

# TIFFmaker50

ActiveX DLL

**Informatik Inc.**

## User Guide

July 8, 2004

Print out this manual for easy reference.

Please also check the Frequently-Asked-Questions in [www.Informatik.com/faq](http://www.Informatik.com/faq).

## Table of Contents

Installation .....	3
Important Message .....	3
Trial Version .....	3
Use of Tiffmaker Command-line Driven Executable .....	3
User-Interactive Version of Tiffmaker .....	4
Use of Tiffmaker ActiveX DLL in your Application.....	4
Parameter Syntax .....	6
Parameters.....	7
Wrapping of Text.....	13
Error Codes.....	14
License Fees.....	15
Technical Support.....	15
Distribution to End-Users.....	15
Copyright.....	16
License Agreement.....	16
Appendix: .....	18
Object Method:.....	18
Index.....	20

## **Installation**

Click on the setup.exe or setupex.exe file and follow the prompts.

The VB version of Tiffmaker requires the VB Runtime file msvbvm60.dll. The standard download file includes then VB runtime file. If the file is missing please contact Informatik Inc.

## **Important Message**

The Tiffmaker program is very powerful and versatile, but if used incorrectly it can overwrite your files. Test your application fully before you use Tiffmaker in a production environment. Make sure that all files that your change (overwrite or append to) are fully backed up. Tiffmaker is supplied AS IS and the developer, copyright holder, supplier assume no responsibility. Please also read the Licensing Agreement section below. Do not use the program if you do not agree with the licensing terms and the disclaimers.

The Tiffmaker is intended to create TIFF files (and other graphics formats) from ASCII text files. The program is not suitable for form filling (many fields with different font characteristics). For **form filling**, you should use the TiffDLL50 ActiveX DLL program ([www.Informatik.com/tiffdll.html](http://www.Informatik.com/tiffdll.html)).

## **Trial Version**

If you are using the Trial Version, please note that all output images are marked 'TRIAL VERSION Tiffmaker'.

## **Use of Tiffmaker Command-line Driven Executable**

(Ignore this section if you are using the ActiveX DLL version)

An command-line driven executable version of Tiffmaker is also available (TIFFMK\_CMD.EXE). This executable version is run with a command line (not as an ActiveX DLL object), The parameter for the executable command string is the same as outlined below for the ActiveX DLL. If you run the executable from a DOS batch file make sure that you add the DOS keyword start with the /w flag, to ensure that each batch line is executed and completed before the next line is started.

Example of a simple one-line command line:

```
C:\ tiffmk_cmd.exe in=c:\filex.txt;out=c:\file1.tif;canvas=.
```

Example of a batch file:

```
Start /w C:\tiffmk_cmd.exe in=c:\filex.txt;out=c:\file1.tif;. . .
Start /w C \tiffmk_cmd.exe in=c:\filey.txt;out=c:\file2.tif;. . .
Start /w C:\tiffmk_cmd.exe in=c:\filez.txt;out=c:\file2.tif;. . .
```

These examples are over-simplified. Of course you need to use the correct folders and probably add additional parameter elements.

If you need to process all the files in a **directory and its sub-folders**, you can build a batch file with a special Batchfile program, downloadable from [www.informatik.com.files.html](http://www.informatik.com.files.html). The Batchfile program will build a batch file consisting of one command line per file, according to a template that you specify, using fixed text and various placeholders. The Batchfile program can also create empty folders to receive output files specified by the batch file. The Batchfile program is free for licensed users of Informatik software.

## **User-Interactive Version of Tiffmaker**

A user-interactive version of Tiffmaker is also available (TIFFMAKER50.EXE). If you cannot find it in the Tiffmaker50 folder, contact Informatik Inc.

For a quick test of the general Tiffmaker capabilities, it is suggested that you run the user-interactive program.

The user-interactive program has its own help file, accessible from the program's menu.

## **Use of Tiffmaker ActiveX DLL in your Application**

**Tiffmaker is not a Control**, it is an **ActiveX DLL** (apartment-threaded). Please consult your programming language's user and reference manual for instructions on how to integrate the ActiveX DLL into your application's code.

For Visual Basic applications, proceed as follows:

From the Projects (or Tools) menu, choose **References** and mark the Tiffmaker50 checkbox, then click on OK.

If the list of References shows MISSING:Tiffmaker50, un-check the item, click on OK, choose References again from the Projects menu, and you will find the correct Tiffmaker50 reference.

In your application, insert one of the sample codes shown below (VB code). The download file also includes a SAMPLE.VBP project file. With the sample\_test.exe utility you can test and experiment with the parameter strings without creating a project.

There are two methods of using the ActiveX DLL:

The **Parameter Method**: All specifications are embedded into a single parameter string in accordance with a strict syntax. The function is then called with the parameter string as the argument. This is the recommended method; it is easier to implement, and the descriptions of the parameters shown below use that method.

The **Object Method**: The individual properties of the object are set before the function is called. The Object method is covered in detail as an appendix at the end of this manual

### Parameter Method:

#### Sample Code for VB (using Parameter Method):

*(For other programming languages, please see sample code in [www.Informatik.com/files.html](http://www.Informatik.com/files.html))*

```
`(General) Declaration
Private obj As Tiffmaker50vic.ClsTiffMk50

Private Sub Form_Load()
    Set obj = New Tiffmaker50vic.ClsTiffMk50
End Sub

Private Sub Command1_Click()
    Dim result As Long
    Dim Param As String
    param = "in=c:\mydir\txtfile.txt;
           out=c:\dir2\somefile.tif;
           canvas=8/11/300;
           format=tif/14"
    result = obj.RunTiffMaker (param)
End Sub

Private Sub Form_Unload(Cancel As Integer)
    Set obj = Nothing
End Sub
```

Because the functions have is a very large number of parameter elements the Tiffmaker deviates somewhat from the general ActiveX DLL format. You only enter ONE parameter string. The single parameter string includes all the specifications (properties). For the syntax of the parameter string please see the sections below.

Upon successful conversion the Tiffmaker function returns a positive number, representing the number of pages processed (or the last vertical position of inserted text, see 'Offset=' section below). If the return value is negative it represents the error code (see 'Error Codes' section below)

## Parameter Syntax

(Parameter Method)

The parameter string (arguments) consists of a number of elements, such as source file, output file, graphics format, etc. Each element is separated by a semi-colon. In turn, most parameter elements are divided into sub-components, separated by slash (/).

The Tiffmaker function can handle several conversions at the same time; for example you can change the size of the image, change the resolution and change the graphics format with one command. Some conversions cannot be run at the same time and you need to run two functions in sequence.

Example of a simple parameter string:

```
In=c:\dir1\filex.txt;out=c:\dir2\somefile.tif;format=tif/14;
```

This sample parameter string specifies the following: File c:\dir1\filex.tif will be opened and saved as file c:\dir2\filez.tif, as a TIFF format 14. The text 'Sample' will be annotated at the top, right-adjusted, opaque. For now, don't worry about the codes; everything is explained below. Just note how the parameter elements are separated by semi-colons and that many of the parameter elements have sub-items separated by a slash (/). Note also, that each parameter elements has a name tag followed by an equal sign (=). There must not be a space between the name tag and the equal sign. 'In=' is correct, but 'in =' is incorrect.

To summarize:

- The parameter string has many parameter elements.
- The order of the parameter elements is irrelevant.
- Most parameters are optional. Only the parameter elements that are relevant need to be included, but you can include blank parameter elements.
- Each parameter elements has a name tag followed by an equal sign (=).
- There must never be a space between the name tag and equal sign, otherwise spaces are irrelevant.
- The strings are not case sensitive. In= is the same as IN= or in=.
- Many parameter elements have sub-items, separated by a slash (/). All leading sub-items must be entered, but lagging items may be omitted. Blank sub-items must also be separated by a slash (/).
- Text strings (for example in text=) should not be enclosed in quotes.
- Semi-colons always separate the parameter strings; so never use a semi-colon inside a text string (for example in text=).

For **debugging**, add a test=1; to the parameter string, like:

```
test=1; in=c:\file.txt;out=c:\dir2\file2.tif;format=tif/14;
```

To retrieve the **software license number**, run the function in test mode, like:

```
test=1
```

The input text file must be text-wrapped. If the text is not wrapped, you need to run the **Text Wrapping** function first. See Text Wrapping Function section below.

A **multi-page** text file is converted into a multi-page TIFF file. Only TIFF files support multi-page files. If you need to convert a multi-page text file into non-Tiff files (for example, BMP, JPEG, PNG) you must include the [ +nnn ] numeric placeholder in the output file name.

**Return value:** Upon successful conversion the Tiffmaker function returns a positive number, representing the number of pages processed. If the return value is negative it represents the error code (see below).

## **Parameters**

If you use the Parameter Method, each parameter element must be preceded with the label followed by the equal sign (no space between the label and the equal sign).

If you use the Object Method, do not use the label and the equal sign; enter only the value string.

**test=**

Example:

```
test=1
```

For **debugging**, use the 'test=' parameter element, like:

```
test=1; in=c:\file1.txt;out=c:\dir2\file2.tif;format=tif/14
```

The function will display the parsed values of the parameter. This will help you to diagnose any problems caused by an incorrectly formulated parameter string.

The test= parameter command also displays the **software license code**. The license code is required if you need technical support.

## **in=**

Example:

```
in=c:\somedir\somefile.txt;
```

This parameter specifies the source document (ASCII text file). The entry must be a fully-qualified path name of an existing file. Do not use a semi-colon character in the file name.

The text file must be text-wrapped. If the text is not wrapped, you need to run the Text Wrapping function first. See Text Wrapping Function section below.

## **out=**

Examples:

```
out=c:\somedir\somefile.tif;
```

```
out=c:\somedir\file[+001];
```

This parameter specifies the output file.

To save a file, the entry must be a fully-qualified path name and the folder (directory) must exist. You must have read/write permission for the specified folder. Do not use a semi-colon character in the file name.

To create serialized output files (single-page files from a multi-page input file, you must add the [ +nnn ] numeric placeholder. The nnn in the numeric placeholder will be converted to auto-incrementing numbers. For example myfile[+001].bmp would be converted to file names myfile001.bmp, myfile002.bmp, myfile003.bmp, etc.

## **format=**

Examples:

```
format=bmp;
```

```
format=tif/133;
```

The Format parameter string is optional. If not specified, the extension name of the output file will determine the format type. If the extension name refers to an unknown or unsupported format, then the file will be saved in TIFF format (default is TIFF Group 4).

The Format parameter consists of two sub-items: the format type and the compression type.

The available format strings are:

Format=tif/0	(uncompressed TIFF, multiple strips)
Format=tif/1	(TIFF/LZW, multiple strips)
Format=tif/2	(TIFF Packbits, multiple strips)
Format=tif/3	(TIFF 3, multiple strips)
Format=tif/4	(TIFF 4, multiple strips)
Format=tif/5	(TIFF CCITT 3, multiple strips)
Format=tif/10	(uncompressed TIFF, single strip)
Format=tif/11	(TIFF/LZW, single strip)
Format=tif/12	(TIFF Packbits, single strip)
Format=tif/13	(TIFF 3, single strip)
Format=tif/14	(TIFF 4, single strip)
Format=tif/15	(TIFF CCITT 3 Fax, single strip)
Format=tif/23	(TIFF 3, FillOrder=2)
Format=tif/24	(TIFF 4, FillOrder=2)
Format=tif/25	(TIFF CCITT 3, FillOrder=2)
Format=tif/33	(TIFF 3, single strip, FillOrder=2)
Format=tif/34	(TIFF 4, single strip FillOrder=2)
Format=tif/35	(TIFF CCITT 3, single strip, FillOrder=2)
Format=tif/135	(TIFF Class F, Dialogic)

Format=bmp/0	Windows Bitmap, uncompressed
Format=png/0	PNG format
Format=pcx/0	PCX format
Format=jpg/75	JPEG, 75% quality **
Format=gif/0	GIF format

\*\* Other quality factors are available (1-99)

### **save=**

Example:

```
save=1 ;
```

This parameter specifies how the output files are saved if the file name already exists.

The parameter values are:

- 0 = Overwrite without warning
- 1 = Append as additional page
- 2 = File already exists; file not saved; return code -9000

Generally, you should only append to TIFF files previously created by this software. If you append to TIFF files created by other software, carefully review the resulting TIFF. In any case make sure that you have made a backup of such TIFF files.

### **canvas=**

Examples:

```
canvas=8/11/300;  
canvas=c:\somedir\sample.bmp;  
canvas=c:\somedir\sample.bmp/1;
```

The canvas parameter has two personalities. It either specifies the dimension (width and height in inches) and the resolution of the output graphics file. Alternatively, it specifies the existing graphics file that you wish to use as a backdrop for the output graphics file (for example if you need a logo on the output file). If the canvas is not specified, it will be defaulted to 8/11/300 (8x11 inches, resolution of 300, a size that fits both A4 size and standard letter size).

### Dimensions and Resolution

If you use the canvas= parameter to specify the dimensions and resolution of the output graphics file you must specify three (3) values, separated by a slash (/):

1. Width in inches (1 inch = 2.5 centimeters)
2. Height in inches
3. Resolution (default is 300)

### Graphics (Canvas) File

If you use the canvas= parameter to specify the path to an existing bitmap file that you wish to use as a backdrop, the following requirements must be met:

- The canvas file must be a TIFF file.
- The canvas file should be monochrome (black and white) if possible, but 6-bit color (256 colors) is also acceptable (treatment depends on the specified output format).
- The full path name of the canvas file must be specified. Do not use a semi-colon character in the file name.
- The dimensions (width, height, resolution) of the canvas file must be exactly what you wish the output file to be.

- The canvas file must have equal horizontal/vertical resolutions.

It is possible to use the canvas file for **only the first page** of a multi-page file. For this option please add a value of 1 in the second parameter element, for example  
`canvas=c:\somedir\canvasfile.tif/1;`

For color output you must use a color file (canvas); maximum color depth is 8-bit (256 color palette)

## **font=**

Examples:

```
font=12b/Courier New;  
font=12b/Courier New/0/75;  
font=12bui/Arial
```

This parameter specifies the font characteristics. Up to four (4) items can be specified, in the correct order, separated by a slash (/):

1. Font size (size 12 is the default)
2. Font name (Arial font is the default)
3. Transparency (default is set to transparent)
4. Character width

If the font size is suffixed with a character 'b' (as in the example above) the text will be printed in bold. If the suffix includes the letter 'i' the text will be rendered italic; if the suffix includes the letter 'u', the text is underlined. The suffix can be a single character or a combination of characters. You can also specify Courier New font with a character 'c' and Times Roman font with a character 'r', for example `font=12bc`, or `font=12r` (this is a practical alternative to entering the font name in the second parameter element).

If not specified, the text will be rendered 'transparent', i.e. any backdrop graphics will be visible between the printed characters. To render the text 'opaque' set the Transparency value to 1.

The character width is rarely used but it allows you to quash the text. The implied standard value (non-specified) is 100. To stretch the text, use a value higher than 100; to compress the text, use a value less than 100. For example a value of 50 renders the text at 50% of the standard width. The character width setting is applicable only to scalable fonts (like Arial) and is ignored for fixed fonts (like Courier New)

If the text contains tabs and columns, you should use Courier New font (non-scaling font).

Advanced use: Each function call uses one font specification; to create different fonts on one page (for example using a different font for a paragraph), you need to call the function for each font, using the appropriate offset values. See 'Offset=' section below.

Font names are case-sensitive.

### **offset=**

Examples:

```
offset=-1/0.75;  
offset=-1/0.75/1;
```

This parameter specifies the offsets (borders) in inches. Two (2) values are required, separated by a slash (/). A third parameter element is optional, see paragraph below:

1. Left border in inches
2. Top border in inches

One inch is equivalent to 2.5 centimeters. The default for both is 0.50 inches. The specified values will be applied to all the pages.

If you enter a value of 1 in the third parameter element, the function will return the vertical position of inserted text (in inches). This is useful only if you run the function repeatedly for the same output document and need to know the vertical position for the next inserted paragraph. For example, if you create a document with paragraphs rendered in different fonts, you must call the function for each paragraph, specifying the font and the offsets.

### **lines=**

Example:

```
lines=25;
```

Unless specified otherwise and if there is no recognized page break, each page will be filled to capacity. You can specify a maximum number of lines for each page with the `lines=` parameter.

The `lines=` specification is subservient to any page breaks embedded in the source document. Page breaks can also be specified with the `break=` parameter (see below).

## **break=**

Examples:

```
break=<pagebreak> ;  
break=...newspage ;
```

This parameter is used to specify a custom page break. You can use any character or string of character to mark as a forced page break. In your source document that same string must exist on a separate line at the place where a page break should occur. The string is case-sensitive. If no page break is specified, Tiffmaker honors the standard page break character (ASCII 12); other wise it fills up the page to capacity.

*Important: Ensure that there is no Page-Break marker at the very end of the file.*

## **space=**

Examples:

```
space=2 ;  
space=0.75 ;
```

This parameter is used to specify the custom spacing between lines. Each unit represents one standard space. For example, an entry of 2 would add one extra space; and entry of 3 would add two extra spaces; an entry of 0.6 would create a smaller space.

## **Wrapping of Text**

The ASCII text files that you process with the Tiffmaker must already be wrapped, that is the text lines must be of correct length to fit the page width (with carriage return and line feed characters). Informatik Inc. can provide a command-line driven utility to convert your text file into wrapped text. For availability and fees please contact Informatik Inc.

## Error Codes

Upon successful conversion the Tiffmaker function returns a positive number, representing the number of pages processed. If the return value is negative it represents the error code (see below). . If you run in Test mode (test=1) the conversion is not executed and the function returns a zero (0).

### Common Error Codes:

-9000	File already exists, no overwrite
-9100	Cannot open source file (wrong format or does not exist)
-9101	Cannot save file
-9102	Canvas higher than 8-bit color depth (limit is 8-bit)
-9103	Width or height not correctly specified (must be specified in pixels)
-9005	Unable to write text
-9111	Source file is not specified
-9112	Destination file is not specified
-9998	Test mode; conversion not executed
-9999	Parameter string is missing

### Tifftek32 Library Error Codes:

00	No error
01	Range error
02	Digitizer board not detected
03	Disk full, file not written
04	Filename not found
05	Variable out of range
06	Unreadable TIFF format
08	TIFF bits per sample not supported
09	Unreadable compression scheme
10	Cannot create file (Make sure the folder exists)
11	Unknown file format
12	Image is compressed DIB
14	Insufficient memory for function
16	Unreadable PCX format
17	Unreadable GIF format
18	Print error
19	Scanner error
25	Unreadable TGA format
26	Bits per pixel not supported
27	Unreadable BMP format
33	No data from device
34	Function timed out
40	Could not lock memory (invalid handle or memory discarded)
41	Print function already executing
42	Invalid image buffer address
43	Unreadable JPEG format
44	Image is too complex for operation

45	Paper could not be unloaded
46	ADF lid was opened
47	ADF bin is empty
48	ADF is not connected
49	ADF is connected but not ready
50	Function not available due to missing module
52	Pointer does not point to readable or writable memory
53	LZW compression/decompression not enabled (contact Informatik)
54	Could not create TWAIN parent window
55	Could not open TWAIN Source Manager
56	Could not open TWAIN Data Source
57	TWAIN image acquisition error
58	None of the elements in two lists were equal
59	Data type mismatch
60	User canceled scan
61	TWAIN function is busy
62	File contains invalid data
63	Unreadable PNG format
64	PNG compressor error
65	No ACK from device
66	TWAIN ADF is empty
67	Stop scanning images
68	Handle not valid
69	TIFF file is in Motorola byte order

## **License Fees**

Please see [www.Informatik.com/tiffdll.html](http://www.Informatik.com/tiffdll.html) or contact Informatik Inc. For contact information please visit [www.Informatik.com/company.html](http://www.Informatik.com/company.html)

## **Technical Support**

Please see [www.Informatik.com/support.html](http://www.Informatik.com/support.html)

## **Distribution to End-Users**

The following files must be included with your distribution:

- Tiffmaker50.DLL (must be registered in the Windows Registry)
- Tifftek32.dll
- VB6 runtime

## **Copyright**

Copyright 2001-2002 Joseph Buchmann. All rights reserved.

## **License Agreement**

Please read this License Agreement carefully before you install and use the software. By installing and using the software you agree with the terms and conditions of this License Agreement. If you do not agree with the terms and conditions you must uninstall the software.

This License Agreement ("Agreement") is a legal agreement between Informatik Inc., Devon Pennsylvania, USA ("Licensor") and you, the user ("Licensee"), and becomes effective on the day the Licensee installs the software. This Agreement covers all materials associated with this software, including, without limitation, the downloadable relating files, printed and online documentation, and any additional supporting files and programs (herein, the "Software").

### **1. GRANT OF LICENSE**

Licensor hereby grants to you, and you accept, a nonexclusive license to use the Software on a specified number of computers/workstations 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. Unless the Licensee purchased a server license, the Software may not be installed on a server, or on a PC/workstation that acts as a server. Hosting a batch service on a server, PC or workstation requires a server version. A site license permits the installation of the Software on an unlimited number of PC/workstations at one site for the enterprise.

ActiveX DLLs may become part of the licensee's own application, but the ActiveX DLL must represent a relative minor part of that application. The applications so developed can be distributed to the number of end users equal to the number of runtime licenses purchased under this license agreement. Under no circumstances must the ActiveX DLL be used to create another toolkit, control, COM object or library, unless strictly used internally by the licensee, and not sold to third parties. A separate license for the ActiveX DLL is required for each developer.

### **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**

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 SOFTWARE IS PROVIDED ON AN "AS IS" BASIS WITHOUT WARRANTY OF ANY KIND. LICENSOR, DISTRIBUTORS OR AGENTS MAKE NO REPRESENTATIONS OR WARRANTIES REGARDING THE USE OR PERFORMANCE OF THE SOFTWARE. LICENSOR DISTRIBUTORS and AGENTS EXPRESSLY DISCLAIM THE WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. LICENSOR, DISTRIBUTOR OR AGENT 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

The legal jurisdiction of this License Agreement shall be Pennsylvania, USA.

## Appendix: Object Method:

Instead of defining a parameter string you can use the properties of the declared object. In your application's code, if you declared TiffDLL50.ClsTiffDLL as an object (named 'obj' in our example), simply type the name of the object (obj) followed by a period (.). The programming interface will then display a list of all available properties. Pick the relevant property and define it, as in the following example:

```
obj.InputFile = "c:\somedir\somefile.tif"  
obj.OutputFile = "c:\tiffdlltest\filexyz.tif"  
obj.OutputFormat = "tif,14"  
etc.
```

The syntax of the object's property string is identical to the string that you would use in the parameter (except for the label; don't use the label with the equal sign). See Parameters section below.

Using the Object method, you run the function with a blank parameter, like

```
result = obj.RunTiffMaker ("")
```

### Sample Code for VB (using Object Method):

*(For other programming languages, please see sample code in [www.Informatik.com/files.html](http://www.Informatik.com/files.html))*

```
`(General) Declaration  
Private obj As TiffMaker50vic.ClsTiffMk50  
  
Private Sub Form_Load()  
    Set obj = New TiffMaker50vic.ClsTiffMk50  
End Sub  
  
Private Sub Command1_Click()  
    Dim result As Long  
    obj.InputFile = "c:\somedir\somefile.txt"  
    obj.OutputFile = "c:\dir1\filexyz.tif"  
    obj.OutputFormat = "tif/14"  
    result = obj.RunTiffMaker ("")  
End Sub  
  
Private Sub Form_Unload(Cancel As Integer)  
    Set obj = Nothing  
End Sub
```

The properties of the object are (all string data types):

**File\_In** (equivalent to **in=** in Parameter Method)  
**File\_Out** (equivalent to **out=** in Parameter Method)  
**Format\_Out** (equivalent to **format=** in Parameter Method)  
**SaveMethod** (equivalent to **save=** in Parameter Method)

**File\_Canvas** (equivalent to **canvas=** in Parameter Method)  
**Text\_Font** (equivalent to **font=** in Parameter Method)  
**Page\_Lines** (equivalent to **lines=** in Parameter Method)  
**Text\_Space** (equivalent to **space=** in Parameter Method)  
**Page\_Break** (equivalent to **break=** in Parameter Method)  
**Text\_Offset** (equivalent to **offset=** in Parameter Method)  
**TestMode** (equivalent to **test=** in Parameter Method)

## **Index**

ActiveX DLL, 4  
annotations, 11, 12, 13  
append pages, 9  
Background image, 10  
Batch file, 4  
Bates stamping, 11, 12, 13  
BMP format, 9  
bold text text, 11  
Border, 12  
Canvas, 10  
canvas size, 10  
Character width, 11  
color, 11  
Command-line version, 3  
Copyright, 16  
cover page, 11  
debugging, 6  
Dimensions, 10  
dimensions, paper, 10  
Distribution, 15  
distribution, files required, 15  
DLL, ActiveX, 4  
Fillorder, TIFF, 9  
first page only, 11  
Font, 11  
footers, 11, 12, 13  
format, 8  
Format, 8  
GIF format, 9  
graphics format, 8  
Graphics Format, 8  
input file, 8  
installation, 3  
Installation, 3  
italic text, 11  
JPEG format, 9  
License Agreement, 16  
license fees, 15  
license number.  
Lines, number of lines, 12  
LZW compression, 9  
multi-page, 9  
multi-page files, 7, 8  
multiple strips, 9  
Object Method, 18  
Offset, 12  
opaque text, 11  
output file, 8  
Output file, 8  
Packbits format, 9  
Page break, 13  
paper size, 10  
Parameter Method, 5  
Parameter Syntax, 6  
Parameters, 7  
PCX format, 9  
PNG format, 9  
Resolution, 10  
rror codes, 14  
Save As file, 8  
serialized pages, 8  
Single strip, TIFF, 9  
single strips, 9  
software license, 7  
source document, 8  
source file, 8  
Spacing, between lines, 13  
Support, 15  
syntax, 6  
Targa format, 9  
testing, 7  
text annotations, 11, 12, 13  
text file, 8  
text wrapping, 13  
Text wrapping, 7  
TIFF format, 9  
Transparency, 11  
transparent text, 11  
underlined text, 11  
User-interactive version, 4  
Visual Basic, 4  
wrapping, text, 13  
Wrapping, text, 7