

TiffDLL90

Version 9.10

Informatik Inc

User Guide

Please Print this User Guide

Always ensure that you have the latest Program and User Guide.
Program and User Guide can be downloaded from www.informatik.com.

July 1, 2010

Table of Contents

Read This First.....	3
Demo Version, Licensed Version	3
First Project	4
How to Use TiffDLL90	4
Sample Code	6
Iterate through Pages.....	7
Functions.....	7
Open File	8
Save File.....	9
Resize Image.....	12
Resize Canvas.....	12
Crop Image	13
Shift Image	13
Margin Cleanup	14
Watermarks	14
Annotations (Extended)	15
Annotations (Simple)	17
Annotations (Multiple Lines)	18
Image Inserts	19
Redactions (Blackout, Whiteout).....	21
Highlighting (Color Background).....	21
Color Flooding	21
Invert Image.....	22
Rotation	22
Flip.....	22
Resolutions.....	23
Byte Parity	23
Information.....	24
Registration Code	25
Combine Files	25
Split Files.....	26
Complex Conversion	26
Printing.....	27
Draw Shapes.....	27
Order of Processing Functions	29
OCR Optical Character Recognition.....	29
Error codes.....	30
Distribution / Deployment	31
Technical Support	31
License, Warranty, Disclaimer.....	32
Copyright and Trademarks	33

Read This First

The TiffDLL90 files can be found in the C:\Program Files\Informatik Inc\TiffDLL90 folder.

TiffDLL90 requires Microsoft .NET Framework 2.0 or later.

Although you have an option to append to an existing file (TIFF only) and an option to alert you if a file already exists, by default all existing files will be overwritten without warning. Your code should always check the existence of a file before you save a file to the same file name.

Functions can be combined. For example you can do text annotations, watermarks, resize all in one process. Keep in mind that the functions are run in a pre-determined order, and each function builds on the result of the previous function. See section on Order of Processing Functions below.

Metadata and non-structural TIFF tags (such as Description, Copyright, etc. are not preserved.

Please contact Technical Support if you require a StrongName compiled version of the TiffDLL90.

Supplied AS IS without liability by developer, distributor, supplier. Always ensure that all affected files are safely backed up.

Demo Version, Licensed Version

You may use the Demo Version for 30 days only. After 30 days, the TiffDLL90 software must be uninstalled or a license must be purchased. The demo version displays a Demo message and annotates processed documents with a 'DEMO' notation at random. Use the Demo version only for testing. Always back up affected files before running the Demo version as the Demo notation placed on documents may render the documents unusable commercially.

For licensed versions, enter the **Registration Code** (license code) in the `_RegistrationCode` property. The license code will be given to you when you purchase the license. The license code must be kept confidential; never distribute the license code with your application. Please read the Registration Code section below.

First Project

To use TiffDLL90 in your project, use these simple steps:

- Copy the **TiffDLL90.dll** file and the **Tifftek32.dll** file to your project folder (typically the Bin folder, or the folder where your project's executable is located).
- Set the CPU Target to x86 so that your application also runs on 64-bit machines. The setting is made in Project -> Properties -> Compile -> Advanced Compile Options.
- In your project window, go to Project -> Add Reference. Browse to and select the TiffDLL90.dll file.
- In your project declare the TiffDLL90 as an object, for example with code like

```
TiffDLL myobj = new TiffDLL90.TiffDLL();      for C#  
or  
Dim myobj As New TiffDLL90.TiffDLL.          for VB.NET
```

- Optionally add the placeholder notation at the top of your project with

```
Using Tiffdll90;      for C#  
or  
Imports TiffDLL90    for VB.NET
```

- Now, you can access all functions, methods and properties of TiffDLL90 with simple dot-notation and Intellisense. Intellisense works really well in VB.NET but has limitations in C-Sharp (see details in section below).
- **IMPORTANT!** TiffDLL90 is a 32-bit application. When compiling your project, set the CPU Target to x86 so that your application also runs on 64-bit machines. The setting is made in Project -> Properties -> Compile -> Advanced Compile Options.
- When deploying your application, don't forget to include the TiffDLL90.dll and the Tifftek32.dll files.

How to Use TiffDLL90

The syntax of TiffDLL90 is very simple but you may see it as somewhat atypical. You specify each property (argument) in a separate code line. The source file, the destination file, the graphics format, the image size, etc, each is a property (argument). Then, when all properties are defined, you add the 'Run' function to your code. The Run function does not

itself have arguments, as each argument has been given in separate code lines. The `OpenFile` and the `SaveFile` properties are mandatory; all other properties are optional.

Step by Step:

Before you start, make sure that you have the `TiffDLL90.dll` file and the `Tifftek32.dll` file in the folder from which the application will be run; in development mode this typically would be the `Bin` folder.

- Specify the source and the destination file with `_OpenFile.FileName` and `_SaveFile.FileName`. Add code lines for conversion specifications. Add the `Run()` command.
- Each commandline starts with the object, for example `myobj._Cropimage`.
- Each code line is for a single property setting. Add as many code lines as necessary. Most properties have a default and need not be set.
- Always use dot-notation. Type the object name, for example 'myobj' in these samples, then enter a period (.) and select an option from the drop down list; type a period again and select the next level option (if available). At the end of the options tree, type an equal sign and select from the list of enumerations (displayed by Intellisense) or enter the value. Never try to type the code yourself, other than text such as file names. Enumerations are listed immediately in VB.NET after you type an equal sign. C# does not immediately list enumerations and you may need to refer to the User Guide, or you can simply enter the first character of the enumeration list name. The enumeration list name is always the same as the property name. For example in the following example you would type a character B after the equal sign.

```
obj._Redaction.BlackOrWhite = BlackOrWhite.Black.
```
- Each commandline ends with an equal sign (=) and the property value.
- Never use brackets (exception to the rule is the `Run` command where the system automatically adds brackets at the end of the command).
- After specifying all the properties, end the code with the `Run()` command.
- Always check for returned error codes. The `Run()` returns zero (0) on success or an error code. For Error Codes see listing below.

You can combine many formatting options. The formatting options are executed in a pre-determined order, not in the sequence of the code lines. Please see the section '**Order of Processing Functions**' below. There may be some combinations of options that, because of the order of execution, may produce an unexpected result.

When you deploy the application you must include the `TiffDLL90.dll` file and the `Tifftek32.dll` file.

Sample Code

VB.NET Code

```
Dim myobj As New TiffDLL90.TiffDLL
Dim sourcefile As String = "gibberish.tif"
Dim outputfile As String = "test.tif"
myobj._OpenFile.Filename = sourcefile
myobj._SaveFile.Filename = outputfile
myobj._SaveFile.Format = TiffDLL90.Format.Auto
myobj._SaveFile.OverwriteFile = TiffDLL90.OverwriteFile.Overwrite
myobj._ResizeCanvas.WidthInches = 9
myobj._ResizeCanvas.HeightInches = 12
myobj._ResizeCanvas.ImageCentered = True
myobj._Watermark.Text = "HELLO"
myobj._Watermark.WatermarkPlacement = TiffDLL90.WatermarkPlacement.Center
myobj._Watermark.HighQuality = True
myobj._AnnotationSimple.Text = "This is a Test"
myobj._AnnotationSimple.PlacementBottom = TiffDLL90.PlacementBottom.Center
Dim result As Integer = myobj.Run()
If result = 0 Then
    MessageBox.Show("Done")
Else
    MessageBox.Show("Error " & result)
End If
```

C# Code

```
TIFFDLL90.TiffDLL myobj = new TIFFDLL90.TiffDLL();
string sourcefile = "gibberish.tif";
string outputfile = "test.tif";
myobj._OpenFile.Filename = sourcefile;
myobj._SaveFile.Filename = outputfile;
myobj._SaveFile.Format = TIFFDLL90.Format.Auto;
myobj._SaveFile.OverwriteFile = TIFFDLL90.OverwriteFile.Overwrite;
myobj._ResizeCanvas.WidthInches = 9;
myobj._ResizeCanvas.HeightInches = 12;
myobj._ResizeCanvas.ImageCentered = true;
myobj._Watermark.Text = "HELLO";
myobj._Watermark.WatermarkPlacement = TIFFDLL90.WatermarkPlacement.Center;
myobj._Watermark.HighQuality = true;
myobj._AnnotationSimple.Text = "This is a Test";
myobj._AnnotationSimple.PlacementBottom = TIFFDLL90.PlacementBottom.Center;
int result = myobj.Run();
if (result == 0)
{
    MessageBox.Show("Done");
}
else
{
    MessageBox.Show("Error " + result);
}
```

Please download the sample projects:

www.informatik.com/files/TiffDll90_SampleVBNET.zip

www.informatik.com/files/TiffDll90_SampleCSharp.zip

Iterate through Pages

If you need to handle pages individually, you must obtain the number of pages in the file, then iterate thru the pages. For some sample code see the Sample Code in the 'Split Files' section below.

Functions

The main functions of TiffDLL90 are:

- Open File
- Save File
- Resize Image
- Resize Canvas
- Crop Image
- Shift Image
- Margin Cleanup
- Watermarks
- Text Annotations
- Image Insert
- Inversion
- Rotation, Conditional Rotation
- Redaction (Black or White)
- Highlighting
- Color Flooding
- Resolution
- Flip
- Image Information (Properties)
- Split Files into Pages (Serialization)
- Combine Files into a Multipage TIFF
- Printing (see section below)

For other functions, such as drawing shapes, etc. see Drawing Shapes section below.

Open File

_OpenFile

Filename	String
PageSelection	Integer
PageRange As PageRange	
FromPage	Integer
ToPage	Integer

A source file name must always be specified

Page Selection:

To open a specific page of a multipage file, use the PageSelection option (1 = First Page). For a range of pages, use the PageRange option and specify both the starting page number and the ending page number. For the last page number you may enter a minus 1 (-1). The range may be in ascending or descending order. Using the descending order is useful if you need to reverse the page order of a multipage file. Note, only TIFF files can be multipage and will honor the page range.

```
myobj._OpenFile.Filename = "c:\mydir\myfile.tif"
myobj._OpenFile.PageSelection = 1
Or
myobj._OpenFile.PageRange.FromPage = 3
myobj._OpenFile.PageRange.ToPage = 10
```

Instead of a file path you can enter the word 'bitmap' as a file name and TiffDLL90 will load the myobj.Sourcebitmap memory bitmap (which your application must create first). If necessary, use the .NET Clone option to re-assign/copy bitmaps. See also Complex Conversions section below.

```
myobj.Sourcebitmap = New Bitmap("c:\mydir\mypicture.png")
myobj._OpenFile.Filename = "bitmap"
```

To open a file just for property information, without conversions and without saving the file, use the keyword 'INFO' as the SaveFile filename. You rarely, if ever, need that option since the property data is populated automatically when a file is opened.

```
myobj._OpenFile.Filename = "c:\mydir\myfile.tif"
myobj._SaveFile.Filename = "INFO"
result = myobj.Run
dim width as integer= myobj._Information.Width
dim height as integer= myobj._Information.height
```

See also section on Information property below.

Save File

SaveFile

Filename	String
Format	
<i>Format.Auto</i> *	Enum Default
Format.TIFF_Auto **	Enum
Format.TIFF_CCITT4_1bit	Enum
Format.TIFF_CCITT3_1bit	Enum
Format.TIFF_LZW_1bit	Enum
Format.TIFF_LZW_8bit	Enum
Format.TIFF_LZW_24bit	Enum
Format.TIFF_JPEG_24bit	Enum
Format.TIFF_Packbits_24bit	Enum
Format.TIFF_Uncompressed_1bit	Enum
Format.TIFF_Uncompressed_8bit	Enum
Format.TIFF_Uncompressed_24bit	Enum
Format.GIF_8bit	Enum
Format.BMP_1bit	Enum
Format.BMP_8bit	Enum
Format.BMP_24bit	Enum
Format.PNG_1bit	Enum
Format.PNG_8bit	Enum
Format.PNG_24bit	Enum
Format.JPEG_24bit	Enum
Format._MemoryBitmap1Bit	Enum
Format._MemoryBitmap8Bits	Enum
Format._MemoryBitmap24Bits	Enum
Format._MemoryBitmapAuto	Enum
SplitFile	
<i>PageSplitActive</i>	Boolean False
SerialStartNumber	String
SerialNumberDigits	Integer
OverwriteFile	
<i>OverwriteFile.Overwrite</i>	Enum Default
OverwriteFile.AppendTIFFonly	Enum
OverwriteFile.ShowError	Enum
TIFFFillorder2	
<i>TIFFFillorder2pOptions.Unch</i>	Enum Default
TIFFFillorder2Options.Yes	Enum
TIFFFillorder2Options.No	Enum
TIFFMultiStrip	
<i>TIFFMultiStripOptions.Unch</i>	Enum Default
TIFFMultiStripOptions.Yes	Enum
TIFFMultiStripOptions.No	Enum
JPEGquality	Integer Default = 75
ByteParity	Boolean False

* **Format.Auto** uses any format from source file

** **Format.TIFF_Auto** uses the TIFF format from source file.

The filename must always be specified. Specify the full file path where the processed file should be saved, including the drive, folder and file extension.

For multipage TIFF files, the destination file name must not be the same name as the source file name. Save the file to a temporary folder, then have your application rename it back to the original name.

Specify a graphics format. To use the format of the source file, select the Auto format. Keep in mind that The TIFF CCITT3 and CCITT4 support only black and white files. The best format for black and white images is CCITT4. For photos in TIFF files you should choose the JPEG-compressed TIFF format.

For JPEG files you set a Quality Factor of 5 to 100. The smaller the value the smaller the file but with reduced quality. The default JPEG Quality is 75; is is recommended that you do not change it.

If the output of a multi-page TIFF file must be split into single-page files, set the PageSplitActive and optionally define the starting serial number and the number of digits. If the starting number and the digits are not specified, the first page will be given the 0001 suffix. Any suffix separator must be added to the file name. For example a given file name of c:\mydir\myfile_tif will be saved as c:\mydir\myfile_0001.tif. For sample code, see section below.

To combine files into one multi-page TIFF file set the AppendTIFFonly property. For sample code see section below.

Unless an Overwrite option is specified, source files are overwritten by same-name destination file names. To append a file to an existing TIFF file, set the OverwriteFile to Append.

Tiff files can be single-strip (default) and multi-strip and can be of Fillorder 1 (default) or 2. Normally, you need not change these settings. To set, use the TIFFFillorder2 and TIFFMultiStip options. The system tries to preserve the settings of the source file.

Rarely, you may have a need that the width and height of the saved images have Byte-Parity, ie. have a width and height value in pixels that is divisible by Eight. To set, use the ByteParity function.

Sample Code:

```
myobj._SaveFile.Filename = "c:\aaa\aaa_out3.tif"  
myobj._SaveFile.Format = TiffDLL90.Format.Auto  
myobj._SaveFile.OverwriteFile = TiffDLL90.OverwriteFile.AppendTIFFonly
```

If you need to create a memory bitmap instead of a file, keep the Filename blank and select a color-depth (1-bit, 8-bit or 24-bit) for the bitmap. The bitmap can then be retrieved as myobj.BMP.

```
myobj._SaveFile.Format = TiffDLL90.Format._MemoryBitmap8Bits
```

You may want to clone the myobj.BMP for your other uses, like

```
Dim myBitmap as bitmap= myobj.BMP.clone
```

To open a file just for property information, use the keyword 'INFO' as the SaveFile filename. See also section on Information property below. See also Complex Conversions section below.

Resize Image

_ResizeImage

ResizeMode	
<i>ResizeMode.Inches</i>	Enum Default
<i>ResizeMode.Pixels</i>	Enum
<i>ResizeMode.Percent</i>	Enum
<i>ResizeMode.PlusMinusInches</i>	Enum
<i>ResizeMode.PlusMinusPixels</i>	Enum
Width	Double
Height	Double
ConstrainProportions	Boolean False
PreserveCanvas	Boolean False
PreserveCanvas_Centered	Boolean False
AdjustResolution	Boolean False

The `ResizeImage` option resizes the image bitmap (increasing or reducing the pixels if the image). Resizing has a direct effect on the file size and the quality of the image.

The new size of the image can be specified in Inches, pixels, percentage of the original, increase or decrease in inches or pixels. If the `ConstrainProportions` are set and you change the size of only one dimension (either width or height), the other dimension will also change to keep the original width-to-height ratio. You can reduce the inside image without changing the overall size of the canvas and in doing so you have an option to center the original image. Finally, so that the resized image is displayed (or printed) at the original size the system can automatically adjust the Resolution.

If you are not familiar with Inches and like to deal with Centimeters, do a simple calculation:
2.54 Centimeters = 1 Inch.

Resize Canvas

_ResizeCanvas

WidthInches	Double
HeightInches	Double
ImageCentered	Boolean False

The canvas is the background on which the image is shown. When you change the canvas, the image remains at the original size; only the canvas increases or decreases. Increasing the canvas creates white margins (right and bottom); decreasing the canvas truncates the image. When you increase the canvas the `ImageCentered` option places the image in the center of the enlarged canvas.

If the canvas is resized to a size smaller than the image, then the ImageCentered option is not applied.

Crop Image

_CropImage

LeftInches	Double
TopInches	Double
WidthInches	Double
HeightInches	Double
LeftPixels	Integer
TopPixels	Integer
WidthPixels	Integer
HeightPixels	Integer
PreserveCanvas	Boolean False
PreserveCanvas_Centered	Boolean False

The CropImage function extracts a specified rectangle of the image into a new file. All specifications are in Inches. An option is available to preserve the original canvas. The cropped image can be placed in the upper-left corner of the canvas or centered.

If WidthInches and/or HeightInches is given as negative values, the crop area will extend to the right and bottom of the image less the WidthInches and HeightInches so specified. For example, if you specify Left and Top as 1.25 and Right and Bottom as -1.25, then all four sides of the image will be cropped by 1.25 inches.

If the specified WidthInches and/or HeightInches extends outside the image boundry, the values will be adjusted back to fit the image.

Shift Image

_ShiftImage

LeftInches	Double
TopInches	Double

The ShiftImage option moves (shifts) the image but keeps the canvas unchanged. Shifting the image down, for example, creates a white margin on top but may truncate part of the image at the bottom. The image can be shifted in any direction. To shift left or up, use negative values.

Margin Cleanup

<u>MarginCleanup</u>	
LeftInches	Double
TopInches	Double
RightInches	Double
BottomInches	Double

The MarginCleanup option allows you to white-out any or all of the four margins (left, top, right, bottom). The option is useful if you like to clean up the white border areas, for example to remove punch holes.

Watermarks

<u>Watermark</u>	
Text	String
Transparency	
Transparency.Opaque	Enum
Transparency.Transparency10	Enum
Transparency.Transparency20	Enum
Transparency.Transparency30	Enum
Transparency.Transparency40	Enum
Transparency.Transparency50	Enum
<i>Transparency.Transparency60</i>	Enum Default
Transparency.Transparency70	Enum
Transparency.Transparency80	Enum
Transparency.Transparency90	Enum
WatermarkStyle	
<i>WatermarkStyle.Outline</i>	Enum Default
WatermarkStyle.Solid	Enum
WatermarkPlacement	
WatermarkPlacement.Upper	Enum
<i>WatermarkPlacement.Center</i>	Enum Default
WatermarkPlacement.Lower	Enum
WatermarkSize	
<i>WatermarkSize.Regular</i>	Enum Default
WatermarkSize.Larger	Enum
WatermarkSize.Smaller	Enum
Brightness	
<i>Brightness.Regular</i>	Enum Default
Brightness.Darker	Enum
Brightness.Lighter	Enum
HighQuality	Boolean

The Watermark option allows you to add watermark stamps to the image. There are two styles of watermarks: Solid and Outline. Solid watermarks insert the text of the watermark in semi-transparent gray, the Outline watermarks use 'hollow' letters for the watermark text. The watermarks can be placed in the center, upper third or lower-third of the page. The default transparency is 60% but you can increase or decrease the transparency with the Transparency or Brightness setting. Not all options are available for monochrome images.

For more versatility, instead of the Watermark option you can use the InsertImage or Annotation options.

The HighQuality option creates a higher quality watermark, but at a much reduced performance. The HighQuality option is effective only for monochrome source files. If the source file is a color file, watermarks are automatically created in 'high quality'.

Annotations (Extended)

AnnotationExtended

Text	String
Placement	
<i>Placement.HorizontalBottomLeft</i>	Enum Default
<i>Placement.HorizontalBottomCenter</i>	Enum
<i>Placement.HorizontalBottomRight</i>	Enum
<i>Placement.HorizontalTopLeft</i>	Enum
<i>Placement.HorizontalTopCenter</i>	Enum
<i>Placement.HorizontalTopRight</i>	Enum
<i>Placement.VerticalLeftTop</i>	Enum
<i>Placement.VerticalLeftCenter</i>	Enum
<i>Placement.VerticalLeftBottom</i>	Enum
<i>Placement.VerticalRightTop</i>	Enum
<i>Placement.VerticalRightCenter</i>	Enum
<i>Placement.VerticalRightBottom</i>	Enum
<i>Placement.CenterPage</i>	Enum
CustomLocationLeftInches	Double
CustomLocationTopInches	Double
FontSize	
FontSize.Size06	Enum
FontSize.Size07	Enum
FontSize.Size08	Enum
FontSize.Size09	Enum
FontSize.Size10	Enum
FontSize.Size11	Enum
<i>FontSize.Size12</i>	Enum Default
FontSize.Size14	Enum
FontSize.Size16	Enum
FontSize.Size18	Enum
FontSize.Size20	Enum
FontSize.Size22	Enum
FontSize.Size24	Enum
FontSize.Size26	Enum
FontSize.Size28	Enum
FontSize.Size36	Enum
FontSize.Size48	Enum

FontSize.Size72	Enum
FontName	
<i>FontName.Arial</i>	Enum Default
FontName.MicrosoftSansSerif	Enum
FontName.TimesRoman	Enum
FontName.ComicSansMS	Enum
FontName.CourierNew	Enum
Color	Color
Font_Custom	Font
Transparency	
<i>Transparency.Opaque</i>	Enum Default
Transparency.Transparency10	Enum
Transparency.Transparency20	Enum
Transparency.Transparency30	Enum
Transparency.Transparency40	Enum
Transparency.Transparency50	Enum
Transparency.Transparency60	Enum
Transparency.Transparency70	Enum
Transparency.Transparency80	Enum
Transparency.Transparency90	Enum
LateralPaddingInches	Double

There are three methods of Annotations. The AnnotationExtended is the version that offers the most extensive options.

Unless you need the many options in this version, for monochrome files you should use the AnnotationSimple (or AnnotationMultiLine) method which runs several times faster.

Specify the text of the insert. The text can be a single-line or a multiple-line string. Specify the location of the annotation. There are nine (9) predefined locations around the margins of the image, or centered in the image. Text inserts are left-adjusted, centered or right-adjusted depending on the location. For multi-line text use the following syntax:

```
"Hello World," & ControlChars.CrLf & "Life is Beautiful"
```

You can also specify the location in Inches by reference to left or top border. To specify the location by reference to the right and bottom border, use negative values.

Select the font from a list of common font names, or specify your own font. If you specify your own font, enter it as a .NET Font object.

```
Dim fnt As Font = New Font("Arial", 12)
or
Dim fnt As Font = New Font("Arial", 12, FontStyle.Bold Or
FontStyle.Italic)

myobj._AnnotationExtended.Font_Custom = fnt
```

By default, the text color is black. You can specify your own color as a .NET Color object.

```
myobj._AnnotationExtended.Color = Color.Red
```

For transparent rendition of the text, use the Transparency option.

LateralPadding adds some space in below the text. For example a value of 0.125 will add 1/8 Inch of padding.

In your text you may want to add values generated by the TiffDLL90 process. For example, when processing a range of pages you may want to refer to the page number that is being processed. The system uses two placeholders: [p] for the currently processed page number and [pp] for the number of pages in the source file. The characters p in the placeholder must be lower-case.

```
Dim txt As String = "Page [p] of [pp]"  
myobj._AnnotationExtended.Text = txt
```

You can also included computed text, for example:

Current date in a custom format:

```
Dim dt As Date = (Now)  
Format(dt, "dd MMMM yyyy")
```

The File's base file name:

```
IO.Path.GetFileNameWithoutExtension(myobj._SaveFile.Filename)
```

If you need to remove an annotation, consider using the Redaction option (white).

Annotations (Simple)

_AnnotationSimple

Text	String
PlacementHorizontal	
<i>PlacementHorizontal.BottomLeft</i>	Enum Default
<i>PlacementHorizontal.BottomCenter</i>	Enum
<i>PlacementHorizontal.BottomRight</i>	Enum
<i>PlacementHorizontal.TopLeft</i>	Enum
<i>PlacementHorizontal.TopCenter</i>	Enum
<i>PlacementHorizontal.TopRight</i>	Enum
SmallerFont	Boolean False
Opaque	Boolean False

AnnotationSimple is a fast method of inserting plain text (single-line) at the bottom or top of the document, either left-adjust, centered or right-adjust. The font size is fixed to two sizes. By default, the text is rendered with a transparent background. The Opaque option has an opaque white background.

In your text you may want to add values generated by the TiffDLL90 process. For example, when processing a range of pages you may want to refer to the page number that is being

processed. The system uses two placeholders: [p] for the currently processed page number and [pp] for the number of pages in the source file. The characters p in the placeholder must be lower-case. See code sample in the AnnotationExtended section above.

If you need color, semi-transparency, special font styles and sizes, multi-line text, etc, please use the AnnotationExtended function.

You can also include computed text, for example:

Current date in a custom format:

```
Dim dt As Date = (Now)
Format(dt, "dd MMMM yyyy")
```

The File's base file name:

```
IO.Path.GetFileNameWithoutExtension(myobj._SaveFile.FileName)
```

If you need to remove an annotation, consider using the Redaction option (white).

Annotations (Multiple Lines)

AnnotationMultipleLine

SpecificationFile	String
FontSize_Optional	Double
Font_Optional	Font
Opaque	Boolean False

The AnnotationMultiLine options allows you to enter text annotations at several locations. The text lines are specified in the following format:

```
Hello World; 1.5; 2.5
Hello World; 2.5; 5
Hello World; 2.5; 8
```

The two numeric values specify the Left and Top location of the text in inches.

To use the option, specify the text file path in the SpecificationFile. By default, the font is Arial and the size for all inserts is 12, but you can specify a different value in Font and Size. Opaque adds a white opaque background behind the text.

To specify your own font use the following syntax:

```
Dim fnt As Font = New Font("Arial", 12)
or
Dim fnt As Font = New Font("Arial", 12, FontStyle.Bold Or
FontStyle.Italic)

myobj._AnnotationMultipleLine.OptionalFont=fnt
```

In your text you may want to add values generated by the TiffDLL90 process. For example, when processing a range of pages you may want to refer to the page number that is being processed. The system uses two placeholders: [p] for the currently processed page number and [pp] for the number of pages in the source file. The characters p in the placeholder must be lower-case. See code sample in the AnnotationExtended section above.

There are two (2) AnnotationExtended instances. The second instance may be used for different fonts.

You can also include computed text, for example:

Current date in a custom format:

```
Dim dt As Date = (Now)
Format(dt, "dd MMMM yyyy")
```

The File's base file name:

```
IO.Path.GetFileNameWithoutExtension(myobj._SaveFile.FileName)
```

Image Inserts

_ImageInsert

ImageFileName	String
LeftInches	Double
TopInches	Double
SizePercentage	Double
WidthInches	Double
HeightInches	Double
Transparency	
<i>Transparency.Opaque</i>	Enum Default
Transparency.Transparency10	Enum
Transparency.Transparency20	Enum
Transparency.Transparency30	Enum
Transparency.Transparency40	Enum
Transparency.Transparency50	Enum
Transparency.Transparency60	Enum
Transparency.Transparency70	Enum
Transparency.Transparency80	Enum
Transparency.Transparency90	Enum
MakeWhiteTransparent	Boolean False
ImageColorDepth	
<i>ImageColorDepth.KeepTo8bits</i>	Enum Default*
ImageColorDepth.ReduceToBlack	Enum
ImageColorDepth.Unchanged	num

The ImageInsert option allows you to insert an image. Specify the insert image file. The image must be of a supported format (TIFF, BMP, PNG, GIF, JPEG). The location of the insert is given in Inches by reference to the left and top border. If you need to specify the

location referenced to the right border or bottom border, use negative values. The image size is adjusted to correspond to the resolution of the document; you may resize it by a given percentage or you can specify your own size in Inches. If you only specify one of the dimensions (width or height) the unspecified dimension will be calculated so that the image keeps its original proportions. Optionally select a transparency factor. For black and white image inserts you may want the white to be fully transparent; choose the MakeWhiteTransparent option.

Instead of a file path you can enter the word 'bitmap' as a file name and TiffDLL90 will load the myobj.Insertbitmap memory bitmap (which your application must create first). If necessary, use the .NET Clone option to re-assign/copy bitmaps.

```
myobj.Insertbitmap = New Bitmap("c:\mydir\mypicture.png")
myobj._ImageInsert.ImageFilename = "bitmap"
```

If you wish to insert several images, save the result as an intermediary bitmap image and repeat the process: use code along the following example:

```
Dim result As Integer
Dim myobj As New TiffDLL90.TiffDLL
myobj._OpenFile.Filename = "c:\somedir\infile.tif"
myobj._SaveFile.Format = TiffDLL90.Format._MemoryBitmap24Bits
myobj._ImageInsert.ImageFileName = "c:\mystory\fireworks1.bmp"
myobj._ImageInsert.TopInches = 1
myobj._ImageInsert.LeftInches = 1
result = myobj.Run()
If result <> 0 Then
    MsgBox(result)
    Exit Sub
End If

myobj.Sourcebitmap = myobj.BMP.Clone
myobj._OpenFile.Filename = "bitmap"
myobj._ImageInsert.ImageFileName = "c:\mystory\fireworks2.bmp"
myobj._ImageInsert.TopInches = 4
myobj._ImageInsert.LeftInches = 1
myobj._SaveFile.Format = TiffDLL90.Format.TIFF_LZW_8bit
myobj._SaveFile.OverwriteFile = TiffDLL90.OverwriteFile.Overwrite
myobj._SaveFile.Filename = "c:\somedir\test.tif"
result = myobj.Run()
If result <> 0 Then
    MsgBox(result)
    Exit Sub
End If
```

* **ImageColorDepth.KeepTo8bits**: When inserting a color image into a monochrome image, by default, in order to keep the file size relatively small, the document will be saved as an 8-bit image. You may change the color depth of the saved image to the color depth of the insert image by selecting ImageColorDepth.Unchanged, or you may force it to monochrome with ImageColorDepth.ReduceToBlack.

Redactions (Blackout, Whiteout)

The Redaction option blanks out a rectangle in either black or white.

Redaction

LeftInches	Double
TopInches	Double
WidthInches	Double
HeightInches	Double
BlackOrWhite	
<i>BlackOrWhite.Black</i>	Enum Default
BlackOrWhite.White	Enum

Highlighting (Color Background)

The Highlight option colors the white background of a specified rectangle. Only a pure white background is colored).

Highlight

LeftInches	Double
TopInches	Double
WidthInches	Double
HeightInches	Double
Color	Color (Default = Yellow)

Color Flooding

The Color Flooding option fills the bordered area with a specified color. The flood area is defined by the specific XY pixel location and extends in all directions until a pixel of a different color is reached. Use only solid colors.

ColorFlooding

PixelX	Integer
PixelY	Integer
Color	Color (Default = White)
UseInchAsUnits	Boolean

Generally, use pixels as the unit of measurement. If you need to use Inches instead of pixels, set the UseInchAsUnits to True. For Centimeters is the factor 2.54 CM=1 Inch.

If you are running Color Flooding in a loop and your computer has limited memory add the following line at the end of each loop iteration (to force the garbage collection) in VB.NET:

```
System.GC.Collect()  
Dim totMem As Long = System.GC.GetTotalMemory(forceFullCollection:=True)
```

Invert Image

`_InvertImage` As Boolean

The `InvertImage` option inverts the image, for example black to white, or vice versa.

Rotation

`_RotateImage`

<code>RotationAngle.None</code>	Enum	Default
<code>RotationAngle.Degrees90</code>	Enum	
<code>RotationAngle.Degrees180</code>	Enum	
<code>RotationAngle.Degrees270</code>	Enum	
<code>RotationAngle.ConditionalIfLandscape90</code>	Enum	
<code>RotationAngle.ConditionalIfLandscape270</code>	Enum	
<code>RotationAngle.ConditionalIfPortrait90</code>	Enum	
<code>RotationAngle.ConditionalIfPortrait270</code>	Enum	

The `RotateImage` rotates the image at a specified right-angle. The Conditional rotations rotate the image only if the image is landscape or portrait as the case may be.

Flip

`_Flip`

<code>Flip.None</code>	Enum	Default
<code>Flip.ToptoBottom</code>	Enum	

The `Flip` option flips the image top-to-bottom (mirror image). For a horizontal flip, combine the flip option with a 180 degree rotation.

Resolutions

Resolution

ResolutionHorizontal	
<i>Resolution.NoChange</i>	Enum Default
Resolution.Res072	Enum
Resolution.Res096	Enum
Resolution.Res100	Enum
Resolution.Res144	Enum
Resolution.Res150	Enum
Resolution.Res192	Enum
Resolution.Res200	Enum
Resolution.Res300	Enum
Resolution.Res400	Enum
Resolution.Res500	Enum
Resolution.Res600	Enum
ResolutionVertical	
<i>Resolution.NoChange</i>	Enum Default
Resolution.Res072	Enum
Resolution.Res096	Enum
Resolution.Res100	Enum
Resolution.Res144	Enum
Resolution.Res150	Enum
Resolution.Res192	Enum
Resolution.Res200	Enum
Resolution.Res300	Enum
Resolution.Res400	Enum
Resolution.Res500	Enum
Resolution.Res600	Enum
ResolutionHorizontal_Other	Integer
ResolutionVertical_Other	Integer
Resample	Boolean False

The Resolution options allows you to change the horizontal and/or vertical resolution of the image with an option to resample the image. The resolution can be picked from a list of common resolution values or you can enter your own value. Resampling means that the image will be reduced by removing pixels if you reduce the resolution or increased by adding pixels with additional pixels if you increase the resolution (so that the image displays or prints at the original size).

Byte Parity

See Save File section above.

Information

Important: If you use the `_Information` data in your code, for number of pages, image properties, etc., you **MUST** have run the `Run()` function (typically with the 'INFO' argument) **before**. The `_Information` data is populated only by the `Run()` function for the file or page.

<code>_Information</code> (Read-Only)	
Pages	Integer
Width (Pixels)	Integer
Height (Pixels)	Integer
HorizontalResolution	Integer
VerticalResolution	Integer
ColorDepth	Integer
Format	Imaging.ImageFormat
CurrentPage	Integer (1=First Page)
JPEGResolutionUnit	Integer (1=Inch, 2=CM)
TIFFcompression	Integer
TIFFrowsperstrip	Integer
TIFFfillorder	Integer

The `Information` options returns various properties of the image, such as number of pages, width, height, current page being processed, etc. The information data elements are populated each time a file (page) is opened. Information data can then be used within the `TiffDLL90` code. Note, the `Run()` function must be run.

TIFFcompression: 1=Uncompressed, 3=CCITT3, 4=CCITT4, 5=LZW, 7=JPEG

Sample:

```
myobj._OpenFile.FileName = "c:\mydir\myfile.tif"
myobj._SaveFile.FileName = "INFO"
result = obj.Run
If myobj.Information.Width > myobj.Information.Height then
...
endif
...
Result=myobj.run()
```

You can also run `TiffDLL90` to simply obtain property information where you do not need to convert and save the file. You will need this option rarely, if ever, since the property information is populated automatically whenever you open a file page. The option may be useful if you need to run through a list of files to count the number of pages or to find out if files are landscape or portrait.

Example:

```
myobj._OpenFile.FileName = "c:\mydir\myfile.tif"
myobj._SaveFile.FileName = "INFO"
result = obj.Run
dim width as integer= myobj._Information.Width
dim height as integer= myobj._Information.height
...
```

Registration Code

`_RegistrationCode` String

For licensed TiffDLL90 the registration code must be entered. If not entered, the program will run in demo mode.

Global Maximum and Minimum Parameter settings:

`_Global_MaximumImageDimension` (Default 30,000)

`_Global_MinimumImageDimension` (Default 100)

`_Global_MaximumResolution` (Default 1,000)

`_Global_MinimumResolution` (Default 10)

Global Max/Min specifications are useful to avoid documents being created that are outside of a normal range, for example too small or far too large.

Combine Files

Files can be combined by simply setting the `AppendTIFFOnly`.

Sample Code:

```
myobj._OpenFile.FileName = "c:\mydir\File1.tif"
myobj._SaveFile.FileName = "c:\mydir\newfile.tif"
myobj._SaveFile.Format = TiffDLL90.Format.Auto
myobj._SaveFile.OverwriteFile = TiffDLL90.OverwriteFile.AppendTIFFOnly
result = myobj.Run()
```

```
myobj._OpenFile.FileName = "c:\mydir\File2.tif"
myobj._SaveFile.FileName = "c:\mydir\newfile.tif"
myobj._SaveFile.Format = TiffDLL90.Format.Auto
myobj._SaveFile.OverwriteFile = TiffDLL90.OverwriteFile.AppendTIFFOnly
result = myobj.Run()
```

Split Files

To split a multipage TIFF into serialized pages files, set the PageSerializationActive. Optionally, you can specify the serial starting number and the number of digits (for zero-fill). If the starting number and the number of digits are not specified, the system uses the starting number 1 and four digits.

Sample Code

```
myobj._OpenFile.FileName = "c:\mydir\myfile.tif"  
myobj._SaveFile.FileName = "c:\mydir\newfile.tif"  
myobj._SaveFile.Format = TiffDLL90.Format.Auto  
myobj._SaveFile.SplitFile.PageSplitActive = True  
myobj._SaveFile.SplitFile.SerialNumberDigits = 5  
myobj._SaveFile.SplitFile.SerialStartNumber = 1  
result = myobj.Run()
```

Alternatively, for more customized options, you can split pages by iterating thru the pages, like:

```
Dim myobj As New TiffDLL90.TiffDLL  
Dim result As Integer  
Dim filename As String = "c:\somedir\5PageDoc.tif"  
myobj._OpenFile.FileName = filename  
myobj._SaveFile.FileName = "INFO"  
result = myobj.Run()  
If result <> 0 Then  
    MessageBox.Show("Error " & result)  
    Exit Sub  
End If  
  
Dim numberpages As Integer = myobj._Information.Pages  
For i As Integer = 1 To numberpages  
    myobj._OpenFile.PageSelection = i  
    myobj._SaveFile.Format = TiffDLL90.Format.Auto  
    myobj._SaveFile.FileName = "c:\somedir\" & Format(i, "0000") & "xyz.tif"  
    result = myobj.Run()  
    If result <> 0 Then  
        MessageBox.Show("Error " & result)  
        Exit Sub  
    End If  
Next
```

Complex Conversion

Each Run call can make several conversions; they are executed in a pre-determined order. See Order of Processing Functions section below. For complex conversions you may need more than one Run. In such case you should retain the output bitmap of the first object and use it as the input for the second object.

Example:

```
Dim result As Integer
Dim myobj As New TiffDLL90.TiffDLL
myobj._OpenFile.FileName = "c:\somedir\somefile.tif"
'.....code
'.....code
result = myobj.Run()
If result <> 0 Then
    MsgBox(result)
    Exit Sub
End If
Application.DoEvents()

Dim bmp As Bitmap = myobj.BMP.Clone

myobj = Nothing
myobj = New TiffDLL90.TiffDLL

myobj.Sourcebitmap = bmp
myobj._OpenFile.FileName = "bitmap"
'.....code
'.....code
result = myobj.Run()
If result <> 0 Then
    MsgBox(result)
    Exit Sub
End If

If bmp Is Nothing = False Then
    bmp.Dispose()
    bmp = Nothing
End If
```

Printing

TiffDLL90 can be used for printing single-page or multi-page files, optionally with conversion functions. For a sample code of a printing project please download the file

www.informatik.com/files/Tiffdll90_SamplePrintVBNET.txt

Draw Shapes

The code sample below (VB.NET) demonstrates how you can use TiffDLL90 to help in drawing shapes and graphics. Basically, you open the file with TiffDLL90, save it to a

working bitmap, then draw your shapes on the bitmap, and finally save the bitmap with the help of TiffDLL90.

```
Dim result as Integer
myobj._OpenFile.FileName = "c:\somedir\somefile.tif"
myobj._SaveFile.Format = TiffDLL90.Format._MemoryBitmap24Bits
result = myobj.Run()
If result <> 0 Then
    MsgBox("Error " & result)
    Exit Sub
End If

Dim g As Graphics = Graphics.FromImage(myobj.BMP)
g.SmoothingMode = Drawing2D.SmoothingMode.AntiAlias

'draw a box
g.DrawRectangle(New Pen(Color.Red, 8), New Rectangle(100, 100, 500, 500))

'draw an arrow
Dim pn As Pen = New Pen(Color.Blue, 12)
Dim ArrowCap As New Drawing2D.AdjustableArrowCap(6, 6)
pn.CustomEndCap = ArrowCap
g.DrawLine(pn, 100, 900, 600, 1200)
'or specified in Inches:
Dim resx as integer = myobj._Information.HorizontalResolution
Dim resy as integer = myobj._Information.VerticalResolution
g.DrawLine(pn, 1.0 * resx, 3 * resy, 2 * resx, 4 * resy)

myobj.Sourcebitmap = myobj.BMP
myobj._OpenFile.FileName = "sourcebitmap"
myobj._SaveFile.Format = TiffDLL90.Format.TIFF_LZW_24bit
myobj._SaveFile.FileName = "c:\somedir\newfile.tif"
result = myobj.Run()
If result <> 0 Then
    MsgBox("Error " & result)
    Exit Sub
End If
g.dispose
```

Order of Processing Functions

1. Inversion
2. Rotation
3. Flip
4. MarginCleanup
5. Highlighting
6. Crop
7. Resize
8. Shift
9. Redaction
10. Resolution
11. Canvas
12. Insert Image
13. Annotations
14. Watermarks
15. ByteParity

The formatting options are executed in a pre-determined order, not in the sequence of the code lines.

OCR Optical Character Recognition

For OCR options please download the separate OCR DLL from www.informatik.com. Separate low license fee applies.

Error codes

- 1004 Cannot load source bitmap
- 1007 Destination File already exists. You can disable the Save_Overwrite option.
- 1008 Width or height of processed image are outside the set limits (max or min). You can change the Min/Max in Global_MaximumImageDimension.
- 1009 The resolution of processed image is outside set limits (max or min). You can change the Min/Max in Global_MaximumResolution.
- 1010 Cannot open TIFF file. File does not exist or is of an incompatible TIFF Format
- 1011 Cannot open page of TIFF file.
- 1012 Cannot open file (format other than TIFF)
- 1013 Cannot open file. Probably an unsupported graphics format or a corrupted file.
- 1014 Two possibilities: a) Cannot find Tifftek32 file. The Tifftek32 file must be in the same folder as the application. b) If developing on a 64-bit machine, you must set the project's CPU Target to x86; see user guide.
- 1021 JPEG Encoder/Save problem
- 1029 Problem saving file. Make sure that the destination file is not open by another application and that the file and the folder are not Read-Only.
- 1031 Error rotation
 - 1051 Error Resizing
- 1052 Width AND Height must be specified, or set ConstraintProportions=True
- 1061 Error Cropping specification. Crop width and height must be specified.
- 1071 Error setting new resolution
- 1081 Error Inversion
- 1091 Error with AnnotationSimple
- 1092 Text file for AnnotationMultiLine not found
- 1093 Error AnnotationMultiLine
- 1094 Error Watermarks
- 1111 Error saving file
- 1121 Error createvimage. Also, make sure that the Tifftek32.dll is in the correct folder.
- 1122 Error InsertBMPreduceMono
- 1131 Error adjusting color depth
- 1141 Error opening InsertImage file. File may not exist or may be of incompatible format
- 1151 Error inserting image
- 1161 Error annotation
- 1181 Error Redaction
- 1186 Error Highlighting
- 1212 Error creating output bitmap
- 1214 Error Image Insert (Fast)
- 1215 Error creating watermark bitmap
- 1216 Error Watermark Insert (Fast)
- 1217 For multipage file processing, destination file must not be the same as the source file
- 1221 Error Flipping
- 1231 Error ColorFlooding
- 1232 Error ColorFlooding. Make sure that the Units of Measurements are correct. See UseInchAsUnits property.

- 2001 Source File Name not specified
- 2002 Destination File Name not specified. Alternatively you can set the _SaveFile.Format to a MemoryBitmap. To obtain properties only, use keyword 'INFO' instead of a file name.
- 2003 Incorrect page specification. Both From and To must be specified. For unknown Lst Page use -1
- 2004 Incorrect Percentage specification. Value must be full number. Example, enter 50 for 50%
- 2005 Incompatible source specification. For output bitmap the output file name must be blank.
- 2006 Incompatible Serialization specification. Start number must be blank if Serialization is not active.
- 2007 Incorrect Registration Code
- 2008 Page Range and Page Selection must not both be selected.
- 2009 PreserveCanvas and PreserveCanvasCentered must not both be selected.

-3001-3009 Unspecified error. Contact support.

Distribution / Deployment

When deploying the application, ensure that the following files are included:

1. TiffDLL90.dll
2. Tifftek32.dll

Do NOT distribute these User Guides.

Do not disclose Registration Codes.

Advise the end user that Microsoft .NET Framework 2.0 or higher is required.

Technical Support

For technical support contact information, please go to www.informatik.com

License, Warranty, Disclaimer

Please read the terms carefully before installing and using the software, as such conduct will indicate your acceptance of all of the terms of this license agreement. If you do not agree with the terms, the software cannot be licensed to you and you must un-install and return the software to Informatik Inc, or its supplier or distributor.

This License Agreement is a legal agreement between Informatik Inc. ("Licensor"), a Delaware Corporation, and you, the user ("Licensee"), and is effective the date Licensee installs the software.

This Agreement covers all materials associated with the TiffDLL software, including, without limitation, the downloadable software product, online documentation, and any additional supporting electronic files (herein, the "Software").

The evaluation version may be used for 30 days after installation. It is unlawful to use the software after the 30 day evaluation period without licensing the software and paying the license fees. If a license is not obtained before the expiration of the 30 day evaluation period, the Software must be un-installed and destroyed.

1. GRANT OF LICENSE

Licensor hereby grants to you, and you accept, a nonexclusive license to use the Software according to the following condition:

You may use the Software on one (1) computer for one (1) developer) owned, leased, or otherwise controlled by you for personal or business purposes, and only as authorized in this License Agreement. The Software may not be used on other computers, nor may it be used by, or transferred to, other computers over a network. You may distribute royalty-free your applications with the TiffDLL90 DLL and required files to run the application. You may not distribute the user guides and you must keep the registration codes confidential. The TiffDLL90 must be a minor part of your application and it must be a desktop application. You may not develop tools that can be used to develop or be integrated with other applications.

2. LICENSOR'S RIGHTS

Licensee acknowledges and agrees that the Software is proprietary to Licensor and protected under international copyright law. Licensee further acknowledges and agrees that all right, title, and interests in and to the Software, including associated intellectual property rights, are and shall remain with Licensor. The License Agreement does not convey to Licensee an interest in or to the Software, but only a limited right of use that may be revoked in accordance with the terms of this License Agreement.

3. OTHER RESTRICTIONS

This License Agreement strictly forbids distribution of the Software with Licensee's application. Distribution of the Software with Licensee's application requires separate authorization and the payment of license fees.

Licensee agrees to make no more than one (1) back-up copy of the Software. Licensee agrees not to assign, sublicense, transfer, pledge, lease, rent, or share the rights assigned under this License Agreement. Licensee agrees not to reverse assemble, reverse compile, or otherwise translate the Software.

4. TERM

This License Agreement is effective when Licensee installs the Software and shall terminate only if the terms of this License Agreement are broken. Licensee agrees to destroy the Software upon termination of this License Agreement.

5. NO WARRANTY; LIMITATION OF LIABILITY

LICENSEE ACKNOWLEDGES THAT THE PROGRAM IS PROVIDED ON AN "AS IS" BASIS WITHOUT WARRANTY OF ANY KIND. LICENSOR MAKES NO REPRESENTATIONS OR WARRANTIES REGARDING THE USE OR PERFORMANCE OF THE SOFTWARE. LICENSOR incl. DEVELOPER, COPYRIGHTHOLDER, DISTRIBUTOR) EXPRESSLY DISCLAIMS THE WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. LICENSOR SHALL HAVE NO LIABILITY TO LICENSEE OR ANY THIRD PARTY FOR ANY LOSS OR DAMAGE CAUSED, DIRECTLY OR INDIRECTLY, BY THE SOFTWARE, INCLUDING, BUT NOT LIMITED TO, ANY INTERRUPTION OF SERVICES, LOSS OF BUSINESS, LOSS OF DATA OR SPECIAL, CONSEQUENTIAL OR INCIDENTAL DAMAGES.

6. GOVERNING LAW

This License Agreement shall be construed and governed in accordance with the laws of Pennsylvania.

7. SEVERABILITY

Should any court of competent jurisdiction declare any term of this License Agreement void or unenforceable, such declaration will have no effect on the remaining terms hereof.

8. NO WAIVER

The failure of either party to enforce any rights granted hereunder or to take action against the other party in the event of any breach hereunder shall not be deemed a waiver by that party as to subsequent enforcement of rights or subsequent actions in the event of future breaches.

Copyright and Trademarks

Copyright 2000-2009 Informatik Inc and J Buchmann. All Rights Reserved
TiffDLL is a Trademark of Inforamtik Inc.