RasterEdge.XDoc.TIFF for .NET SDK

Developer’s Guide

RasterEdge.DocImagingSDK 9.8.7

2016-05-19

Getting Started

System Requirements for .NET

Supported Operating System

The following Microsoft Windows operating systems are supported:

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

Development Environments

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

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

.NET Framework versions supported

The following .NET Framework versions are supported:

- .NET Framework 2.0
- .NET Framework 3.0
- .NET Framework 3.5
- .NET Framework 4.0
- .NET Framework 4.5
- .NET Framework 4.5.1
- .NET Framework 4.5.2
- .NET Framework 4.6

Reference RasterEdge.XDoc.TIFF in .NET project

Necessary Libraries

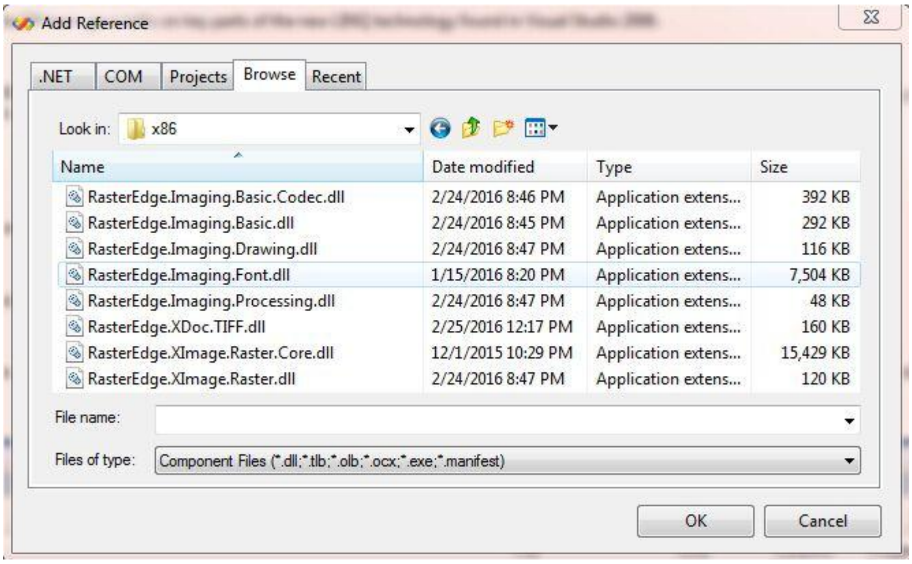
To use RasterEdge.XDoc.TIFF library successfully, the following libraries are necessary:

- RasterEdge.Imaging.Basic.dll
- RasterEdge.Imaging.Basic.Codec.dll
- RasterEdge.Imaging.Drawing.dll
- RasterEdge.Imaging.Processing.dll
- RasterEdge.XImage.Raster.Core.dll
- RasterEdge.XImage.Raster.dll
- RasterEdge.XDoc.TIFF.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**.



Look in: x86

Name	Date modified	Type	Size
RasterEdge.Imaging.Basic.Codec.dll	2/24/2016 8:46 PM	Application extens...	392 KB
RasterEdge.Imaging.Basic.dll	2/24/2016 8:45 PM	Application extens...	292 KB
RasterEdge.Imaging.Drawing.dll	2/24/2016 8:47 PM	Application extens...	116 KB
RasterEdge.Imaging.Font.dll	1/15/2016 8:20 PM	Application extens...	7,504 KB
RasterEdge.Imaging.Processing.dll	2/24/2016 8:47 PM	Application extens...	48 KB
RasterEdge.XDoc.TIFF.dll	2/25/2016 12:17 PM	Application extens...	160 KB
RasterEdge.XImage.Raster.Core.dll	12/1/2015 10:29 PM	Application extens...	15,429 KB
RasterEdge.XImage.Raster.dll	2/24/2016 8:47 PM	Application extens...	120 KB

File name:

Files of type: Component Files (*.dll;*.tlb;*.olb;*.ocx;*.exe;*.manifest)

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

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

FAQ

Errors On Visual Studio

If you get the error as follows:

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

Please check your project configures as following ways:

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

Right-click the project -> Properties ->

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

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

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

Right-click the project -> Properties ->

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

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

Errors On IIS

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

1. Not registered the .net framework to the iis. (One of reasons: install a .net framework before the installation of iis.)

2. The site configured in IIS has no sufficient authority to operate. (Modify permission)

There are some solutions:

1. cd to C:\Windows\Microsoft.NET\Framework64\v2.0.50727,Command to re-register net framework to the iis:aspnet_regiis-i.

2. Right-click the correspond site-> Edit Permissions -> Security -> Group or user names-> Edit -> Add -> Add Everyone 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.

Feature List

TIFF Generator

- Create TIFF from byte array
- Create TIFF from file path
- Create TIFF from stream
- Create TIFF from bitmaps
- Create TIFF form PDF Document
- Create TIFF form Office Document
- Create TIFF form CVS Document

TIFF Document

- Get TIFF Document properties
- Get a particular TIFF page
- Convert To Images
- Append TIFF Document
- Combine TIFF Document
- Split Document
- Add TIFF page/pages
- Delete TIFF page/pages
- Extract TIFF pages
- Insert TIFF page/pages
- Sort TIFF page
- Swap TIFF page
- Rotate pages

TIFF Page

- Get TIFF page properties
- Convert To Image

Annotations

- Add Annotation
- Add Image Annotation

Save

- Save TIFF to file path
- Save TIFF to byte array
- Save TIFF to stream

Programmer Guide

TIFF Object Generator

Our RasterEdge.XDoc.TIFF dll allows developer create TIFF file from TIFF, PDF, Office, Open Office, CSV, RTF, Text, Image and create empty TIFF file.

Extract TIFF document from TIFF file

There are two ways to extract TIFF file from source TIFF file.

First way:

1:Open an existing TIFF file.

2:Call method ExtractPages to extract a new TIFF file.

3:Call method Save/SaveToStream/SaveToBytes to save TIFF file.

The following demo code will show you how to complete the extraction:

C#

```
//open a TIFF file
String inputPath = @"F:\7Pages.tif";
String outputPath = @"F:\output.tif";
TIFFDocument tifDoc = new TIFFDocument(inputPath);
//define the page indexes to extract
int[] extractPageIndex = new int[] { 2, 4, 6 };
//extract page2, page4, page6,and use them to create a new TIFF file.
tifDoc.ExtractPages(extractPageIndex, outputPath);
```

Second way:

1: Call the static method GetOnePageDocument to extract a new TIFF file directly.

C#

```
//open a TIFF file
String inputFilePath = @"F:\input.tif";
String outputFilePath = @"F:\output.tif";
//get the first page from input,tif file, and use it to create a new TIFF file
TIFFDocument.GetOnePageDocument(inputFilePath, 0, outputFilePath);
```

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

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

***Description*:**

Extract a new TIFF file form the source TIFF file,and save it to the given file path.

***Parameters*:**

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

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

Description:

Extract a new TIFF file form the source TIFF file,and save it to the given stream

Parameters:

Name	Description	Valid Value
extractIds	page index to extract	0 to page count - 1
stream	output stream	a valid stream

```
public static void GetOnePageDocument(string sourceFilePath, int index,
string destnFilePath);
```

Description:

Extract a specifed page from source TIFF file and use it to create a new TIFF file.

Parameters:

Name	Description	Valid Value
sourceFilePath	input TIFF file path	must exist
index	the page to extract	0 to page count – 1
destnFilePath	output file path	a valid file path

Create Tiff from Images

When you have an array of images, you just need one step to create a tiff file with these images. The following demo code will show you how to complete the creation:

C#

```
//get two imges.
Bitmap bmp1 = new Bitmap(@"F:\1.png");
Bitmap bmp2 = new Bitmap(@"F:\2.png");
Bitmap[] bmps = new Bitmap[] { bmp1, bmp2 };
//create the new TIFF file.
TIFFDocument tifDoc = new TIFFDocument(bmps);
```

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

```
public TIFFDocument(Bitmap[] images);
```

Description:

Create TIFF file from images.

Parameters:

Name	Description	Valid Value
images	Images for creating TIFF file	Can’t be null

Create TIFF file from Office

http://www.rasteredge.com/how-to/csharp-imaging/word-convert-tiff

Create TIFF file from Open Office

//a link

Create TIFF from PDF

http://www.rasteredge.com/how-to/csharp-imaging/pdf-convert-tiff

Create TIFF from CSV

http://www.rasteredge.com/how-to/csharp-imaging/excel-convert-tiff

Create TIFF from RTF

http://www.rasteredge.com/how-to/csharp-imaging/word-convert-tiff

Create TIFF from Text

//a link

Create TIFF Document object from file path/data

You can easily create a TIFF document object from file path ,just follow the demo code below:

C#

```
//load TIFF file Document object from file path
String inputPath = @"F:\input.tif";
TIFFDocument tifDoc = new TIFFDocument(inputPath);
```

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

```
public TIFFDocument(string fileName);
```

***Description*:**

Create TIFF Document object from file path.

***Parameters*:**

Name	**Description**	**Valid Value**
fileName	the file path stands for a TIFF file	there must be a valid tiff file at the location

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

***Description*:**

Create TIFF file from byte array, but the byte array must be a valid TIFF file.

***Parameters*:**

Name	**Description**	**Valid Value**
fileData	a byte array contains a whole valid tiff data	must be valid

```
public TIFFDocument(Stream stream);
```

Description:

Create a TIFF file from stream, but the stream must be a valid TIFF file.

Parameters:

Name	**Description**	**Valid Value**
stream	a stream contains a whole valid tiff data	must be valid

```
public TIFFDocument(Bitmap[] images, ImageCompress imageCompression);
```

***Description*:**

Create TIFF file from images with specified compression.

***Parameters*:**

Name	**Description**	**Valid Value**
images	images for creating TIFF file	can’ t be null
imageCompression	set compression for output TIFF file	Uncompressed, LZW, CCITT1D, Group3Fax, Group4Fax, JPEG, PackBits

```
public TIFFDocument(Bitmap[] images, ImageOutputOption options);
```

***Description*:**

Create TIFF file from images with option setting.

***Parameters*:**

Name	**Description**	**Valid Value**
images	images for creating TIFF file	can’ t be null
options	a series of settings for output TIFF file	-

Working with Document

Get Page Count

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

C#

```
//open a document.
String inputFilePath = @"C:\input.tif";
TIFFDocument tifDoc = new TIFFDocument(inputFilePath);
//get the total page number of the tiff file.
int pageNumber = tifDoc.GetPageCount();
```

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

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

Description:

Get the total page number of the TIFF file.

Return:

Total page number, 0 if failed.

Get Document Type

You can do as follow to get the document type of input file:

C#

```
//open a document.
String inputFilePath = @"C:\input.tif";
TIFFDocument tifDoc = new TIFFDocument(inputFilePath);
//get document type of the input file
DocumentType type = tifDoc.GetDocumentType();
```

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

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

Description:

Get the document type of the input file.

Return:

TIFF, Invalid or other format if failed

Insert/Add empty page(s) into a TIFF file

To insert page(s) into TIFF file, you can work as follows:

1. Open two existing TIFF files through TIFFDocument object.

2. Extract the particular page from file A.

3. Call the AddPage, AddPages, InsertPage or InsertPages to complete the inserting page(s) to file B.

4. Call the TIFFDocument object's Save/SaveToStream/SaveToBytes method and save file B object to file/stream.

The following demo codes will show you how to do:

C#

```
//open two tiff files
String inputFilePath_A = @"F:\input_A.tif";
String inputFilePath_B = @"F:\input_B.tif";
TIFFDocument tifDoc_A = new TIFFDocument(inputFilePath_A);
TIFFDocument tifDoc_B = new TIFFDocument(inputFilePath_B);
//get one page from file A
BasePage page = tifDoc_A.GetPage(0);
//insert the page into file B
//the third page will be the insert page in output.tif
tifDoc_B.InsertPage(page, 2);
tifDoc_B.Save(@"F:\output.tif");
```

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

```
public override void InsertPage(BasePage basePage, int pageIdx);
```

Description:

Insert a tiff page to TIFF file at specified position.

Parameters:

Name	Description	Valid Value
basePage	a tiff page object	can't be null
pageIdx	position of the inserted page.	0 to page count

```
public override void InsertPages(BasePage[] Bapages, int pageOffset);
```

Description:

Insert tiff pages to TIFF file at specified position.

Parameters:

Name	Description	Valid Value
Bapages	tiff page object array	length can't be 0,page item can't be null
pageOffset	position of the inserted page	0 to page count

```
public override void AddPage(BasePage newPage);
```

Description:

Add a tiff page to TIFF file, the new page will be the last page of the file

Parameters:

Name	**Description**	**Valid Value**
newPage	A tiff page object	can’t be null

```
public override void AddPages(BasePage[] pages);
```

Description:

Add tiff pages to TIFF file, the new pages will follow the last page of the file.

Parameters:

Name	**Description**	**Valid Value**
pages	tiff page object array	length can’t be 0 ,page item can’t be null

Delete TIFF Page(s)

To delete/remove TIFF page(s) form TIFF file, the following steps will be work:

1: Open an existing TIFF file.

2: Call the method DeletePage to delete specified page.

3: Call the method Save/SaveToBytes/SaveToStream to save the file to disk/stream.

The following demo code will show you how to delete TIFF page(s):

C#

```
//open a tiff file
String inputFilePath = @"F:\input.tif";
TIFFDocument tifDoc = new TIFFDocument(inputFilePath);
//delete the first page
tifDoc.DeletePage(0);
//save the file
tifDoc.Save(@"F:\output.tif");
```

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

```
public override void DeletePage(int pageIdx);
```

Description:

Delete the specified page from input TIFF file.

Parameters:

Name	Description	Valid Value
pageIdx	the page index of the delete page	0 to page count - 1

```
public void DeletePages(int[] deleteIds);
```

Description:

Delete TIFF pages from input TIFF file

Parameters:

Name	Description	Valid Value
deleteIds	the page index of delete pages	valid value of every item in the array: 0 to page number - 1

```
public override void DeletePages(int fromPageId, int pageCount);
```

Description:

Delete specified TIFF pages form input TIFF file.

Parameters:

Name	Description	Valid Value
fromPageId	the page index of Page deletion starting	0 to page count - 1
pageCount	how much pages to delete	1 to page count - 1

Swap Two Pages

The following demo code will show you how to swap two pages's place in TIFF file:

C#

```
//open a tiff file
String inputFilePath = @"F:\input.tif";
TIFFDocument tifDoc = new TIFFDocument(inputFilePath);
//swap two pages' place
tifDoc.SwapTwoPages(0, 1);
//save the file
tifDoc.Save(@"F:\output.tif");
```

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

```
public override void SwapTwoPages(int pageIdxF, int pageIdxB);
```

Description:

Swap two specified pages' place

Parameters:

Name	**Description**	**Valid Value**
pageIdxF	the first page's index	0 to page count - 1
pageIdxB	the second page's index	0 to page count - 1

Sort/Reorder TIFF Pages

To sort TIFF file pages, you can do as follows:

1: Open an existing TIFF file.

2: Set the sort page index.

3: Call the SortPage method to complete the sorting pages.

4: Call the method Save/SaveToBytes/SaveToStream to save the file to disk/stream.

C#

```
//open a tiff file
String inputFilePath = @"F:\7Pages.tif";
TIFFDocument tifDoc = new TIFFDocument(inputFilePath);
//show page count of the document
int pagecount = tifDoc.GetPageCount();
//define the new order:
//1:the length of the array MUST BE equal to the page count
//2:each page index SHOULD in the array and only once, otherwise,
//the method will throw an exception.
int[] pageOrders = new int[] { 4, 1, 3, 2, 6, 5, 0 };
tifDoc.SortPage(pageOrders);
//save the file
tifDoc.Save(@"F:\output.tif");
```

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

```
public override void SortPage(int[] orderPageIdxs);
```

Description:

Sort the TIFF file pages with specified order;

Parameters:

Name	Description	Valid Value
orderPageIdxs	new order of TIFF pages	0 to page count - 1

Extract TIFF Page(s) to TIFF file/stream

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

1. Open an existing TIFF file.
2. Define the page indexes will be extracted.
3. Call the method ExtractPages to extract pages and output to file or stream.

C#

```
//open a tiff file
String inputFilePath = @"F:\7Pages.tif";
//specify the output path of the extract TIFF file
String outputPath = @"F:\output.tif";
TIFFDocument tifDoc = new TIFFDocument(inputFilePath);
//show the page count, the page index to extract must below page count
int pagecount = tifDoc.GetPageCount();
//select pages to extract
int[] pageIndexes = new int[] { 0, 1, 2 };
tifDoc.ExtractPages(pageIndexes, outputPath);
```

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

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

Description:

Extract specified pages form TIFF file and save them into another TIFF file.

Parameters:

Name	Description	Valid Value
extractIds	the page indexes that will be extracted	0 to page count - 1
filePath	output file path	Valid file path

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

Description:

Extract specified pages form TIFF file and save them into stream.

Parameters:

Name	Description	Valid Value
extractIds	the page indexes that will be extracted	0 to page count - 1
stream	output stream	Valid file stream or memory stream

Get a Particular Page

To get a particular page form TIFF file, the following demo code will be necessary:

C#

```csharp
//open a tiff file
String inputFilePath = @"F:\input.tif";
TIFFDocument tifDoc = new TIFFDocument(inputFilePath);
// get the first page, actually the second page
TIFFPage page = (TIFFPage)tifDoc.GetPage(1);
```

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

```csharp
public override BasePage GetPage(int pageIdx);
```

Description:

Get specified page form TIFF Document.

Parameters:

Name	Description	Valid Value
pageIdx	The page index	0 to page count - 1

Rotate TIFF Page(s)

Rotate page 90 degree or Flip the page.

C#

```csharp
//open a tiff file
String inputFilePath = @"F:\input.tif";
TIFFDocument tifDoc = new TIFFDocument(inputFilePath);
// get the first page, actually the second page
TIFFPage page = (TIFFPage)tifDoc.GetPage(0);
//rotate 90 degree.
page.Rotate(RotateOder.Clockwise90);
```

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

```csharp
public void Rotate(RotateOder order);
```

Description:

Rotate page 90 degree or flip the page.

Parameters:

Name	Description	Valid Value
order	CounterClockwise/Clockwise/Flip the page	This is an enum

Combine/Append TIFF Files

The demo code to combine multiple TIFF files into one:

C#

```
String inputFilePath_1 = @"F:\1.tif";
String inputFilePath_2 = @"F:\2.tif";
String inputFilePath_3 = @"F:\3.tif";
String[] inputFilePaths = new String[3] { inputFilePath_1, inputFilePath_2,
inputFilePath_3 };
String outputPath = @"F:\output.tif";
//Combine 3 tiff files into one
TIFFDocument.CombineDocument(inputFilePaths, outputPath);
```

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

```
public static void CombineDocument(Stream[] sourceFiles, string outFile);
```

Description:

Combine multiple TIFF files into one, and save it to the specified file path.

Parameters:

Name	Description	Valid Value
sourceFiles	these files will be combined into one TIFF file	valid TIFF file stream
outFile	output file path	valid file path

```
public static void CombineDocument(string[] sourceFiles, Stream stream);
```

Description:

Combine multiple TIFF files into one, and save it to the specified stream

Parameters:

Name	Description	Valid Value
sourceFiles	these files will be combined into one TIFF file	valid TIFF file path
stream	output file stream	valid stream

```
public static void CombineDocument(Stream[] sourceFiles, Stream stream);
```

Description:

Combine multiple TIFF files into one, and save it to the specified stream

Parameters:

Name	Description	Valid Value
sourceFiles	these files will be combined into one TIFF file	valid TIFF file stream
stream	output tiff stream	valid stream

```
public static void CombineDocument(string[] sourceFiles, string destn);
```

Description:

Combine multiple TIFF files into one, and save it to the specified file path

Parameters:

Name	**Description**	**Valid Value**
sourceFiles	these files will be combined into one TIFF file	valid TIFF file path
destn	output file path	valid file path

```
public override int AppendDocument(BaseDocument appendDoc);
```

Description:

Append new TIFF file to the original one.

Parameters:

Name	**Description**	**Valid Value**
appendDoc	the new TIFF file	can't be null

Split by page index

The following demo code will spilt the TIFF file by page index:

C#

```
String inputFilePath = @"F:\7Pages.tif";
String outputPath_1 = @"F:\output_1.tif";
String outputPath_2 = @"F:\output_2.tif";
String[] outputFilePaths = new String[2] { outputPath_1, outputPath_2 };
//Split Tiff file into two files.
TIFFDocument.SplitDocument(inputFilePath, 3, outputFilePaths);
```

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

```
public static void SplitDocument(string source, int index, string[]
fileName);
```

Description:

Split Tiff file into two and save them to the specified file path

Parameters:

Name	Description	Valid Value
source	input TIFF file path	valid input file path
index	0 to index pages will be the first output file, the rest of pages will be the second output file.	0 to page count - 1
fileName	the path to save.	valid output file path

```
public static void SplitDocument(string source, int index, Stream[]
streams);
```

Description:

Split Tiff file into two and save them to stream.

Parameters:

Name	Description	Valid Value
source	input TIFF file path	valid input file path
index	0 to index pages will be the first output file, the rest of pages will be the second output file.	0 to page count - 1
streams	the stream to save output files	valid stream

```
public static void SplitDocument(Stream inputFilePath, int[]
splitIndexes, Stream[] outputStreams);
```

Description:

Split Tiff file into files with specified page indexes and save them to streams.

Parameters:

Name	Description	Valid Value
inputFilePath	Input tiff file stream	valid input tiff stream
splitIndexes	The page indexes to be use to split file.	0 to page count - 1
streams	the stream to save output files	valid stream

```
public static void SplitDocument(Stream inputStream, int[] splitIndexes,
string[] outputFilePaths);
```

Description:

Split Tiff file into files with specified page indexes and save them to files on the disk.

Parameters:

Name	**Description**	**Valid Value**
inputFilePath	Input tiff file stream	valid input tiff stream
splitIndexes	The page indexes to be use to split file.	0 to page count - 1
outputFilePaths	File paths to save output tiff files.	valid file path

```
public static void SplitDocument(string inputFilePath, int[]
splitIndexes, Stream[] outputStreams);
```

Description:

Split Tiff file into files with specified page indexes and save them to streams.

Parameters:

Name	**Description**	**Valid Value**
inputFilePath	Input tiff file stream	valid input tiff stream
splitIndexes	The page indexes to be use to split file.	0 to page count - 1
outputStreams	the stream to save output files	valid stream

```
public static void SplitDocument(string inputFilePath, int[]
splitIndexes, string[] outputFilePaths);
```

Description:

Split Tiff file into files with specified page indexes and save them to files on the disk.

Parameters:

Name	**Description**	**Valid Value**
inputFilePath	Input tiff file stream	valid input tiff stream
splitIndexes	The page indexes to be use to split file.	0 to page count - 1
outputFilePaths	File paths to save output tiff files.	valid file path

Convert to Images

To convert TIFF Document to images, just follow the steps below:

1: Open an existing TIFF file.

2: Call the method ConvertToImages to complete the conversion.

The following demo code will show you how to convert TIFF document to images.

C#

```
//open a TIFF file
String inputPath = @"F:\7Pages.tif";
//set the output directory;
String outputDir = @"F:\Tiff2Png\";
TIFFDocument tifDoc = new TIFFDocument(inputPath);
//convert all pages to png images
//it will create png images in the directory whose name is "demo_"
//note:you must create the output folder on your disk.
tifDoc.ConvertToImages(ImageType.PNG, outputDir, "demo_");
```

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

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

Description:

Convert all TIFF file pages to image with specified format and save them in the given folder.

Parameters:

Name	Description	Valid Value
target	format of output image	choose from the enum
directory	output directory	must exist
fileName	output images' name without suffix.	a string

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

Description:

Convert all TIFF file pages to image with specified format and save them in the stream

Parameters:

Name	Description	Valid Value
targetType	format of output image	choose from the enum
streams	output stream	valid stream

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

Description:

Convert all TIFF file pages to target format images with specified zoom value and save them to streams,

Parameters:

Name	Description	Valid Value
targetType	format of output image	choose from the enum
zoomValue	the magnification of the output image.	if zoomValue*pageWidth>5500 or zoomValue*pageHeight>5500,the original bitmpa will be returned.
streams	output stream	valid stream

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

Convert all TIFF file pages to target format images with specified resolution and save them to streams.

```
Parameters:
```

Name	Description	Valid Value
targetType	format of output image	choose from the enum
resolution	resolution of output image	>0
streams	output stream	valid stream

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

Description

Convert all TIFF file pages to target format images with specified zoom value and save them on the given folder.

Parameters:

Name	Description	Valid Value
targetType	format of output image	choose from the enum
zoomValue	the magnification of the output image.	if zoomValue*pageWidth>5500 or zoomValue*pageHeight>5500,the original bitmpa will be returned.
directory	output directory	must exist
fileName	output images' name without suffix.	a string

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

Description

Convert all TIFF file pages to target format images with specified resolution and save them on the given folder.

Parameters:

Name	Description	Valid Value
targetType	format of output image	choose from the enum
resolution	Output image's resolution	>0
directory	output directory	must exist
fileName	output images' name without suffix.	a string

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

Description:

Convert all TIFF file pages to target format image with option settings and save them to streams.

Parameters:

Name	**Description**	**Valid Value**
targetType	format of output image	choose from the enum
option	settings for output image	-
streams	output stream	valid stream

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

Description:

Convert all TIFF pages to target format images with option settings and save them to the given folder.

Parameters:

Name	**Description**	**Valid Value**
targetType	format of output image	choose from the enum
option	settings for output image	-
directory	output directory	must exist
fileName	output images’ name without suffix.	a string

TIFF Convert to PDF Document

Add extra references:

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

- RasterEdge.XDoc.PDF.dll

To achieve the conversion, please do as follows:

1: Open an existing TIFF file.

2: Call the method ConvertToDocument to complete the conversion.

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

C#

```
//open a TIFF file
String inputPath = @"F:\input.tif";
String outputPath = @"F:\output.pdf";
TIFFDocument tifDoc = new TIFFDocument(inputPath);
//convert TIFF file to PDF file.
tifDoc.ConvertToDocument(DocumentType.PDF, outputPath);
```

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

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

***Description*:**

Convert TIFF file to PDF file, and save it to the given file path.

***Parameters*:**

Name	**Description**	**Valid Value**
targetType	format of output file	DocumentType.PDF
filePath	output file path	a valid file path

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

***Description*:**

Convert TIFF file to PDF file, and save it to the stream.

***Parameters*:**

Name	**Description**	**Valid Value**
targetType	format of output file	DocumentType.PDF
stream	output stream	a valid stream

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

***Description*:**

Convert TIFF file to PDF file with specified zoom value, and save it to the given file path.

***Parameters*:**

Name	**Description**	**Valid Value**
targetType	format of output file	DocumentType.PDF

zoomValue	the magnification of the output image.	if zoomValue*pageWidth>5500 or zoomValue*pageHeight>5500,you will get the original bitmap's size
filePath	output file path	a valid file path.

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

***Description*:**

Convert TIFF file to PDF file with specified zoom value, and save it to the stream.

***Parameters*:**

Name	**Description**	**Valid Value**
targetType	format of output file	DocumentType.PDF
zoomValue	the magnification of the output image.	if zoomValue*pageWidth>5500 or zoomValue*pageHeight>5500,you will get the original bitmap's size
desStream	output stream	a valid stream

```
public override void ConvertToDocument(DocumentType targetType,
ImageCompress compression, string filePath);
```

***Description*:**

Convert TIFF file to PDF file with specified compression and save it to the given file path.

***Parameters*:**

Name	**Description**	**Valid Value**
targetType	format of output file	DocumentType.PDF
compression	set output file's compression	choose from the enum
filePath	output file path	a valid file path

```
public override void ConvertToDocument(DocumentType targetType,
ImageCompress compression, Stream desStream);
```

***Description*:**

Convert TIFF file to PDF file with specified compression and save it to the stream

***Parameters*:**

Name	**Description**	**Valid Value**
targetType	format of output file	DocumentType.PDF
compression	set output file's compression	choose from the enum
desStream	output stream	a valid stream

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

***Description*:**

Convert TIFF file to PDF file with specified resolution and save it to the given file path.

***Parameters*:**

Name	**Description**	**Valid Value**
targetType	format of output file	DocumentType.PDF

resolution	resolution of output file	>0
filePath	output file path	a valid file path

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

Description:

Convert TIFF file to PDF file with specified resolution and save it to the stream.

***Parameters*:**

Name	Description	Valid Value
targetType	format of output file	DocumentType.PDF
resolution	resolution of output file	>0
desStream	output stream	A valid stream

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

***Description*:**

Convert TIFF file to PDF file with option settings and save it to the given file path.

***Parameters*:**

Name	Description	Valid Value
targetType	format of output file	DocumentType.PDF
filePath	output file path	a valid file path
options	Settings for output file	-

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

***Description*:**

Convert TIFF file to PDF file with option settings and save it to the stream.

***Parameters*:**

Name	Description	Valid Value
targetType	format of output file	DocumentType.PDF
desStream	output stream	a valid stream
options	Settings for output file	-

Working with TIFF Page

Get Page width/height

C#

```
//open a TIFF file
TIFFDocument tifDoc = new TIFFDocument(@"F:\input.tif");
TIFFPage page = (TIFFPage)tifDoc.GetPage(0);
//Get physical width of the page in inch
float width = page.GetWidth();
float height = page.GetHeight();
//Get width in pixel
int widthInPixel = page.GetWidthInPixel();
int heightInPixel = page.GetHeightInPixel();
```

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

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

***Description*:**

Get TIFF page's physical width in inch

***Return*:**

0 if failed.

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

***Description*:**

Get TIFF page's physical height in inch

***Return*:**

0 if failed.

```
public int GetWidthInPixel();
```

***Description*:**

Get TIFF page's width in pixel

***Return*:**

0 if failed.

```
public int GetHeightInPixel();
```

***Description*:**

Get TIFF page's height in pixel

***Return*:**

0 if failed.

Convert to Image

To convert TIFF page to image, you just need several steps as follows:

1: Open an existing TIFF file through TIFF document object.

2: Call Method GetPage to get a TIFF page object

3: Call method ConvertToImage to convert TIFF page to image and save it to file path, byte array or stream.

The following demo code will show the conversion in details:

C#

```
//open a TIFF file
String inputFilePath = @"F:\7Pages.tif";
String outputFilePath = @"F:\output.png";
TIFFDocument tifDoc = new TIFFDocument(inputFilePath);
//get the first page of the file.
int pageIndex = 0;
TIFFPage page = (TIFFPage)tifDoc.GetPage(pageIndex);
//convert TIFF page to image
Bitmap bmp = page.ConvertToImage();
//save the bitmap to file path.
bmp.Save(outputFilePath);
```

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

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

Description:

Convert TIFF page to bitmap with default source page size.

Return:

A bitmap object, null if failed.

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

Description:

Convert TIFF page to bitmap with specified zoom value.

Parameters:

Name	Description	Valid Value
zoomValue	the magnification of the output image.	if zoomValue*pageWidth>5500 or zoomValue*pageHeight>5500,the original bitmpa will be returned.

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

Description:

Convert TIFF page to bitmap with specified resolution.

Parameters:

Name	Description	Valid Value
targetResolution	resolution of the output image	integer, larger than 0

Return:

A bitmap object, null if failed.

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

Description:

Convert TIFF page to bitmap with specified image size.

Parameters:

Name	Description	Valid Value
targetSize	size of the output image	can’t be null, width >0 && height >0

Return:

A bitmap object, null if failed.

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

Description:

Convert the TIFF page to bitmap with specified height, and the width will be scaled with the same proportion.

Parameters:

Name	Description	Valid Value
height	height of the output image	>0

Return:

A bitmap object, null if failed.

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

Description:

Convert the TIFF page to bitmap with specified width, and the height will be scaled with the same proportion.

Parameters:

Name	Description	Valid Value
width	width of the output image	>0

```
public override void ConvertToImage(ImageType toType, string filePath);
```

Description:

Convert TIFF page to bitmap with specified format and save it to specified file path.

Parameters:

Name	Description	Valid Value
toType	format of output	The value listed in the ImageType.cs
filePath	output file path	valid file path

```
public override void ConvertToImage(ImageType toType, float zoomValue,
string filePath);
```

Description:

Convert TIFF page to bitmap with specified format and zoom value.

Parameters:

Name	Description	Valid Value

toType	format of output	vhoose from the enum
zoomValue	the magnification of the output image.	if zoomValue*pageWidth>5500 or zoomValue*pageHeight>5500,the original bitmpa will be returned.
filePath	output file path	valid file path

```
public override void ConvertToImage(ImageType toType, int resolution,
string filePath);
```

Description:

Convert TIFF page to bitmap with specified format and resolution.

Parameters:

Name	Description	Valid Value
toType	format of output	choose from the enum
resolution	resolution of output bitmap	>0
filePath	output file path	valid file path

```
public override byte[] ConvertToImageBytes(ImageType toType);
```

Description:

Convert TIFF page to bitmap with specified format

Parameters:

Name	Description	Valid Value
toType	format of the output file	choose from the enum

Return:

A byte array, null or empty byte array if failed.

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

Description:

Convert TIFF page to bitmap with specified format and zoom value.

Parameters:

Name	Description	Valid Value
toType	format of output file	choose from the enum
zoomValue	the magnification of the output image.	if zoomValue*pageWidth>5500 or zoomValue*pageHeight>5500,the original bitmpa will be returned.

Return:

A byte array, null or empty byte array if failed.

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

Description:

Convert TIFF page to bitmap with specified format and resolution.

Parameters:

Name	Description	Valid Value

toType	format of output file	choose from the enum
targetResolution	resolution of output bitmap	>0

Return:

A byte array, null or empty byte array if failed.

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

Description:

Convert TIFF page to bitmap with specified format and save it to stream.

Parameters:

Name	Description	Valid Value
toType	format of output file	choose from the enum
stream	stream to save output bitmap	can't be null

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

Description:

Convert TIFF page to bitmap with specified format and zoom value

Parameters:

Name	Description	Valid Value
toType	format of output file	choose from the enum
zoomValue	the magnification of the output image.	if zoomValue*pageWidth>5500 or zoomValue*pageHeight>5500,the original bitmpa will be returned.
stream	stream to save output bitmap	can't be null

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

Description:

Convert TIFF page to bitmap with specified format and resolution

Parameters:

Name	Description	Valid Value
toType	format of output file	choose from the enum
resolution	resolution of output bitmap	>0
stream	stream to save output bitmap	can't be null

```
public override void ConvertToImage(ImageType targetType,
ImageOutputOption option, string filePath);
```

***Description*:**

Convet TIFF page to target format image with option settings and save it to the given file path.

***Parameters*:**

Name	Description	Valid Value
toType	format of output file	choose from the enum
option	settings for output image	-
filePath	output file path	a valid string

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

***Description*：**

Convert all TIFF pages to target format images with option sittings and save them to streams.

***Parameters*：**

Name	Description	Valid Value
toType	format of output file	choose from the enum
option	settings for output image	-
stream	output stream	a valid stream.

Working with Annotation

Add Annotations

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

- RasterEdge.Imaging.Annotation.dll

Three steps to add an annotation on the page:

1: Add RasterEdge.Imaging.Annotation dll to your reference.

2: Create a annotation through AnnotationGenerator

3: Add the annotation on the page.

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

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

Table 1-1

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

C#

```
//open a TIFF file
TIFFDocument tifDoc = new TIFFDocument(@"F:\input.tif");
TIFFPage page = (TIFFPage)tifDoc.GetPage(0);
//create a text annotation
TextAnnotation annotation = AnnotationGenerator.CreateTextAnnotation(10,
100F, 200F, 100F, "www.RasterEdge.com", new System.Drawing.Font("Arial",
12F));
//add it on the page
page.AddAnnotation(annotation);
tifDoc.Save(@"F:\output.tif");
```

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

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

***Description*:**

Add annotation on the page.

***Parameters*:**

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

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

***Description*：**

Add annotation on the page.

***Parameters*：**

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

Bitonal(1 bits per pixel) TIFF File Note:

To keep the original tiff file’s color depth, we will convert all the annotations to black and white format.

Fill color:

The rectangle on your tiff file will be only an outline without setting the fill color or setting the fill color with a very light color.

```
C#
private static AnnotationHandler CreateRectangleAnnotattion()
{
        RectangleAnnotation rect =
        AnnotationGenerator.CreateRectangleAnnotation(new
        RectangleF(0, 0, 100, 100));
        return rect;
}
```

Set the fill color with a dark color and there will be a pure black block on your tiff file.

```
C#
private static AnnotationHandler CreateRectangleAnnotattion()
{
        RectangleAnnotation rect =
        AnnotationGenerator.CreateRectangleAnnotation(new
        RectangleF(0, 0, 100, 100));
        //Create a fill rectangle
        rect.Fill = new
        RasterEdge.Imaging.Annotation.Basic.AnnotationBrush();
        rect.Fill.FillType = FillType.Solid;
        //Set a dark color and you'll get a pure black block annotation
        rect.Fill.Solid_Color = new REColor(Color.Black);
        return rect;
}
```

Set the fill color only to ARGB(255,182,182,182) can get you a pure white block on the tiff file.

C#

```
private static AnnotationHandler CreateRectangleAnnotattion()
{
        RectangleAnnotation rect =
        AnnotationGenerator.CreateRectangleAnnotation(new
        RectangleF(0, 0, 100, 100));
        //Create a fill rectangle
        rect.Fill = new
        RasterEdge.Imaging.Annotation.Basic.AnnotationBrush();
        rect.Fill.FillType = FillType.Solid;
        //Only set color to RGBA(255,182,182,182) and you can get a pure
        white block
        rect.Fill.Solid_Color = new REColor(Color.FromArgb(255, 182,
        182, 182));
       return rect;
}
```

Add Image Annotation

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

C#

```csharp
//open a TIFF file
TIFFDocument tifDoc = new TIFFDocument(@"C:\input.tif");
//get specified tiff page
TIFFPage page = (TIFFPage)tifDoc.GetPage(0);
//load an image
BaseImage image = new REImage(@"C:\logo.png");
//set the image annotation location
PointF position = new PointF(200f,200f);
//add image annotation on the specified page location
page.AddImage(image, position);
tifDoc.Save(@"C:\output.tif");
```

Related APIs (**TIFFPage.cs**):

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

***Description*:**

Add an image on the page with specified location.

***Parameters*:**

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

```csharp
public override void AddImage(BaseImage image, Rectangle targetRect,
ImageCompress compress, TransformType type);
```

***Description*:**

Add an image annotation to specified Rectangle with image compression and transform on the tiff page.

***Parameters*:**

Name	Description	Valid Value
image	the image to add on the page	can't be null
targetRect	The erea that the image annotation will be added	-
compress	The compression method of added image annotation	The value listed in the ImageCompress.cs
type	Transform type	The value listed in the TransformType.cs

```csharp
public override void AddImage(BaseImage image, PointF point, float scale,
float left, float top, float right, float bottom);
```

***Description*：**

Add an image annotation to specified page position with different scaling.

***Parameters*：**

Name	Description	Valid Value
image	the image to add on the page	can’t be null
point	location of the image	x>=0 && x<=page.width y>=0 && y<=page.height
scale		
left		
top		
right		
bottom		

Save TIFF file

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

C#

```
//load TIFF file Document object from file path
String inputPath = @"F:\input.tif";
TIFFDocument tifDoc = new TIFFDocument(inputPath);
//save the tiff Document object to the file path
tifDoc.Save(@"F:\output.tif");
```

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

```
public override void Save(string filePath);
```

***Description*:**

Save TIFF document object to the given file path.

***Parameters*:**

Name	Description	Valid Value
filePath	output file path	a valid file path

```
public override void SaveToStream(Stream stream);
```

***Description*:**

Save TIFF document object to stream.

***Parameters*:**

Name	Description	Valid Value
stream	output stream	a valid stream

```
public override byte[] SaveToBytes();
```

***Description*:**

Save TIFF document object to byte array.

***Return*:**

A byte array, null if failed.