:

Negate all page of **RasterImage** object.

Parameters:

Name	Description	Valid Value
grayScale	if only negate gray scale values in image	true: yes false: no

```csharp
public void NegateImage(bool grayScale, int pageIndex);
```

Description:

Negate the specified page of **RasterImage** object.

Parameters:

Name	Description	Valid Value
grayScale	if only negate gray scale values in image	true: yes false: no
pageIndex	page index	0 to page count - 1

Oil paint

Simulate an oil painting

```csharp
C#
//load a RasterImage object
RasterImage image = new RasterImage(@"F:\input.png");
//assign a processor to the RasterImage object
ImageProcess processor = new ImageProcess(image);
processor.OilPaintImage(2);
image.Save(@"F:\output.png");
```

Related API(s)(**ImageProcess.cs**)

public void OilPaintImage(double radius);

Description:

Apply to all pages of **RasterImage** object.

Parameters:

Name	Description	Valid Value
radius	radius of the circular neighborhood	>=0

public void OilPaintImage(double radius, int pageIndex);

Description:

Apply to the specified page of **RasterImage** object.

Parameters:

Name	Description	Valid Value
radius	radius of the circular neighborhood	>=0
pageIndex	page index	0 to page count -1

Optimized huffman

With this method you can get a smaller jpeg file

```
C#
//load a RasterImage object
RasterImage image = new RasterImage(@"F:\input.jpg");
//assign a processor to the RasterImage object
ImageProcess processor = new ImageProcess(image);
processor.OptimizedHuffman(true);
image.Save(@"F:\output.jpg");
```

Related API(s)(**ImageProcess.cs**)

public void OptimizedHuffman(bool useOptimizeHuffmanCode);

Parameters:

Name	Description	Valid Value
useOptimizeHuffmanCode	if use optimized Huffman code	true: use false: not use

Perceptible

Posterize

Simulate a poster effect

```C#
//load a RasterImage object
RasterImage image = new RasterImage(@"F:\input.png");
//assign a processor to the RasterImage object
ImageProcess processor = new ImageProcess(image);
processor.PosterizeImage(3,false);
image.Save(@"F:\output.png");
```

Related API(s)(**ImageProcess.cs**)

public void PosterizeImage(int levels, bool dither);

Description:

Apply to all pages of **RasterImage** object.

Parameters:

Name	Description	Valid Value
levels	number of color levels allowed in each channel	>=0
dither	whether to use dither method	true: use false: not use

public void PosterizeImage(int levels, bool dither, int pageIndex);

Description:

Apply to the specified page of **RasterImage** object.

Parameters:

Name	Description	Valid Value
levels	number of color levels allowed in each channel	>=0
dither	whether to use dither method	true: use false: not use
pageIndex	page index	0 to page count - 1

Quantize image's color

Reduce color number to the specified number.

```C#
//load a RasterImage object
RasterImage image = new RasterImage(@"F:\input.png");
//assign a processor to the RasterImage object
ImageProcess processor = new ImageProcess(image);
processor.QuantizeColor(256,DitherMethod.NoDitherMethod,ColorSpace.RGB);
image.Save(@"F:\output.png");
```

Related API(s)(**ImageProcess.cs**)

public void QuantizeColor(int num, DitherMethod ditherMehod, ColorSpace colorspace);

Description:

Quantize all pages of **RasterImage** object.

Parameters:

Name	Description	Valid Value
num	target number of colors	>=0
ditherMehod	dither method to quantize the image	choose from the enum **DitherMethod**
colorspace	target color space	choose from enum **ColorSpace**

public void QuantizeColor(int num, DitherMethod ditherMehod, ColorSpace colorspace, int pageIndex);

Description:

Quantize the specified page of **RasterImage** object.

Parameters:

Name	Description	Valid Value
num	target number of colors	>=0
ditherMehod	dither method to quantize the image	choose from the enum **DitherMethod**
colorspace	target color space	choose from enum **ColorSpace**
pageIndex	page index	0 to page count -1

Reduce noise

Remove alpha channel

Remove image's alpha channel

```
C#

//load a RasterImage object
RasterImage image = new RasterImage(@"F:\input.png");
//assign a processor to the RasterImage object
ImageProcess processor = new ImageProcess(image);
processor.RemoveAlphaChannel();
image.Save(@"F:\output.png");
```

Related API(s)(**ImageProcess.cs**)

public void RemoveAlphaChannel();

Description:

Remove all pages' alpha channel of **RasterImage** object.

public void RemoveAlphaChannel(int pageIndex);

Description:

Remove the specified page's alpha channel of **RasterImage** object.

Parameters:

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

Replace color

Replace the "old color" with "new color"

```C#
//load a RasterImage object
RasterImage image = new RasterImage(@"F:\input.png");
//assign a processor to the RasterImage object
ImageProcess processor = new ImageProcess(image);
//replace color white with color green.
processor.ReplaceColor(Color.White,Color.Green,3,false);
image.Save(@"F:\output.png");
```

Related API(s)(**ImageProcess.cs**)

public void ReplaceColor(Color oldColor, Color newColor, double distance, bool invert);

Description:

Apply to all pages of **RasterImage** object.

Parameters:

Name	Description	Valid Value
oldColor	the color will be replaced	can't be null
newColor	the colro to replace the 'oldColor'	can't be null
distance	colors within this distance are considered equal.	>=0
invert	use 'newColor' to replace the 'oldColor' or replace colors are not 'oldColor'.	true: replace colors not mathces the 'oldColor' false: replace the 'oldColor'

public void ReplaceColor(Color oldColor, Color newColor, double distance, int pageIndex, bool invert);

Description:

Apply to the specified page of **RasterImage** object.

Parameters:

Name	Description	Valid Value
oldColor	the color will be replaced	can't be null
newColor	the colro to replace the 'oldColor'	can't be null
distance	colors within this distance are considered equal.	>=0
pageIndex	page index	0 to page count - 1
invert	use 'newColor' to replace the 'oldColor' or replace colors are not 'oldColor'.	true: replace colors not mathces the 'oldColor' false: replace the 'oldColor'

Resample

Resize image without color mixing, much faster than Resize API

```csharp
//load a RasterImage object
RasterImage image = new RasterImage(@"F:\input.png");
//assign a processor to the RasterImage object
ImageProcess processor = new ImageProcess(image);
processor.ResampleImage(new Size(500, 500),false);
image.Save(@"F:\output.png");
```

C#

Related API(s)(**ImageProcess.cs**)

public void ResampleImage(Size size, bool aspectRatio);

Description:

Resize all pages of **RasterImage** object to the specified size.

Parameters:

Name	Description	Valid Value
size	the target size	can't be null
aspectRatio	if maintain the aspect ratio	true: preserve the aspect ratio false: not perserve

public void ResampleImage(Size size, bool aspectRatio, int pageIndex);

Description:

Resize the specified page of **RasterImage** object to the target size.

Parameters:

Name	Description	Valid Value
size	the target size	can't be null
aspectRatio	if maintain the aspect ratio	true: preserve the aspect ratio false: not perserve
pageIndex	page index	0 to page count -1

ReSize

Resize image.

```csharp
//load a RasterImage object
RasterImage image = new RasterImage(@"F:\input.png");
//assign a processor to the RasterImage object
ImageProcess processor = new ImageProcess(image);
processor.ReSizeImage(new Size(500, 500), InterpolateMethod.Average16InterpolatePixel);
image.Save(@"F:\output.png");
```

C#

Related API(s)(**ImageProcess.cs**)

public void ReSizeImage(Size size, InterpolateMethod polateMethod);

Description:

Resize all pages of **RasterImage** object to the specified size.

Parameters:

Name	Description	Valid Value
size	the target size	can't be null
polateMethod	interpolate method to resize the image	choose from the enum **InterpolateMethod**

public void ReSizeImage(Size size, InterpolateMethod polateMethod, int pageIndex);

Description:

Resize the specified page of **RasterImage** object to the target size.

Parameters:

Name	Description	Valid Value
size	the target size	can't be null
polateMethod	interpolate method to resize the image	choose from the enum **InterpolateMethod**
pageIndex	page index	0 to page count - 1

Resize image to thumbnail size

Fast thumbnail generation

```C#
//load a RasterImage object
RasterImage image = new RasterImage(@"F:\input.png");
//assign a processor to the RasterImage object
ImageProcess processor = new ImageProcess(image);
processor.ResizeImageToThumbnailSize(new Size(256,256),false);
image.Save(@"F:\output.png");
```

Related API(s)(**ImageProcess.cs**)

public void ResizeImageToThumbnailSize(Size size, bool aspectRatio);

Description:

Resize all the pages of **RasterImage** object to thumbnail size.

Parameters:

Name	Description	Valid Value
size	the target size	can't be null
aspectRatio	if maintain the aspect ratio	true: preserve the aspect ratio false: not preserve

public void ResizeImageToThumbnailSize(Size size, bool aspectRatio, int pageIndex);

Description:

Resize the specified page of **RasterImage** object to thumbnail size.

Parameters:

Name	Description	Valid Value
size	the target size	can't be null
aspectRatio	if maintain the aspect ratio	true: preserve the aspect ratio false: not preserve
pageIndex	page index	0 to page count - 1

Roll

Roll the image with specified rows in the vertical direction or columns in the horizontal direction

```csharp
C#

//load a RasterImage object
RasterImage image = new RasterImage(@"F:\input.png");
//assign a processor to the RasterImage object
ImageProcess processor = new ImageProcess(image);
processor.RollImage(10,10);
image.Save(@"F:\output.png");
```

Related API(s)(**ImageProcess.cs**)

public void RollImage(int xoffset, int yoffset);

Description:

Aplly to all pages of **RasterImage** object.

Parameters:

Name	Description	Valid Value
xoffset	colunms to rolll in the horizonal direction	>=0
yoffset	rows to rolll in the horizonal direction	>=0

public void RollImage(int xoffset, int yoffset, int pageIndex);

Description:

Aplly to the specified page of **RasterImage** object.

Parameters:

Name	Description	Valid Value
xoffset	colunms to rolll in the horizonal direction	>=0
yoffset	rows to rolll in the vertical direction	>=0
pageIndex	page index	0 to page count - 1

Rotate

Rotate image by specified number of degrees.

```csharp
//load a RasterImage object
RasterImage image = new RasterImage(@"F:\input.png");
//assign a processor to the RasterImage object
ImageProcess processor = new ImageProcess(image);
processor.RotateImage(30);
image.Save(@"F:\output.png");
```

Related API(s)(**ImageProcess.cs**)

public void RotateImage(double degree);

Description:

Rotate all pages of **RasterImage** object to the specified degree.

Parameters:

Name	Description	Valid Value
degree	rotate degree	>=0

public void RotateImage(double degree, int pageIndex);

Description:

Rotate the specified page of **RasterImage** object to the target degree.

Parameters:

Name	Description	Valid Value
degree	rotate degree	>=0
pageIndex	page index	0 to page count - 1

Rotational blur

Rotational blur

```csharp
//load a RasterImage object
RasterImage image = new RasterImage(@"F:\input.png");
//assign a processor to the RasterImage object
ImageProcess processor = new ImageProcess(image);
processor.RotationalBlurImage(3);
image.Save(@"F:\output.png");
```

Related API(s)(**ImageProcess.cs**)

public void RotationalBlurImage(double angle);

Description:

Apply to all pages of **RasterImage** object.

Parameters:

Name	Description	Valid Value
angle	the target angle	>=0

public void RotationalBlurImage(double angle, int pageIndex);

Description:

Apply to the specified page of **RasterImage** object.

Parameters:

Name	Description	Valid Value
angle	the target angle	>=0
pageIndex	page index	0 to page count -1

Segment

Selective blur

Blur Image

```csharp
C#
//load a RasterImage object
RasterImage image = new RasterImage(@"F:\input.png");
//assign a processor to the RasterImage object
ImageProcess processor = new ImageProcess(image);
processor.SelectiveBlur(0,1,220);
image.Save(@"F:\output.png");
```

Related API(s)(**ImageProcess.cs**)

public void SelectiveBlur(double radius, double sigma, double threshold);

Description:

Parameters:

Name	Description	Valid Value
radius	radius of the Gaussian, in pixels.The higher the value, the smoother the image.	>=0
sigma	standard deviation of the Gaussian, in pixels	default:0
threshold	pixles less or equal to the threshold will be blured,expressed as percentage	>=0

public void SelectiveBlur(double radius, double sigma, double threshold, int pageIndex);

Description:

Parameters:

Name	Description	Valid Value
radius	radius of the Gaussian, in pixels.The higher the value, the smoother the image.	>=0
sigma	standard deviation of the Gaussian, in pixels	default:0
threshold	pixles less or equal to the threshold will be blured,expressed as percentage	>=0
pageIndex	page index	0 to page count -1

Separate

Separate a specified channel from the image.

Related API(s)(**ImageProcess.cs**)

public void SeparateImage(ChannelType channel);

Description:

Aplly to all pages of **RasterImage** object.

Parameters:

Name	Description	Valid Value
channel	the channel will be extract out	choose from the enum **ChannelType**

public void SeparateImage(ChannelType channel, int pageIndex);

Description:

Apply to the specified page of **RasterImage** object.

Parameters:

Name	Description	Valid Value
channel	the channel will be extract out	choose from the enum **ChannelType**
pageIndex	page index	0 to page count - 1

Sepia tone

Make images like achieved in a photo darkroom

```
//load a RasterImage object
RasterImage image = new RasterImage(@"F:\input.png");
//assign a processor to the RasterImage object
ImageProcess processor = new ImageProcess(image);
processor.SepiaToneImage(0.8);
image.Save(@"F:\output.png");
```

Related API(s)(**ImageProcess.cs**)

public void SepiaToneImage(double threshold);

Description:

Aplly to all pages of **RasterImage** object.

Parameters:

Name	Description	Valid Value
threshold	the sepia tone factor	0 to 1

public void SepiaToneImage(double threshold, int pageIndex);

Description:

Apply to the specified page of **RasterImage** object.

Parameters:

Name	Description	Valid Value
threshold	the sepia tone factor	0 to 1
pageIndex	page index	0 to page count -1

Set delay time

Set gif file's delay time

C#

```
//load a RasterImage object
RasterImage image = new RasterImage(@"F:\input.gif");
//assign a processor to the RasterImage object
ImageProcess processor = new ImageProcess(image);
processor.SetDelayTime(5);
image.Save(@"F:\output.gif");
```

Related API(s)(**ImageProcess.cs**)

public void SetDelayTime(int delay);

Parameters:

Name	Description	Valid Value
delay	delay time in milliseconds	>=0

Set interlace

Set interlace for image.

```
C#
//load a RasterImage object
RasterImage image = new RasterImage(@"F:\input.png");
//assign a processor to the RasterImage object
ImageProcess processor = new ImageProcess(image);
processor.SetImageInterlace(ImageProcess.InterlaceType.PNG);
image.Save(@"F:\output.png");
```

Related API(s)(**ImageProcess.cs**)

public void SetImageInterlace(ImageProcess.InterlaceType interlace);

Parameters:

Name	Description	Valid Value
interlace	interlace type	choose from the enum InterlaceType

Shade

Shadow

Sharpen

Sharpen pixels in image

```
C#
//load a RasterImage object
RasterImage image = new RasterImage(@"F:\input.png");
//assign a processor to the RasterImage object
ImageProcess processor = new ImageProcess(image);
processor.SharpenImage(0, 1);
image.Save(@"F:\output.png");
```

Related API(s)(**ImageProcess.cs**)

public void SharpenImage(double radius, double sigma);

Description:

Sharpen all pages of **RasterImage** object

Parameters:

Name	Description	Valid Value
radius	radius of the Gaussian, in pixels. The higher the value, the smoother the image	>=0
sigma	standard deviation of the Laplacian, in pixels	default:1

public void SharpenImage(double radius, double sigma, int pageIndex);

Description:

Sharpen the specified page of **RasterImage** object

Parameters:

Name	Description	Valid Value
radius	radius of the Gaussian, in pixels. The higher the value, the smoother the image	>=0
sigma	standard deviation of the Laplacian, in pixels	default:1
pageIndex	page index	0 to page count - 1

Shear

Create parallelogram by sliding image by X or Y axis

```
C#

//load a RasterImage object
RasterImage image = new RasterImage(@"F:\input.png");
//assign a processor to the RasterImage object
ImageProcess processor = new ImageProcess(image);
processor.ShearImage(30,30);
image.Save(@"F:\output.png");
```

Related API(s)(**ImageProcess.cs**)

public void ShearImage(double xShearAngle, double yShearAngle);

Description:

Shear all pages of **RasterImage** object

Parameters:

Name	Description	Valid Value
xShearAngle	the number of degrees to shear the image, relative to the X axis	>=0
yShearAngle	the number of degrees to shear the image, relative to the Y axis	>=0

public void ShearImage(double xShearAngle, double yShearAngle, int pageIndex);

Description:

Shear the specified page of **RasterImage** object

Parameters:

Name	Description	Valid Value
xShearAngle	the number of degrees to shear the image, relative to the X axis	>=0
yShearAngle	the number of degrees to shear the image, relative to the Y axis	>=0
pageIndex	page index	0 to page count - 2

Sketch

Simulates a pencil sketch

Related API(s)(**ImageProcess.cs**)

public void SketchImage(double radius, double sigma, double angle);

Description:

Sketch all pages of **RasterImage** object.

Parameters:

Name	Description	Valid Value
radius	radius of the Gaussian, in pixels.The higher the value, the soomther the image	>=0
sigma	tandard deviation of the Laplacian, in pixels	default:1
angle	the sktech angle	>=0

public void SketchImage(double radius, double sigma, double angle, int pageIndex);

Description:

Sketch the specified page of **RasterImage** object.

Parameters:

Name	Description	Valid Value
radius	radius of the Gaussian, in pixels.The higher the value, the soomther the image	>=0
sigma	tandard deviation of the Laplacian, in pixels	default:1
angle	the sktech angle	>=0
pageIndex	page index	0 to page count -1

Solarize

Negate the image's pixels which above the threshold.

```csharp
//load a RasterImage object
RasterImage image = new RasterImage(@"F:\input.png");
//assign a processor to the RasterImage object
ImageProcess processor = new ImageProcess(image);
processor.SolarizeImage(0.5);
image.Save(@"F:\output.png");
```

Related API(s)(**ImageProcess.cs**)

public void SolarizeImage(double factor);

Description:

Solorize all pages of **RasterImage** object.

Parameters:

Name	Description	Valid Value
factor	the threshold value	0 to 255

public void SolarizeImage(double factor, int pageIndex);

Description:

Solorize a specified page of **RasterImage** object.

Parameters:

Name	Description	Valid Value
factor	the threshold value	0 to 255
pageIndex	page index	0 to 255

Splice

Splice the image with the specified rectangle, the rectagle's background can be set with:
process.Background

```csharp
//load a RasterImage object
RasterImage image = new RasterImage(@"F:\input.png");
//assign a processor to the RasterImage object
ImageProcess processor = new ImageProcess(image);
processor.SpliceImage(10,10,20,20);
image.Save(@"F:\output.png");
```

Related API(s)(**ImageProcess.cs**)

public void SpliceImage(int xoffset, int yoffset, int width, int height);

Description:

Splice all pages of **RasterImage** object.

Parameters:

Name	Description	Valid Value
xoffset	start x of rectangle	>=0 && <= page's width
yoffset	start y of rectangle	>=0 && <= page's height
width	rectangle's width	>=0 && <= page's width
height	rectangle's height	>=0 && <= page's height

public void SpliceImage(int xoffset, int yoffset, int width, int height, int pageIndex);

Description:

Splice the specified page of **RasterImage** object.

Parameters:

Name	Description	Valid Value
xoffset	start x of rectangle	>=0 && <= page's width
yoffset	start y of rectangle	>=0 && <= page's height
width	rectangle's width	>=0 && <= page's width
height	rectangle's height	>=0 && <= page's height
pageIndex	page index	0 to page count - 1

Spread

Displace the image's pixles randomly by the specified amount

```csharp
C#
//load a RasterImage object
RasterImage image = new RasterImage(@"F:\input.png");
//assign a processor to the RasterImage object
ImageProcess processor = new ImageProcess(image);
processor.SpreadImage(2);
image.Save(@"F:\output.png");
```

Related API(s)(**ImageProcess.cs**)

public void SpreadImage(int amount);

Description:

Spread all pages of **RasterImage** object.

Parameters:

Name	Description	Valid Value
amount	shift value	>=0

public void SpreadImage(int amount, int pageIndex);

Description:

Spread the specified page of **RasterImage** object.

Parameters:

Name	Description	Valid Value
amount	shift value	>=0
pageIndex	page index	0 to page count -1

Swirl

Rotate the pixels in image with specified degree.

C#

```
//load a RasterImage object
RasterImage image = new RasterImage(@"F:\input.png");
//assign a processor to the RasterImage object
ImageProcess processor = new ImageProcess(image);
processor.SwirlImage(20);
image.Save(@"F:\output.png");
```

Related API(s)(**ImageProcess.cs**)

public void SwirlImage(double degree);

Description:

Swill all pages of **RasterImage** object.

Parameters:

Name	Description	Valid Value
degree	rotate degree	>0

public void SwirlImage(double degree, int pageIndex);

Description:

Swill the specified page of **RasterImage** object.

Parameters:

Name	Description	Valid Value
degree	rotate degree	>0
pageIndex	page index	0 to page count -1

Texture

Modifies the image(texture) to appear in tiles based on the original image's backgrround

C#

```
RasterImage background = new RasterImage(500, 500, Color.Red);
//load a RasterImage object
RasterImage image = new RasterImage(@"F:\input.png");
//assign a processor to the RasterImage object
ImageProcess processor = new ImageProcess(background);
processor.TextureImage(image);
background.Save(@"F:\output.png");
```

Related API(s)(**ImageProcess.cs**)

public void TextureImage(RasterImage texture);

Description:

Put the texture on all pages of the **RasterImage** object.

Parameters:

Name	Description	Valid Value
texture	the texture on background image	can't be null

public void TextureImage(RasterImage texture, int pageIndex);

Description:

Parameters:

Put the texture on the specified page of the **RasterImage** object.

Name	Description	Valid Value
texture	the texture on background image	can't be null
pageIndex	page index	0 to page count -1

Threshold

If the channel value is less or equal to the given value will be set to 0 or will be set to max. This operation will be applied to every channel.

```
//load a RasterImage object
RasterImage image = new RasterImage(@"F:\input.png");
//assign a processor to the RasterImage object
ImageProcess processor = new ImageProcess(image);
processor.ThresholdImage(120);
image.Save(@"F:\output.png");
```

C#

Related API(s)(**ImageProcess.cs**)

public void ThresholdImage(int threshold);

Description:

Apply to all pages of **RasterImage** object.

Parameters:

Name	Description	Valid Value
threshold	threshold value	0 to 255

public void ThresholdImage(int threshold, int pageIndex);

Description:

Apply to the specified page of **RasterImage** object.

Parameters:

Name	Description	Valid Value
threshold	threshold value	0 to 255
pageIndex	page index	0 to page count -1

Transform color space

Change image's color space

```C#
//load a RasterImage object
RasterImage image = new RasterImage(@"F:\input.png");
//assign a processor to the RasterImage object
ImageProcess processor = new ImageProcess(image);
processor.TransformColorspace(ColorSpace.GRAY);
image.Save(@"F:\output.png");
```

Related API(s)(**ImageProcess.cs**)

public void TransformColorspace(ColorSpace colorSpace);

Description:

Parameters:

Name	Description	Valid Value
colorSpace	target color space	choose from the enum **ColorSpace**

public void TransformColorspace(ColorSpace colorSpace, int pageIndex);

Description:

Parameters:

Name	Description	Valid Value
colorSpace	target color space	choose from the enum **ColorSpace**
pageIndex	page index	0 to page count -1

Transform compression

Change image' s compression type

```C#
//load a RasterImage object
RasterImage image = new RasterImage(@"F:\input.tif");
//assign a processor to the RasterImage object
ImageProcess processor = new ImageProcess(image);
processor.TransformCompression(Compression.Group4);
image.Save(@"F:\output.tif");
```

Related API(s)(**ImageProcess.cs**)

public void TransformCompression(Compression compression);

Description:

Parameters:

Name	Description	Valid Value
compression	target compression	choose from the enum **Compression**

public void TransformCompression(Compression compression, int pageIndex);

Description:

Parameters:

Name	Description	Valid Value
compression	target compression	choose from the enum **Compression**
pageIndex	page index	0 to page count - 1

Transform compress quality

Change the compress quality of JPEG, MIFF or PNG file.

```
C#

//load a RasterImage object
RasterImage image = new RasterImage(@"F:\input.png");
//assign a processor to the RasterImage object
ImageProcess processor = new ImageProcess(image);
processor.TransformCompressQuality(80);
image.Save(@"F:\output.png");
```

Related API(s)(**ImageProcess.cs**)

public void TransformCompressQuality(int quality);

Parameters:

Name	Description	Valid Value
quality	compress quality	1 to 100

Transform GIF version

Change gif file's version. But version 87a not support animation delays nor transparent background colors.

Related API(s)(**ImageProcess.cs**)

public void TransformGIFVersion(GIFVersion version);

Parameters:

Name	Description	Valid Value
version	version of gif file	choose from the enum **GIFVersion**

Transform PNG compress level

Change png file's compress level

Related API(s)(**ImageProcess.cs**)

public void TransformPNGCompressLevel(PNGCompressLevel level);

Parameters:

Name	Description	Valid Value
level	png file's compression level	0 to 9. 0:least but fast compression 9:best but slowest compression

Transform PNG compress strategy

Change png file's compress strategy.

```
//load a RasterImage object
RasterImage image = new RasterImage(@"F:\input.png");
//assign a processor to the RasterImage object
ImageProcess processor = new ImageProcess(image);
processor.TransformPNGCompressStrategy(PNGCompressStrategy.Filtered);
image.Save(@"F:\output.png");
```

C#

Related API(s)(**ImageProcess.cs**)

public void TransformPNGCompressStrategy(PNGCompressStrategy
strategy);

Parameters:

Name	Description	Valid Value
strategy	png file's compress strategy	choose from the enum **PNGCompressStrategy**

Transform PNG filter

Change png file's filter.

```
//load a RasterImage object
RasterImage image = new RasterImage(@"F:\input.png");
//assign a processor to the RasterImage object
ImageProcess processor = new ImageProcess(image);
processor.TransformPNGFilter(PNGFilter.PAETH );
image.Save(@"F:\output.png");
```

C#

Related API(s)(**ImageProcess.cs**)

public void TransformPNGFilter(PNGFilter filter);

Parameters:

Name	Description	Valid Value
filter	filter type	choose from the enum **PNGFilter**

Transform resolution

Change image's resolution

```csharp
//load a RasterImage object
RasterImage image = new RasterImage(@"F:\input.png");
//assign a processor to the RasterImage object
ImageProcess processor = new ImageProcess(image);
processor.TransformResolution(72,72);
image.Save(@"F:\output.png");
```

Related API(s)(**ImageProcess.cs**)

public void TransformResolution(int horiResolution, int vertResolution);

Description:

Change all pages' resolution of the **RasterImage** object

Parameters:

Name	Description	Valid Value
horiResolution	horizonal resolution	>0
vertResolution	vertical resolution	>0

public void TransformResolution(int horiResolution, int vertResolution, int pageIndex);

Description:

Change the specified page's resolution of the **RasterImage** object

Parameters:

Name	Description	Valid Value
horiResolution	horizonal resolution	>0
vertResolution	vertical resolution	>0
pageIndex	page index	0 to page count -1

Transpose

```csharp
//load a RasterImage object
RasterImage image = new RasterImage(@"F:\input.png");
//assign a processor to the RasterImage object
ImageProcess processor = new ImageProcess(image);
processor.TransposeImage();
image.Save(@"F:\output.png");
```

Related API(s)(**ImageProcess.cs**)

public void TransposeImage();

public void TransposeImage(int pageIndex);

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

Transverse

Related API(s)(**ImageProcess.cs**)

public void TransverseImage();

public void TransverseImage(int pageIndex);

Parameters:

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

Trim

Image edge trim

Related API(s)(**ImageProcess.cs**)

public void TrimImage();

public void TrimImage(int pageIndex);

Parameters:

Name	Description	Valid Value
pageIndex	page index	0 to page count -

Vignette

Soften image edge

Related API(s)(**ImageProcess.cs**)

public void VignetteImage(double radius, double sigma, int x, int y);

Description:

Parameters:

Name	Description	Valid Value
radius	the radius of the Gaussian, in pixels.The higher the value, the smoother the image	>0
sigma	standard deviation of the Gaussian, in pixels	default:1
x	major axis of an ellipse	>0
y	minor axis of an ellipse	>0

public void VignetteImage(double radius, double sigma, int x, int y, int pageIndex);

Description:

Parameters:

Name	Description	Valid Value
radius	the radius of the Gaussian, in pixels.The higher the value, the smoother the image	>0
sigma	standard deviation of the Gaussian, in pixels	default:1
x	major axis of an ellipse	>0
y	minor axis of an ellipse	>0
pageIndex	page index	0 to page count - 1

Wave

Change images refer to the sine wave

```csharp
//load a RasterImage object
RasterImage image = new RasterImage(@"F:\input.png");
//assign a processor to the RasterImage object
ImageProcess processor = new ImageProcess(image);
processor.WaveImage(3, 6);
image.Save(@"F:\output.png");
```

Related API(s)(**ImageProcess.cs**)

public void WaveImage(double amplitude, double waveLength);

Description:

Wave all pages of **RasterImage** object.

Parameters:

Name	Description	Valid Value
amplitude	amplitude of the wave	>0
waveLength	wavelength of the wave	>0

public void WaveImage(double amplitude, double waveLength, int pageIndex);

Description:

Wave the specified page of **RasterImage** object.

Parameters:

Name	Description	Valid Value
amplitude	amplitude of the wave	>0
waveLength	wavelength of the wave	>0
pageIndex	page index	0 to page count -1

Draw Annotation

This section will provide enough information for developers to understand how to draw various annotations on **RasterImage** object. There are two ways to finish the drawing. The first is by RasterEdge.Imaging.Annotation.dll, and the other is through the static method of class **DrawMethod.**

Arc

It's really put an 'ellipse' into a rectangle which defined by the two coordinates. You can draw a partial arc with start degree and end degree, but it can be hard to determine the end degree, so you'd better set the end degree to multiplies of ninety degrees.

Draw a partial arc without other additional sets:

Draw a partial arc with optional settings through DrawingInfo

C#

```
//load a RasterImage object
RasterImage image = new RasterImage(@"F:\input.png");
//create a DrawingInfo object
DrawingInfo info = new DrawingInfo();
//set arc outline color
info.OutLineColor = Color.Red;
//set arc outline
info.OutlineWidth = 2;
//set arc fill color
info.FillColor = Color.Blue;
//set startDegree to 0 and endDegree to 360 you will get a closed arc
//well, it's an ellipse.
DrawMethod.DrawArc(image,20,10,80,50,45,270,info);
image.Save(@"F:\output.png");
```

Related API(s)(**DrawMethod.cs**)

public static void DrawArc(RasterImage image, double startX, double startY, double endX, double endY, double startDegree, double endDegree)

Description：

Draw an arc on all pages of **RasterImage** object with no additional settings.

Parameters：

Name	Description	Valid Value
image	input RasterImage object	can't be null
startX	bound rectangle's start X	0 to page's width
startY	bound rectangle's start Y	0 to page's height
endX	bound rectangle's end X	0 to page's width
endY	bound rectangle's end Y	0 to page's height
startDegree	arc's start degree	0 to 360
endDegree	arc's end degree	0 to 360

public static void DrawArc(RasterImage image, double startX, double startY, double endX, double endY, double startDegree, double endDegree, int pageIndex)

Description：

Draw an arc on the specified page of **RasterImage** object with no additional settings.

Parameters：

Name	Description	Valid Value
image	input RasterImage object	can't be null
startX	bound rectangle's start X	0 to page's width
startY	bound rectangle's start Y	0 to page's height
endX	bound rectangle's end X	0 to page's width
endY	bound rectangle's end Y	0 to page's height
startDegree	arc's start degree	0 to 360
endDegree	arc's end degree	0 to 360
pageIndex	pick the specified page as canvas	0 to page count - 1

public static void DrawArc(RasterImage image, double startX, double startY, double endX, double endY, double startDegree, double endDegree, DrawingInfo info)

Description：

Draw an arc on all pages of **RasterImage** object with optional settings.

Parameters：

Name	Description	Valid Value
image	input RasterImage object	can't be null
startX	bound rectangle's start X	0 to page's width
startY	bound rectangle's start Y	0 to page's height
endX	bound rectangle's end X	0 to page's width
endY	bound rectangle's end Y	0 to page's height
startDegree	arc's start degree	0 to 360
endDegree	arc's end degree	0 to 360
info	optional settings for the arc	-

public static void DrawArc(RasterImage image, double startX, double startY, double endX, double endY, double startDegree, double endDegree, DrawingInfo info, int pageIndex)

Description：

Draw an arc on the specified page of **RasterImage** object with optional settings.

Parameters：

Name	Description	Valid Value
image	input RasterImage object	can't be null
startX	bound rectangle's start X	0 to page's width
startY	bound rectangle's start Y	0 to page's height
endX	bound rectangle's end X	0 to page's width
endY	bound rectangle's end Y	0 to page's height
startDegree	arc's start degree	0 to 360
endDegree	arc's end degree	0 to 360
info	optional settings for the arc	-
pageIndex	pick the specified page as canvas	0 to page count - 1

Bezier

Draw a Bezier curve segment. Four points are needed, a start point 'knot', two control points and an end point 'knot'. The first control point defines the direction and the second describe how fast the curve deviates from the attached end 'knot' points.

Draw a Bezier curve with no additional settings:

C#

```
//load a RasterImage object
RasterImage image = new RasterImage(@"F:\input.png");
//points to simulate a bezier path
List<Point> potins = new List<Point>();
potins.Add(new Point(40, 10));
potins.Add(new Point(20, 50));
potins.Add(new Point(90, 10));
potins.Add(new Point(70, 40));
DrawMethod.DrawBezier(image, potins);
image.Save(@"F:\output.png");
```

Draw a Bezier curve with optional settings:

C#

```
//load a RasterImage object
RasterImage image = new RasterImage(@"F:\input.png");
//create a DrawingInfo object
DrawingInfo info = new DrawingInfo();
//set annotation's outline color
info.OutLineColor = Color.Red;
//set annotation's outline
info.OutlineWidth = 2;
//set annotation's fill color
info.FillColor = Color.Blue;
//the points for the bezier
List<Point> potins = new List<Point>();
potins.Add(new Point(40, 10));
potins.Add(new Point(20, 50));
potins.Add(new Point(90, 10));
potins.Add(new Point(70, 40));
DrawMethod.DrawBezier(image, potins, info);
image.Save(@"F:\output.png");
```

Related API(s)(**DrawMethod.cs**)

public static void DrawBezier(RasterImage image, List<Point> pointList)

Description：

Draw a bezier curve on all pages of **RasterImage** object without additional settings.

Parameters：

Name	Description	Valid Value
image	the input image	can't be null
pointList	consist of 4 points to describe a bezier curve path	-

public static void DrawBezier(RasterImage image, List<Point> pointList, int pageIndex)

Description：

Draw a bezier curve on the specified page of **RasterImage** object without additional settings.

Parameters：

Name	Description	Valid Value
image	the input image	can't be null
pointList	consist of 4 points to describe a bezier curve path	-
pageIndex	pick the specified page as canvas	0 to page count - 1

public static void DrawBezier(RasterImage image, List<Point> pointList, DrawingInfo info)

Description：

Draw a bezier curve on all pages of **RasterImage** object with optional settings.

Parameters：

Name	Description	Valid Value
image	the input image	can't be null
pointList	consist of 4 points to describe a bezier curve path	-
info	optional settings for the curve	-

public static void DrawBezier(RasterImage image, List<Point> pointList, DrawingInfo info, int pageIndex)

Description：

Draw a bezier curve on the specified page of **RasterImage** object with optional settings.

Parameters：

Name	Description	Valid Value
image	the input image	can't be null
pointList	consist of 4 points to describe a bezier curve path	-
info	optional settings for the curve	-
pageIndex	pick the specified page as canvas	0 to page count - 1

public static Bitmap DrawBezier(List<Point> pointList)

Description：

Draw a Bezier curve with no additional settings and return it as a bitmap.

Parameters：

Name	Description	Valid Value
pointList	consist of 4 points to describe a bezier curve path	-

public static Bitmap DrawBezier(List<Point> pointList, DrawingInfo info)

Description：

Draw a Bezier curve with optional settings and return it as a bitmap.

Parameters：

Name	Description	Valid Value
pointList	consist of 4 points to describe a bezier curve path	-
info	optional settings for the curve	-

Circle

Draw a circle though any point on it's circumference.

Draw a circle with no additional settings:

```C#
RasterImage image = new RasterImage(@"F:\input.png");
//center of the circle
Point center = new Point(100, 100);
//a point on the circle.
Point anyPointOnCirle = new Point(100, 50);
DrawMethod.DrawCircle(image, center, anyPointOnCirle);
image.Save(@"F:\output.png");
```

Draw a circle with optional settings:

```C#
//load a RasterImage object
RasterImage image = new RasterImage(@"F:\input.png");
//create a DrawingInfo object
DrawingInfo info = new DrawingInfo();
//set annotation's outline color
info.OutLineColor = Color.Red;
//set annotation's outline
info.OutlineWidth = 2;
//set annotation's fill color
info.FillColor = Color.Blue;
//center of the circle
Point center = new Point(100, 100);
//a point on the circle.
Point anyPointOnCirle = new Point(100, 50);
DrawMethod.DrawCircle(image, center, anyPointOnCirle, info);
image.Save(@"F:\output.png");
```

Related API(s)(**DrawMethod.cs**)

public static void DrawCircle(RasterImage image, Point center, Point anyPointOnCircle, DrawingInfo info)

Description：

Draw a circle on all pages of the **RasterImage** object with optional settings.

Parameters：

Name	Description	Valid Value
image	the input image	can't be null
center	circle's center	a valid point
anyPointOnCircle	any point on the circle's circumference	a valid point
info	optional settings for the circle	-

public static void DrawCircle(RasterImage image, Point center, Point anyPointOnCircle, DrawingInfo info, int pageIndex)

Description：

Draw a circle on the specified page of the **RasterImage** object with optional settings.

Parameters：

Name	Description	Valid Value
image	the input image	can't be null
center	circle's center	a valid point
anyPointOnCircle	any point on the circle's circumference	a valid point
info	optional settings for the circle	-
pageIndex	pick the specified page as canvas	0 to page count - 1

public static void DrawCircle(RasterImage image, Point center, Point anyPointOnCircle)

Description：

Draw a circle on all pages of **RasterImage** object with no additional settings.

Parameters：

Name	Description	Valid Value
image	the input image	can't be null
center	circle's center	a valid point
anyPointOnCircle	any point on the circle's circumference	a valid point

public static void DrawCircle(RasterImage image, Point center, Point anyPointOnCircle, int pageIndex)

Description：

Draw a circle on the specified page of the **RasterImage** object with no additional settings.

Parameters：

Name	Description	Valid Value
image	the input image	can't be null
center	circle's center	a valid point
anyPointOnCircle	any point on the circle's circumference	a valid point
pageIndex	pick the specified page as canvas	0 to page count - 1

public static Bitmap DrawCircle(Point center, Point anyPointOnCircle)

Description：

Draw a circle with no additional settings and return it as a bitmap

Parameters：

Name	Description	Valid Value
center	circle's center	a valid point
anyPointOnCircle	any point on the circle's circumference	a valid point

public static Bitmap DrawCircle(Point center, Point anyPointOnCircle, DrawingInfo info)

Description：

Draw a circle with optional settings and return it as a bitmap

Parameters：

Name	Description	Valid Value
center	circle's center	a valid point
anyPointOnCircle	any point on the circle's circumference	a valid point
info	optional settings for the circle	-

Ellipse

Draw an ellipse through the center point, major semi-axis and minor semi-axis. Set start degree to 0 and end degree to 360 to get a complete ellipse, otherwise a partial ellipse will be returned. Draw an ellipse with no additional settings:

```C#
//load a RasterImage object
RasterImage image = new RasterImage(@"F:\input.png");
DrawMethod.DrawEllipse(image, 100, 100, 60, 30, 0, 360);
image.Save(@"F:\output.png");
```

Draw an ellipse with optional settings:

```C#
//load a RasterImage object
RasterImage image = new RasterImage(@"F:\input.png");
//create a DrawingInfo object
DrawingInfo info = new DrawingInfo();
//set annotation's outline color
info.OutLineColor = Color.Red;
//set annotation's outline
info.OutlineWidth = 2;
//set annotation's fill color
info.FillColor = Color.Blue;
DrawMethod.DrawEllipse(image, 100, 100, 60, 30, 0, 360, info);
image.Save(@"F:\output.png");
```

Related API(s)(**DrawMethod.cs**)

public static void DrawEllipse(RasterImage image, int centerPointX, int centerPointY, int

majorSemiAxis, int minorSemiAxis, int arcStat, int arcEnd)

Description：

Draw an ellipse on all pages of **RasterImage** object with no additional settings.

Parameters：

Name	Description	Valid Value
image	the input image	can't be null
centerPointX	x coordinate of ellipse's center	0 to page's width
centerPointY	y coordinate of ellipse's center	0 to page's height
majorSemiAxis	ellipse's major semi-axis	0 to page's width
minorSemiAxis	ellipse's major semi-axis	0 to page's height
arcStat	ellipse's start degree	0 to 360
arcEnd	ellipse's end degree	0 to 360

public static void DrawEllipse(RasterImage image, int centerPointX, int centerPointY, int majorSemiAxis, int minorSemiAxis, int arcStat, int arcEnd, int pageIndex)

Description：

Draw an ellipse on the specified page of **RasterImage** object with no additional settings.

Parameters：

Name	Description	Valid Value
image	the input image	can't be null
centerPointX	x coordinate of ellipse's center	0 to page's width
centerPointY	y coordinate of ellipse's center	0 to page's height
majorSemiAxis	ellipse's major semi-axis	0 to page's width
minorSemiAxis	ellipse's major semi-axis	0 to page's height
arcStat	ellipse's start degree	0 to 360
arcEnd	ellipse's end degree	0 to 360
pageIndex	pick a page as the canvas	0 to page count - 1

public static void DrawEllipse(RasterImage image, int centerPointX, int centerPointY, int majorSemiAxis, int minorSemiAxis, int arcStat, int arcEnd, DrawingInfo info)

Description：

Draw an ellipse on all pages of **RasterImage** object with optional settings.

Parameters：

Name	Description	Valid Value
image	the input image	can't be null
centerPointX	x coordinate of ellipse's center	0 to page's width
centerPointY	y coordinate of ellipse's center	0 to page's height
majorSemiAxis	ellipse's major semi-axis	0 to page's width
minorSemiAxis	ellipse's major semi-axis	0 to page's height
arcStat	ellipse's start degree	0 to 360
arcEnd	ellipse's end degree	0 to 360
info	Optional settings for ellipse	-

public static void DrawEllipse(RasterImage image, int centerPointX, int centerPointY, int majorSemiAxis, int minorSemiAxis, int arcStat, int arcEnd, DrawingInfo info, int pageIndex)

Description：

Draw an ellipse on the specified page of **RasterImage** object with optional settings

Parameters：

Name	Description	Valid Value
image	the input image	can't be null
centerPointX	x coordinate of ellipse's center	0 to page's width
centerPointY	y coordinate of ellipse's center	0 to page's height
majorSemiAxis	ellipse's major semi-axis	0 to page's width
minorSemiAxis	ellipse's minor semi-axis	0 to page's height
arcStat	ellipse's start degree	0 to 360
arcEnd	ellipse's end degree	0 to 360
info	optional settings for ellipse	-
pageIndex	pick a page as the canvas	0 to page count - 1

public static Bitmap DrawEllipse(int majorSemiAxis, int minorSemiAxis)

Description：

Draw an ellipse with no additional settings and return it as a bitmap

Parameters：

Name	Description	Valid Value
majorSemiAxis	ellipse's major semi-axis	0 to page's width
minorSemiAxis	ellipse's minor semi-axis	0 to page's height

public static Bitmap DrawEllipse(int majorSemiAxis, int minorSemiAxis, DrawingInfo info)

Description：

Draw an ellipse with optional settings and return it as a bitmap.

Parameters：

Name	Description	Valid Value
majorSemiAxis	ellipse's major semi-axis	0 to page's width
minorSemiAxis	ellipse's minor semi-axis	0 to page's height
info	optional settings for ellipse	-

FreeHand

Draw free hand with a list of points.
Draw free hand with no additional settings:

```csharp
C#

//load a RasterImage object
RasterImage image = new RasterImage(@"F:\input.png");
//random points to simulate a freehand path
Random random = new Random();
List<Point> points = new List<Point>();
for (int i = 0; i < 100; i=i+5)
{
points.Add(new Point(random.Next(100, 500), random.Next(100,500)));
}
DrawMethod.DrawFreeHand(image, points);
image.Save(@"F:\output.png");
```

Draw free hand with optional settings:

```csharp
C#

//load a RasterImage object
RasterImage image = new RasterImage(@"F:\input.png");
//create a DrawingInfo object
DrawingInfo info = new DrawingInfo();
//set annotation's outline color
info.OutLineColor = Color.Red;
//set annotation's outline
info.OutlineWidth = 2;
//set annotation's fill color
info.FillColor = Color.Blue;
//random points to simulate a freehand path
Random random = new Random();
List<Point> points = new List<Point>();
for (int i = 0; i < 100; i=i+5)
{
points.Add(new Point(random.Next(100, 500), random.Next(100,500)));
}
DrawMethod.DrawFreeHand(image, points, info);
image.Save(@"F:\output.png");
```

Related API(s)(**DrawMethod.cs**)

public static void DrawFreeHand(RasterImage image, List<Point> pointList)

Description：

Draw free hand on all pages of **RasterImage** object with no additional settings.

Parameters：

Name	Description	Valid Value
image	the input image	can't be null
pointList	input points to simulate the free hand path	can't be null

public static void DrawFreeHand(RasterImage image, List<Point> pointList, int pageIndex)

Description：

Draw free hand on the specified page of **RasterImage** object with no additional settings.

Parameters：

Name	Description	Valid Value
image	the input image	can't be null
pointList	input points to simulate the free hand path	can't be null
pageIndex	pick a page as the canvas	0 to page count - 1

public static void DrawFreeHand(RasterImage image, List<Point> pointList, DrawingInfo info)

Description：

Draw free hand on all pages of **RasterImage** object with optional settings

Parameters：

Name	Description	Valid Value
image	the input image	can't be null
pointList	input points to simulate the free hand path	can't be null
info	optional settings for the free hand	-

public static void DrawFreeHand(RasterImage image, List<Point> pointList, DrawingInfo info, int pageIndex)

Description：

Draw free hand on the specified page of **RasterImage** object with optional settings

Parameters：

Name	Description	Valid Value
image	the input image	can't be null
pointList	input points to simulate the free hand path	can't be null
info	optional settings for the free hand	-
pageIndex	pick a page as the canvas	0 to page count - 1

public static Bitmap DrawFreeHand(List<Point> pointList)

Description：

Draw a free hand with no additional settings and return it as a bitmap.

Parameters：

Name	Description	Valid Value
pointList	input points to simulate the free hand path	can't be null

public static Bitmap DrawFreeHand(List<Point> pointList, DrawingInfo info)

Description：

Draw a free hand with optional settings and return it as a bitmap.

Parameters：

Name	Description	Valid Value
pointList	input points to simulate the free hand path	can't be null
info	optional settings for the free hand	-

Image

Put an image annotation on the input file.

```csharp
//load a RasterImage object
RasterImage image = new RasterImage(@"F:\input.png");
//load a image as the annotation
RasterImage annotation = new RasterImage(@"F:\anno.png");
DrawMethod.DrawImage(image, annotation, 100, 100);
image.Save(@"F:\output.png");
```

Related API(s)(**DrawMethod.cs**)

public static void DrawImage(RasterImage image, RasterImage childImage, int xOffset, int yOffset)

Description：

Put an image annotation on all pages of **RasterImage** object with no additional settings.

Parameters：

Name	Description	Valid Value
image	image be treat as canvas	can't be null
childImage	image be treat as annotation	can't be null
xOffset	annotation's x offset	0 to page width
yOffset	annotation's y offset	0 to page height

public static void DrawImage(RasterImage image, RasterImage childImage, int xOffset, int yOffset, int pageIndedx)

Description：

Put an image annotation on the specified page of **RasterImage** object with no additional settings.

Parameters：

Name	Description	Valid Value
image	image be treat as canvas	can't be null
childImage	image be treat as annotation	can't be null
xOffset	annotation's x offset	0 to page width
yOffset	annotation's y offset	0 to page height
pageIndedx	pick a page as the canvas	0 to page count - 1

public static void DrawImage(RasterImage image, RasterImage childImage, int xOffset, int yOffset, DrawingInfo info)

Description：

Put an image annotation on all pages of **RasterImage** object with optional settings.

Parameters：

Name	Description	Valid Value
image	image be treat as canvas	can't be null
childImage	image be treat as annotation	can't be null

xOffset	annotation's x offset	0 to page width
yOffset	annotation's y offset	0 to page height
info	optional settings for the image ammotation	-

public static void DrawImage(RasterImage image, RasterImage childImage, int xOffset, int yOffset, DrawingInfo info, int pageIndedx)

Description：

Put an image annotation on the specified page of **RasterImage** object with optional settings.

Parameters：

Name	Description	Valid Value
image	image be treat as canvas	can't be null
childImage	image be treat as annotation	can't be null
xOffset	annotation's x offset	0 to page width
yOffset	annotation's y offset	0 to page height
info	optional settings for the image ammotation	-
pageIndedx	pick a page as the canvas	0 to page count - 1

Line

Draw a straight line on the image.

Draw a straight line with no additional settings:

```csharp
C#
//load a RasterImage object
RasterImage image = new RasterImage(@"F:\input.png");
DrawMethod.DrawLine(image, 0, 0, 100, 100);
image.Save(@"F:\output.png");
```

Draw a straight line with optional settings:

```csharp
C#
//load a RasterImage object
RasterImage image = new RasterImage(@"F:\input.png");
//create a DrawingInfo object
DrawingInfo info = new DrawingInfo();
//set line color
info.OutLineColor = Color.Red;
//set line width
info.OutlineWidth = 4;
DrawMethod.DrawLine(image, 0, 0, 100, 100,info);
image.Save(@"F:\output.png");
```

Related API(s)(**DrawMethod.cs**)

public static void DrawLine(RasterImage image, int startX, int startY, int endX, int endY, DrawingInfo info)

Description：

Draw a straight line on all pages of **RasterImage** object with optional settings.

Parameters：

Name	Description	Valid Value
image	the input image	can't be null
startX	start x coordinate of the line	0 to page with
startY	start y coordinate of the line	0 to page height
endX	end x coordinate of the line	0 to page with
endY	end y coordinate of the line	0 to page height
info	optional settings for the line	-

public static void DrawLine(RasterImage image, int startX, int startY, int endX, int endY, DrawingInfo info, int pageIndedx)

Description：

Draw a straight line on the specified page of **RasterImage** object with optional settings.

Parameters：

Name	Description	Valid Value
image	the input image	can't be null
startX	start x coordinate of the line	0 to page with
startY	start y coordinate of the line	0 to page height
endX	end x coordinate of the line	0 to page with
endY	end y coordinate of the line	0 to page height
info	optional settings for the line	-
pageIndedx	pick a page as the canvas	0 to page count - 1

public static void DrawLine(RasterImage image, int startX, int startY, int endX, int endY)

Description：

Draw a straight line on all pages of **RasterImage** object with no additional settings:

Parameters：

Name	Description	Valid Value
image	the input image	can't be null
startX	start x coordinate of the line	0 to page with
startY	start y coordinate of the line	0 to page height
endX	end x coordinate of the line	0 to page with
endY	end y coordinate of the line	0 to page height

public static void DrawLine(RasterImage image, int startX, int startY, int endX, int endY, int pageIndex)

Description：

Draw a straight line on the specified page of **RasterImage** object with no additional settings:

Parameters：

Name	Description	Valid Value
image	the input image	can't be null
startX	start x coordinate of the line	0 to page with
startY	start y coordinate of the line	0 to page height
endX	end x coordinate of the line	0 to page with
endY	end y coordinate of the line	0 to page height
pageIndex	pick a page as the canvas	0 to page count - 1

public static Bitmap DrawLine(int startX, int startY, int endX, int endY)

Description：

Draw a straight line with no additional settings and return it as a bitmap.

Parameters：

Name	Description	Valid Value
startX	start x coordinate of the line	0 to page with
startY	start y coordinate of the line	0 to page height
endX	end x coordinate of the line	0 to page with
endY	end y coordinate of the line	0 to page height

public static Bitmap DrawLine(int startX, int startY, int endX, int endY, DrawingInfo info)

Description：

Draw a straight line with optional settings and return it as a bitmap.

Parameters：

Name	Description	Valid Value
startX	start x coordinate of the line	0 to page with
startY	start y coordinate of the line	0 to page height
endX	end x coordinate of the line	0 to page with
endY	end y coordinate of the line	0 to page height
info	optional settings for the straight line	-

Polygon

Draw a polygon with no additional settings:

```csharp
//load a RasterImage object
RasterImage image = new RasterImage(@"F:\input.png");
/random points to simulate a polygon path
Random random = new Random();
List<Point> points = new List<Point>();
for (int i = 0; i < 100; i = i + 5)
{
points.Add(new Point(random.Next(100, 500), random.Next(100,500)));
}
DrawMethod.DrawPolygon(image,points);
image.Save(@"F:\output.png");
```

Draw a polygon with optional settings:

```csharp
//load a RasterImage object
RasterImage image = new RasterImage(@"F:\input.png");
//create a DrawingInfo object
DrawingInfo info = new DrawingInfo();
//set line color
info.OutLineColor = Color.Red;
//set line width
info.OutlineWidth = 2;
//set fill color to blue.//(0,0,0,0)for a transparency color.
info.FillColor = Color.Blue;
//random points to simulate a polygon path
Random random = new Random();
List<Point> points = new List<Point>();
for (int i = 0; i < 100; i = i + 5)
{
points.Add(new Point(random.Next(100, 500), random.Next(100,500)));
}
DrawMethod.DrawPolygon(image,points,info);
image.Save(@"F:\output.png");
```

Related API(s)(**DrawMethod.cs**)

public static void DrawPolygon(RasterImage image, List<Point> poinList)

Description：

Draw a polygon on all pages of **RasterImage** object with no additional settings.

Parameters：

Name	Description	Valid Value
image	the input image	can't be null
poinList	points to simulate a polygon area	can't be null

public static void DrawPolygon(RasterImage image, List<Point> poinList, int pageIndex)

Description：

Draw a polygon area on the specified page of **RasterImage** object with no additional settings.

Parameters：

Name	Description	Valid Value
image	the input image	can't be null
poinList	points to simulate a polygon area	can't be null
pageIndex	pick a page as the canvas	0 to page count - 1

public static void DrawPolygon(RasterImage image, List<Point> poinList, DrawingInfo info)

Description：

Draw a polygon on all pages of RasterImage object with optional settings.

Parameters：

Name	Description	Valid Value
image	the input image	can't be null
poinList	points to simulate a polygon area	can't be null
info	the optional settings for the polygon	-

public static void DrawPolygon(RasterImage image, List<Point> poinList, DrawingInfo info, int pageIndex)

Description：

Draw a polygon on the specified page of RasterImage object with optional settings.

Parameters：

Name	Description	Valid Value
image	the input image	can't be null
poinList	points to simulate a polygon area	can't be null
info	the optional settings for the polygon	-
pageIndex	pick a page as the canvas	0 to page count - 1

public static Bitmap DrawPolygon(List<Point> poinList)

Description：

Draw a polygon with no additional settings and return it as a bitmap.

Parameters：

Name	Description	Valid Value
poinList	points to simulate a polygon area	can't be null

public static Bitmap DrawPolygon(List<Point> poinList, DrawingInfo info)

Description：

Draw a polygon with optional settings and return it as a bitmap.

Parameters：

Name	Description	Valid Value
poinList	points to simulate a polygon area	can't be null
info	the optional settings for the polygon	-

Polygon Line

Draw a polygon line with no additional settings:

```csharp
//load a RasterImage object
RasterImage image = new RasterImage(@"F:\input.png");
//random points to simulate a polygon path
Random random = new Random();
List<Point> points = new List<Point>();
for (int i = 0; i < 100; i = i + 5)
{
points.Add(new Point(random.Next(100, 500), random.Next(100,500)));
}
DrawMethod.DrawPolygonLine(image, points);
image.Save(@"F:\output.png");
```

Draw a polygon line with no optional settings:

```csharp
//load a RasterImage object
RasterImage image = new RasterImage(@"F:\input.png");
//create a DrawingInfo object
DrawingInfo info = new DrawingInfo();
//set line color
info.OutLineColor = Color.Red;
//set line width
info.OutlineWidth = 2;
//set fill color to transparency
info.FillColor = Color.FromArgb(0,0,0,0);
//random points to simulate a polygon path
Random random = new Random();
List<Point> points = new List<Point>();
for (int i = 0; i < 100; i = i + 5)
{
points.Add(new Point(random.Next(100, 500), random.Next(100,500)));
}
DrawMethod.DrawPolygonLine(image, points, info);
image.Save(@"F:\output.png");
```

Related API(s)(**DrawMethod.cs**)

public static void DrawPolygonLine(RasterImage image, List<Point> poinList)

*Description***：**

Draw polygon line on all pages of **RasterImage** with no additionalsettings.

*Parameters***：**

Name	Description	Valid Value
image	the input image	can't be null
poinList	a list of points tosimulate a polygon line path	can't be null

public static void DrawPolygonLine(RasterImage image, List<Point> poinList, int pageIndedx)

*Description***：**

Draw a polygon line on the specified page of **RasterImage** object with no additional settings.

*Parameters***：**

Name	Description	Valid Value
image	the input image	can't be null
poinList	a list of points to simulate a polygon linepath	can't be null
pageIndedx	pick a page as the canvas	0 to page count - 1

public static void DrawPolygonLine(RasterImage image, List<Point> poinList, DrawingInfo info)

*Description***：**

Draw polygon line on all pages of **RasterImage** with optionalsettings.

*Parameters***：**

Name	Description	Valid Value
image	the input image	can't be null
poinList	a list of points tosimulate a polygon line path	can't be null
info	optional settings for the polygonline	-

public static void DrawPolygonLine(RasterImage image, List<Point> poinList, DrawingInfo info, int pageIndex)

*Description***：**

Draw a polygon line on the specified page of the **RasterImage** with optional settings.

*Parameters***：**

Name	Description	Valid Value
image	the input image	can't be null
poinList	a list of points tosimulate a polygon line path	can't be null
info	optional settings for the polygonline	-
pageIndex	pick a page as the canvas	0 to page count - 1

public static Bitmap DrawPolygonLine(List<Point> poinList)

*Description***：**

Draw a polygon line with no additional settings and return it asbitmap.

*Parameters***：**

Name	Description	Valid Value
poinList	a list of points to simulate a polygon line path	can't be null

public static Bitmap DrawPolygonLine(List<Point> poinList, DrawingInfo info)

Description：

Draw a polygon line with optional settings and return it as bitmap.

Parameters：

Name	Description	Valid Value
poinList	a list of points to simulate a polygon line path	can't be null
info	optional settings for the polygon line	-

Rectangle

Draw a rectangle with no additional settings:

```C#
//load a RasterImage object
RasterImage image = new RasterImage(@"F:\input.png");
DrawMethod.DrawRectangle(image, new Rectangle(100, 100, 100, 200));
image.Save(@"F:\output.png");
```

Draw a rectangle with optional settings:

```C#
//load a RasterImage object
RasterImage image = new RasterImage(@"F:\input.png");
//create a DrawingInfo object
DrawingInfo info = new DrawingInfo();
//set rectangle outline color
info.OutLineColor = Color.Red;
//set rectangle outline width
info.OutlineWidth = 2;
//set fill color to transparency
info.FillColor = Color.FromArgb(0,0,0,0);
DrawMethod.DrawRectangle(image, new Rectangle(100,100,100,200), info);
image.Save(@"F:\output.png");
```

Related API(s)(**DrawMethod.cs**)

public static void DrawRectangle(RasterImage image, Rectangle rect)

Description：

Draw rectangle on all pages of **RasterImage** object with no additional settings.

Parameters：

Name	Description	Valid Value
image	the input image	can't be null
rect	the rectangle annotation	can't be null

public static void DrawRectangle(RasterImage image, Rectangle rect, int pageIndex)

Description：

Draw rectangle on the specified page of **RasterImage** object with no additional settings.

Parameters：

Name	Description	Valid Value
image	the input image	can't be null
rect	the rectangle annotation	can't be null
pageIndex	pick a page as the canvas	0 to page count - 1

public static void DrawRectangle(RasterImage image, Rectangle rect, DrawingInfo info)

Description：

Draw a rectangle on all pages of **RasterImage** object with optional settings.

Parameters：

Name	Description	Valid Value
image	the input image	can't be null
rect	the rectangle annotation	can't be null
info	optional settings for the rectangle	-

public static void DrawRectangle(RasterImage image, Rectangle rect, DrawingInfo info, int pageIndex)

Description：

Draw a rectangle on the specified page of **RasterImage** object with optional settings.

Parameters：

Name	Description	Valid Value
image	the input image	can't be null
rect	the rectangle annotation	can't be null
info	optional settings for the rectangle	-
pageIndex	pick a page as the canvas	0 to page count - 1

public static Bitmap DrawRectangle(int width, int height)

Description：

Draw a rectangle with no additional settings and return it as bitmap.

Parameters：

Name	Description	Valid Value
width	rectangle's width	>0
height	rectangle's height	>0

public static Bitmap DrawRectangle(int width, int height, DrawingInfo info)

Description：

Draw a rectangle with optional settings and return it as a bitmap.

Name	Description	Valid Value
width	rectangle's width	>0
height	rectangle's height	>0
info	optional settings for the rectangle	-

Text

Draw text on the input file, if the font name don't set, the 'Calibri' will be used defaultly.

Draw text with no additional settings:

```csharp
C#
//load a RasterImage object
RasterImage image = new RasterImage(@"F:\input.png");
//set the text
String text = "www.RasterEdge.com";
DrawMethod.DrawText(image, text, 100,100);
image.Save(@"F:\output.png");
```

Draw text with optional settings:

```csharp
C#
//load a RasterImage object
RasterImage image = new RasterImage(@"F:\input.png");
//create a DrawingInfo object
DrawingInfo info = new DrawingInfo();
//set font family
info.FontName = "Calibri";
//set font size
info.FontSize = 40;
//set font fill color
info.FillColor = Color.Blue;
//set font outline color
info.OutLineColor = Color.Red;
//set the text
String text = "www.RasterEdge.com";
DrawMethod.DrawText(image, text, 100,100,info);
image.Save(@"F:\output.png");
```

Related API(s)(**DrawMethod.cs**)

public static void DrawText(RasterImage image, String text, int xOffset, int yOffset)

Description：

Draw text on all pages of **RasterImage** object with no additional settings.

Parameters：

Name	Description	Valid Value
image	the input image	can't be null
text	the text annotation	can't be null
xOffset	x coordinate of the annotation	0 to page width
yOffset	y coordinate of the annotation	0 to page height

public static void DrawText(RasterImage image, String text, int xOffset, int yOffset, int pageIndex)

Description：

Draw text on the specified page of **RasterImage** object with no additional settings.

Parameters：

Name	Description	Valid Value
image	the input image	can't be null
text	the text annotation	can't be null
xOffset	x coordinate of the annotation	0 to page width
yOffset	y coordinate of the annotation	0 to page height
pageIndex	pick a page as the canvas	0 to page count - 1

public static void DrawText(RasterImage image, String text, int xOffset, int yOffset, DrawingInfo info)

Description：

Draw text on all pages of **RasterImage** object with no additional settings.

Parameters：

Name	Description	Valid Value
image	the input image	can't be null
text	the text annotation	can't be null
xOffset	x coordinate of the annotation	0 to page width
yOffset	y coordinate of the annotation	0 to page height
info	optional settings for the text	-

public static void DrawText(RasterImage image, String text, int xOffset, int yOffset, DrawingInfo info, int pageIndex)

Description：

Draw text on the specified page of **RasterImage** object with optional settings.

Parameters：

Name	Description	Valid Value
image	the input image	can't be null
text	the text annotation	can't be null
xOffset	x coordinate of the annotation	0 to page width
yOffset	y coordinate of the annotation	0 to page height
info	optional settings for the text	-
pageIndex	pick a page as the canvas	0 to page count - 1

public static Bitmap DrawText(String text, int xOffset, int yOffset, int width, int height)

Description：

Draw a text without additional settings and return it as bitmap.

Parameters：

Name	Description	Valid Value
text	the text annotation	can't be null
xOffset	x coordinate of the annotation	0 to page width
yOffset	y coordinate of the annotation	0 to page height
width	the annotation's width	0 to page width
height	the annotation's height	0 to page height

public static Bitmap DrawText(String text, int xOffset, int yOffset, int width, int height, DrawingInfo info)

Description：

Draw a text with optional settings and return it as bitmap.

Parameters：

Name	Description	Valid Value
text	the text annotation	can't be null
xOffset	x coordinate of the annotation	0 to page width
yOffset	y coordinate of the annotation	0 to page height
width	the annotation's width	0 to page width
height	the annotation's height	0 to page height
info	optional settings for the text	-