

# TMSSequoia ScanFix ActiveX Control

---

User's Guide  
ImageBASIC 3.1

*IMAGE*  *BASIC*

Diamond Head Software, Inc.  
1217 Digital Drive Ste. 125  
Richardson, Texas 75081  
(972) 479-9205

## Copyright Notices

Under the copyright laws, neither the documentation nor the software may be copied, photocopied, reproduced, translated, or reduced to any electronic medium or machine-readable form, in whole or in part, without the prior written consent of Diamond Head Software, Inc., except in the manner described in the documentation.

This software product contains proprietary software components developed by a number of different software companies, referred herein as "Third Party Licensors". This documentation and the software that you purchased are protected by one or more of the following copyright notices:

Portions of this product, © 1994, 1995, 1996, 1997 Diamond Head Software, Inc. All rights reserved.  
Portions of this product, © 1994, 1995, 1996 TMSSequoia, Inc. All rights reserved.

Company and product names mentioned in this documentation are trademarks or registered trademarks of their respective companies. Windows is a trademark and Microsoft is a registered trademark of Microsoft Corporation.

DIAMOND HEAD SOFTWARE, INC. AND ITS THIRD PARTY LICENSORS MAKE NO WARRANTIES, EXPRESS OR IMPLIED, INCLUDING WITHOUT LIMITATION THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, REGARDING THE SOFTWARE. DIAMOND HEAD SOFTWARE, INC. AND ITS THIRD PARTY LICENSORS DO NOT WARRANT, GUARANTEE OR MAKE ANY REPRESENTATIONS REGARDING THE USE OR THE RESULTS OF THE USE OF THE SOFTWARE IN TERMS OF ITS CORRECTNESS, ACCURACY, RELIABILITY, CURRENTNESS OR OTHERWISE. THE ENTIRE RISK AS TO THE RESULTS AND PERFORMANCE OF THE SOFTWARE IS ASSUMED BY YOU. THE EXCLUSION OF IMPLIED WARRANTIES IS NOT PERMITTED BY SOME JURISDICTIONS. THE ABOVE EXCLUSION MAY NOT APPLY TO YOU.

IN NO EVENT WILL DIAMOND HEAD SOFTWARE INC. OR ITS THIRD PARTY LICENSORS AND/OR THEIR DIRECTORS, OFFICERS, EMPLOYEES OR AGENTS BE LIABLE TO YOU FOR ANY CONSEQUENTIAL, INCIDENTAL OR INDIRECT DAMAGES (INCLUDING DAMAGES FOR LOSS OF BUSINESS PROFITS, BUSINESS INTERRUPTION, LOSS OF BUSINESS INFORMATION, AND THE LIKE) ARISING OUT OF THE USE OR INABILITY TO USE THE SOFTWARE EVEN IF DIAMOND HEAD SOFTWARE INC. OR ITS THIRD PARTY LICENSORS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. BECAUSE SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF LIABILITY FOR CONSEQUENTIAL OR INCIDENTAL DAMAGES, THE ABOVE LIMITATIONS MAY NOT APPLY TO YOU. Diamond Head Software Inc.'s and its Third Party Licensors' liability to you for actual damages from any cause whatsoever, and regardless of the form of the action (whether in contract, tort (including negligence), product liability or otherwise), will be limited to \$50.

# Contents

Chapter 1 : Introduction	1
Linking of ImageBASIC Controls.....	1
Licensing Verification.....	2
Programming Considerations.....	2
Image Processing with the ScanFix Control .....	3
Standard Image Processing Options.....	5
Premium Image Processing Options.....	6
Chapter 2 : Image Processing Options	7
ScanFix Image Processing Properties.....	7
Enhancing an Image .....	7
Select Source and Destination .....	7
Activate Image Enhancement Options .....	8
Begin Image Enhancement .....	9
Chapter 3: Image Enhancement Options	10
Enabling Image Enhancement.....	10
Speck Removal .....	11
Speck Removal Properties .....	11
Dot Shading Removal .....	13
Dot Shading Removal Properties .....	13
Inverted Text Correction.....	16
Inverted Text Correction Properties.....	16
Image Registration and Positioning.....	19
Horizontal Registration Properties.....	19
Vertical Registration.....	21
Line Detection and Removal .....	24
Horizontal Line Detection and Removal Properties.....	24
Vertical Line Detection and Removal.....	27
Skew Detection and Correction.....	29
Deskewing Properties.....	30
SubImage Selection .....	32
SubImage Selection Properties .....	32
Removing Runs of Periods .....	35
Premium Enhancement Options .....	36
Character Completion .....	37
Character Smoothing .....	38
Character Smoothing Properties .....	38
Intelligent Cropping, or Removing Borders .....	40
Intelligent Cropping Properties.....	40

Orientation and Rotation.....	41
Orientation and Rotation Properties.....	41
The OptionString Property.....	43
OptionString Syntax.....	43
Processing Mode Options .....	44
Image Cropping Options .....	44
AutoRotate Specific Options .....	46
Debugging Options.....	47
Deskew Control Options.....	48
Dot Shading Options.....	48
Fax Management Options.....	49
File Reading Options.....	50
File Writing Options.....	50
Inverse Text Management Options.....	53
Other String Options.....	53
Registration Control Options .....	53
SubImage Control Options.....	54
 Chapter 3: ScanFix Control Reference .....	 55
Properties, Methods and Events.....	55
 Index .....	 89

# Chapter 1 : Introduction

---

## Linking of ImageBASIC Controls

Virtually all ImageBASIC controls can accept image data from other ImageBASIC controls. The process of designating where each ImageBASIC component gets its image data is referred to as linking of controls. With the exception of those controls that can directly access files or scanners, all ImageBASIC controls must be linked to another ImageBASIC control to receive any image data.

Linking of controls is the primary method of moving an image through a series of processing steps. For example, an image may be originally captured through the PixScan Control, passed to a TMS Display Control for operator verification, optionally routed through the ScanFix control for enhancement, then to a TMSFile control to be written to disk, and finally to a TextBridge Control for OCR processing to generate indexing information for that file.

### Creating the Link

All of the ImageBASIC controls that can accept and process image data from other ImageBASIC controls have a property named **ImageDataSource** to create the data link between controls, this property is set to the Name of the source ImageBASIC control. Any image that is received by the named control will also be sent to the linked control. For example, if the **ImageDataSource** property of a TMS Display control is set to the PixScan control, each time a new image is scanned, the display control will receive that image.

Linking each control requires only one line of code that can be executed at any time. The source of image data can be changed during program execution by naming another ImageBASIC control in the ImageDataSource property, as shown here:

```
ScanFix1.ImageDataSource = TMSDispl.Link
```

Each time the image data flowing from one ImageBASIC control changes, the receiving control's **ImageDataChanged** event is triggered. From this event, any procedure that is to be performed on each image can be started. For example, each time an OCR control **ImageDataChanged** event occurs, an OCR attempt could be started on the new image data.

# Licensing Verification

In order for any application using the ScanFix Control to function, it must be able to verify that the control is licensed. This verification usually takes the form of a search for a license in a database or on a hardware key. In all cases, however, the developer is given some control over when the control requests licensing verification.

# Programming Considerations

ScanFix works from disk to disk (reading an input file from the disk and depositing an output file on the disk) or from memory to memory. It also works from disk to memory and *vice versa*. Source and destination are selected with the **InputFrom** and **OutputTo** properties, either of which may be set to *0--Memory* or *1--File*.

The four possible combinations are as follows.

Image Source		Image Destination
File	to	File
File	to	Memory
Memory	to	File
Memory	to	Memory

It is best to have ScanFix process image files outside of the scanning processing loop, because the processing time for the ScanFix operation is significantly greater than scanning time, and in a large scale batch activity the capacities will not be balanced. A scanning operation that we know of uses multiple 486 machines running ScanFix on TIFF files in a directory to process the output of one high speed scanner.

Depending upon the size and complexity of the original image, ScanFix requires at least 8 megabytes of RAM and should be performed on a 486 or better machine. To Deskew, Remove Lines, Remove Dot Shading, and correct Inverse Type averages from 5 to 15 seconds on a 300 dpi image.

The ScanFix properties consist primarily of processing parameters for the ScanFix operation. Many of the properties are toggle switches, with True/False settings. Some of these, when set to True, activate other parameter properties which must be addressed.

---

# Image Processing with the ScanFix Control

Using the image processing options of the ScanFix Control is basically a three step process:

1. The input and output destination must be set.
2. Properties called processing parameters must be set to tell ScanFix exactly what to do with the image.
3. The **ScanFix** method is executed to start processing.

## 1. Setting Source and Destination

The first step is to set the source and destination. Both input and output can be either a file or memory. The choice is made by setting the **InputFrom** and **OutputTo** properties to *0--Memory* or *1--File*.

- If **InputFrom** is set to *0--Memory*, the **ImageDataSource** property must be set to an ImageBASIC control that will supply image data.
- If **InputFrom** is set to *1--File*, the **InputFileName** property must be set to a valid image file name.
- If **OutputTo** is set to *0--Memory*, the **ImageDataSource** property of some other ImageBASIC control must be set to specify the ScanFix control. For example, a TMS Display control may accept the image for operator verification.
- If **OutputTo** is set to *1--File*, the **OutputFileName** property must be set. Input and output filenames cannot be the same.

## 2. Setting Processing Parameters

The second step, setting processing parameters, requires that you set a number of property values to determine what ScanFix does during processing. Most of the ScanFix image processing options are enabled when an enabling property is set to True. Enabling an option activates other properties which set parameters to that utility.

For instance, if you are going to deskew an image (make the text horizontal on the page), the properties that need to be set are as follows:

```
' activate the deskewing capability
ScanFix1.DeskewActivate = True

' minimum length in pixels to check for skew;
' prevents attempts to identify skew over very
' short distances
ScanFix1.DeskewMinDetectLength = 300
```

```
' sets the maximum acceptable skew; 1 pixel skew
' over this length
ScanFix1.DeskewMaxAcceptableSkew = 150
```

Many of the processing options have parameter properties, but others need only be activated.

### 3. Start Processing

The third step in image enhancement is to execute the **ScanFix** method:

```
ScanFix1.ScanFix
```

The ScanFix control will begin to process the image specified in the **InputFrom** property using the Processing Options that have been activated. When it is finished, the processed image will be sent to the destination specified in the **OutputTo** property.

## Standard Image Processing Options

ScanFix offers an extremely wide variety of options, both as properties and as string functions. ScanFix may be used to process single images or multi-page image files. Following are the key processing options.

### ***Skew Detection and Correction***

ScanFix detects and corrects up to 20% skew in a text image. It does so by using text or lines as a guide, and does not require training or registration marks. The **DeskewActivate** property is a toggle which must be set to True (-1) to activate this faculty. The **DeskewMinDetectLength** and **DeskewMaxAcceptableSkew** properties are parameters and set the length of text to use for deskewing and the maximum deviation from horizontal that is acceptable, respectively. Both these properties are in units of pixels. The properties and their use are fully detailed in "Skew Detection and Correction" on page 29.

### ***Document Registration***

ScanFix can position text to within 3-4 pixels (about 1/100<sup>th</sup> of an inch at 300 DPI) on the page, providing a consistent left and upper margin for documents. It does so without the need for training or registration marks. It also offers the ability to register the document to vertical or horizontal margin lines. Two properties, **HorzRegActivate** and **VertRegActivate**, can be separately activated to perform registration. Each activates its own set of parameters. The differences between these options are detailed in "



Image Registration and Positioning" on page 18.

### ***Line Detection and Removal with Character Repair***

ScanFix can automatically detect and remove both horizontal and vertical lines, and is capable of differentiating between large text and small lines. When removing lines, ScanFix has a Character Repair feature which reconstructs characters intersected by removed lines. The **HorzLineActivate** and **VertLineActivate** properties must be set to True (-1) to activate this capability, and several parameter properties are also activated. All these properties are explained in "Line Detection and Removal" on page 24.

### ***Inverse Type Detection and Correction***

ScanFix can automatically detect zones of inverse (white-on-black) text and converts them into normal text. It will process inverse text zones of virtually any shape, including rectangles, circles, banners, etc. Inverted text will be corrected when the ScanFix() function is called when the **InverseActivate** property is set to True (-1). The associated parameter properties set the size and location of the expected inverted area. See "Inverted Text Correction" on page 16 for details.

### ***Isolated Speck Removal***

ScanFix detects specks and processes only those areas where the shading exists, such as shaded tables in the center of a page. It also cleans up random specks throughout the page. The properties **DespeckVertical**, **DespeckHorizontal**, and **DespeckIsolated** all accept pixel size values to search for and remove. Refer to "Speck Removal" on page 11 for implementation details.

### ***Dot Shading Detection and Removal***

ScanFix automatically detects dot-pattern shading and determines the size of the individual dots in shaded areas. The **DotShadingActivate** property can be set to True to activate several parameter properties for the automatic removal of shaded zones. Refer to "Dot Shading Removal" on page 13 for implementation details.

### ***Sub-Image Processing***

To separate out a section of the image, the **SubImageActivate** property must be set to True, which activates five parameter properties. These properties hold the pixel coordinates which define the sub-image, so it is very important that the image is properly aligned. The precise techniques for implementing subimage processing are found in "SubImage Selection" on page 32.

## Premium Image Processing Options

In addition to the standard features detailed above, three premium features are available in ScanFix. The three features are Character Smoothing, Automatic Orientation, and Intelligent Cropping.

### ***Character Smoothing and Advanced Line Removal***

Character smoothing is an optional feature which corrects character flaws, making the surfaces more regular and clear. This process enables OCR engines to recognize dot matrix printed text and other low resolution images. Advanced Line Removal will remove poor quality or degraded lines that standard line removal may miss. Refer to "Character Completion" on page 37 for details and options.

### ***Image Rotation and Orientation***

ScanFix can automatically rotate your scanned image before or after processing, and it can be set to orient documents to portrait or landscape, as desired, or to automatically position them right side up. This is very useful if you suspect that your input file is rotated or upside down and wish to automatically correct it. These properties are explained in "Orientation and Rotation" on page 41.

### ***Intelligent Cropping***

ScanFix can crop black or white borders from an image. The removal of black borders is accomplished by "bleaching" the border. White borders are fully removed, registering the image in the process. See "Intelligent Cropping, or Removing Borders" on page 40 for details.

# Chapter 2 : Image Processing Options

---

## ScanFix Image Processing Properties

Most ScanFix properties are processing parameters for the ScanFix operations. Many of them are toggle switches, with only True and False settings. Some of these switches will activate other properties which set parameters for their operation.

For example, the **InverseActivate** property enables the detection and correction of inverse text (white letters on a black background). If it is True, four other properties set parameters for its action: **InverseHeight**, **InverseWidth**, **InverseEdge**, and **InverseReport**. Setting any of the four properties will have no effect unless the **InverseActivate** property is True.

---

## Enhancing an Image

Using any of the image enhancement options is basically a three step process:

- 1) Specify the source and destination of the images
- 2) Activate select enhancement options and set parameters
- 3) Begin enhancement by triggering the ScanFix method.

### Select Source and Destination

The ScanFix control can read image data from either it Parent Image Display Window or directly from an image file. Likewise, the control can send its output data to either an Image Display Window or to a file. The four possible combinations are as follows:

Image Source		Image Destination
File	to	File
File	to	Display
Display	to	File
Display	to	Display

### Source and Destination Properties

The following properties must be set to specify this information:

InputFrom	<p>Specifies the image source. The options for this property are as follows:</p> <ul style="list-style-type: none"> <li>0      Memory (Default)</li> <li>1      File</li> </ul> <p>If <i>0--Memory</i> is selected, the <b>ImageDataSource</b> property must specify an ImageBASIC control which will supply the image to enhance.</p> <p>If <i>1--File</i> is selected, the <b>InputFileName</b> property must be set to a valid image file name.</p>
OutputTo	<p>Specifies the destination of the data output by the ScanFix control. The options for this property are as follows:</p> <ul style="list-style-type: none"> <li>0      Memory (Default)</li> <li>1      File</li> </ul> <p>If <i>0--Memory</i> is selected, the <b>ImageDataSource</b> of another ImageBASIC control must specify this ScanFix control so that the output image data is forwarded.</p> <p>If <i>1--File</i> is selected, the <b>OutputFileName</b> property must be set to a valid image file name.</p>
InputFileName	<p>Fully qualified path to an image file for processing. Only valid if the <b>InputFrom</b> property is set to <i>1--File</i>.</p>
OutputFileName	<p>Fully qualified path to the output file after processing. Only valid if the <b>OutputTo</b> property is set to <i>1--File</i>.</p>

#### Example -- Setting Source and Destination

```

SFix1.InputFrom = 1   ' File
ScanFix1.OutputTo = 1       ' File
ScanFix1.InputFileName = "c:\newscans\img_001.tif"
ScanFix1.OutputFileName = "c:\process\img_001p.tif"

```

## Activate Image Enhancement Options

Most image enhancement options are enabled by setting an activating property to True. For example, the **InverseActivate** property will detect inverse text (white letters on a black background) and correct it. If it is set to True, four other properties set parameters for its action: **InverseHeight**, **InverseWidth**, **InverseEdge**, and **InverseReport**. Setting any of the four properties will have no

effect unless the **InverseActivate** property is set to True. To set these parameters, you would need code similar to that below.

### **Example -- Setting Image Enhancement Options**

```
' inverse text will be detected and corrected
ScanFix1.InverseActivate = True

' the minimum height of inverse text detected
ScanFix1.InverseHeight = 100

' the minimum width of inverse text detected
ScanFix1.InverseWidth = 300

' distance from text to edge of inverse area
ScanFix1.InverseEdge = 20

' sends a report to the Message event
ScanFix1.InverseReport = True
```

## **Begin Image Enhancement**

After all the processing parameters have been set, execute the **ScanFix** method to begin processing:

```
ScanFix1.ScanFix
```

The image enhancement options that have been activated will be performed. "Chapter 3: Image Enhancement Options" on page 10 details all of the available options and the properties used to enable and set parameters for the enhancements.

# Chapter 3: Image Enhancement Options

---

## Enabling Image Enhancement

Image enhancement encompasses a number of different procedures that improve the readability of an image, both for a human operator and for OCR and ICR engines. A common byproduct of this improvement is a reduced storage size.

As a general rule, each image enhancement option is enabled through a single Boolean property and parameters are set for that option through one or more additional properties. A few enhancement options are both enabled and set parameters through a single integer or enumerated property.

The following table indicates the enabling property for each enhancement option. The parameter properties for each option are detailed in the following pages.

Image Enhancement Option	Enabling Property
Speck Removal	DespeckIsolated DespeckVertical DespeckHorizontal
Dot Shading Detection and Removal	DotShadingActivate
Inverted Text Correction	InverseActivate
Registration	HorzRegActivate VertRegActivate
Line Detection and Removal	HorzLineActivate VertLineActivate
Skew Detection and Correction	DeskewActivate
SubImage Processing	SubImageActivate
Dot Leader Detection and Removal	PeriodRemovalExpectedFrequency
Character Smoothing*	CompletionSmooth CompletionGrow CompletionErode
Automatic Orientation*	AutoRotatePortrait AutoRotateRevert AutoRotateUpright TurnAfter TurnBefore

Intelligent Crop*	CropBlack CropWhite
-------------------	------------------------

\* Premium enhancement options

# Speck Removal

When removing specks from an image, the ScanFix control can be instructed to find and remove either isolated spots of a defined size, or the control can identify and remove regions of spot shading. The removal of isolated specks is described in this section, and the removal of dot shaded areas in the section **Dot Shading Removal**" on page 13.

## Speck Removal Properties

The Visual Basic programmer must set four properties to control the removal of isolated spots:

DespeckHorizontal	All image objects less wide than the number of pixels specified in this property will be removed. For example, if <b>DespeckHorizontal</b> is set to 2, any vertical lines only 2 pixels wide will be removed.
DespeckIsolated	Specifies the maximum <i>width</i> and <i>height</i> of an isolated speck that will be removed. May be set to any integer value, indicating the width and height of the specks in pixels.
DespeckVertical	All image objects less tall than the number of pixels specified in this property will be removed. For example, if <b>DespeckVertical</b> is set to 2, any horizontal lines only 2 pixels wide will be removed.

Too large values in the above properties will remove text elements like the dots from i's. Measure the thinnest parts of characters to determine how aggressively to despeck. If characters tend to be thicker in one plane (such as this letter O, which is thicker in the Horizontal plane), then set a higher value in that plane.

To remove only small isolated specks, set only the **DespeckIsolated** property. The removal process for this procedure is faster than using **DespeckHorizontal** and **DespeckVertical**

### Example -- Speck Removal

The following settings will remove all objects in the image which are either one (1) pixel wide, or one (1) pixel high, or two (2) pixels in both dimension. In this

example, we are linking the ScanFix control to the file control. It will also receive and send information to this control:

```

' link the ScanFix control to a file control for
' input and to another file control for output
ScanFix1.ImageDataSource = TMSFile1.Link
TMSFile2.ImageDataSource = ScanFix1.Link

' set InputFrom and OutputTo to memory,
' since the file controls will load and save the
' image to disk
ScanFix1.InputFrom = 0           'ImageDataSource
ScanFix1.OutputTo = 0           'Memory

' set the input file name
TMSFile1.InputFileName = "c:\newfiles\tiff0001.tif"

' activate the auto detect despeck and activate
ScanFix1.DespeckActivate = True

' set the despeck values
ScanFix1.DespeckHorizontal = 1
ScanFix1.DespeckVertical = 1
ScanFix1.DespeckIsolated = 2

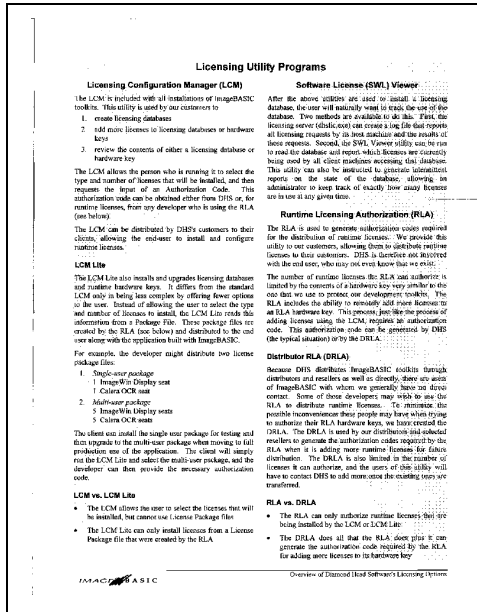
' begin processing
ScanFix1.ScanFix

' save the new image to file
TMSFile2.Save "c:\processed\img_001.tif"

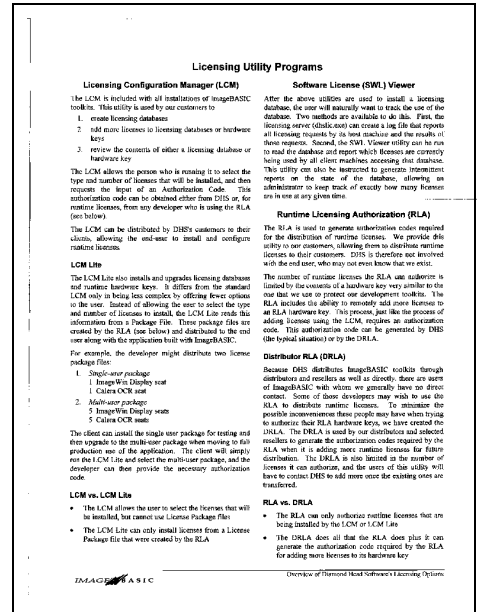
```

The following images show the improvement made by the removal of random specks from an image. The image on the left was the original image, and the image on the right was created by ScanFix applying the speck removal parameters shown in the sample immediately above.





Before speck removal; file size 40.5 KB



After speck removal; file size 37.8 KB

## Dot Shading Removal

The ScanFix control can automatically detect dot-pattern shading and determine the size of the individual dots in shaded areas. Dot shading removal erases both black-on-white shading and gray shading (which can also be removed by scanner/copier contrast settings).

The accuracy of OCR engines is greatly reduced in shaded text areas, and shaded text is often treated as graphics instead of text. ScanFix allows the OCR reading of dot-shaded text by automatically detecting zones of shading, determining the size of individual dots, and removing the dots.

The ScanFix control will also remove isolated specks in an image, also increasing OCR accuracy and decreasing image file size. The removal of isolated specks is detailed in the section 'Speck Removal' on page 11.

## Dot Shading Removal Properties

The removal of dot shaded areas must be activated by setting the **DotShadingActivate** property to True. The following properties then set parameters to the removal process:

DotShadingActivate	Must be set to True to activate the detection and removal of dot shaded areas.
DotShadingHeight	The minimum height of a dot shaded area that will be removed. The default is 50 (pixels), or 1/6th inch at 300 DPI.
DotShadingHorzAdjust	May be set to an integer value to increase the horizontal size (width) of dots that the control will remove. The number of pixels specified in this property will be added to the dot size that was automatically determined by the control, and all dots of this larger width will be removed.
DotShadingMaxSpeckSize	The ScanFix control automatically detects the size of the dots comprising a dot shaded region. The DotShadingMaxSpeckSize property sets a maximum dot size, either horizontally or vertically, the control will attempt to remove.
DotShadingProtect	Characters can be degraded during the dot shading removal process, particularly if the characters touch several dots. The DotShadingProtect property may be set to True to perform reconstruction of the degraded characters.
DotShadingReport	May be set to True to output a report in the Message event. This report shows the position of the dot shaded region in the image.
DotShadingVertAdjust	May be set to an integer value to increase the vertical size (height) of dots that the control will remove. The number of pixels specified in this property will be added to the dot size that was automatically determined by the control, and all dots of this larger height will be removed.
DotShadingWidth	The minimum width of a dot shaded area that will be removed. The default is 100 (pixels), or 1/3rd inch at 300 DPI.

### ***Example -- Dot Shading Removal***

For this example, we will link ScanFix to the display control and set the InputFrom and OutputTo to the display

```

' link the ScanFix control to one display control
' for input and another for output
ScanFix1.ImageDataSource = TMSDisp1.Link
TMSDisp2.ImageDataSource = ScanFix1.Link

```

```

        ' set the InputFrom and OutputTo
        ' properties to receive and give information from
        ' the display
ScanFix1.InputFrom = 0          ' ImageDataSource
ScanFix1.OutputTo = 0          ' Memory

        ' activate auto detection and removal of dot
        ' shaded regions
ScanFix1.DotShadingActivate = True

        ' find only regions at least 300 pixels high (1"
        ' at 300 DPI)
ScanFix1.DotShadingHeight = 300

        ' find only regions at least 600 pixels high (2"
        ' at 300 DPI)
ScanFix1.DotShadingWidth = 600

        ' remove dots 1 pixel wider than automatically detected
ScanFix1.DotShadingHorzAdjust = 1

        ' remove dots 1 pixel taller than automatically
        ' detected
ScanFix1.DotShadingVertAdjust = 1

        ' in all cases, do not remove dots larger than 4
        ' pixels
ScanFix1.DotShadingMaxSpeckSize = 4

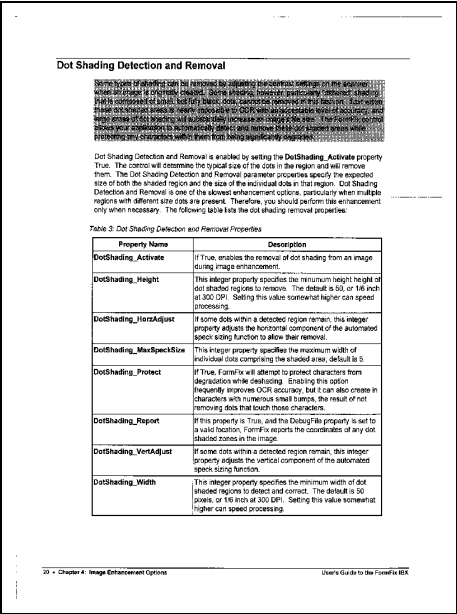
        ' reconstruct characters that are degraded by dot
        ' removal
ScanFix1.DotShadingProtect = True

        ' do not print report to Message event
ScanFix1.DotShadingReport = False

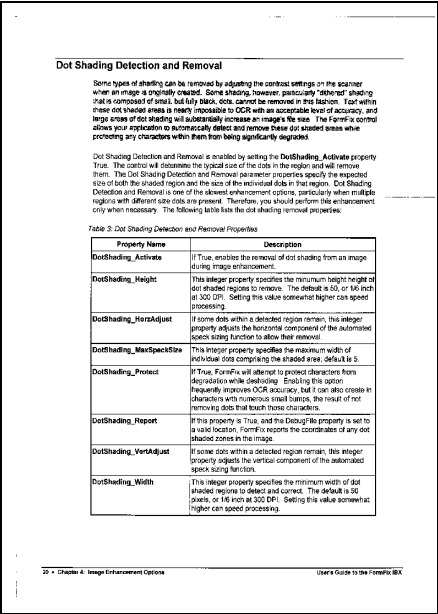
        ' begin processing
ScanFix1.ScanFix

```

The following images show the removal of a dot shaded region from an image. As you can see in the output image on the right, the text beneath the dot shading is now legible and can be read by an OCR engine.



Before shading removal; file size 31.1 KB



After shading removal; file size 29.6 KB

# Inverted Text Correction

The ScanFix control can automatically detect inverted text (white text on a black background) and convert it into normal text (black text on a white background). It will process multiple zones of inverse text on a single page, or an entire inverted image. The ScanFix engine will also recognize and correct inverse zones of different shapes, such as rectangles, circles, ovals, banners, etc. Inverted text correction is most useful because OCR engines presently cannot read white-on-black text. By correcting the source of the difficulty, ScanFix now makes it possible.

## Inverted Text Correction Properties

Inverted text correction must be activated by setting the **InverseActivate** property to True before calling the **ScanFix** method. The following properties set parameters to the detection process:

- InverseActivate
- Must be set to True to activate the detection and correction of inverted text.

InverseEdge	Specifies the horizontal distance between the edges of the inverse (black) zone and the inverse text itself. This value indicates the minimum horizontal length of black pixels along the left and right edges of the inverse zone.
InverseHeight	Specifies the height in pixels of the smallest expected band of inverse text. The default is 50, or 1/6 <sup>th</sup> inch at 300 DPI.
InverseReport	Setting to True will generate a report in the <b>Message</b> event listing the locations of all bands of inverted text.
InverseWidth	Specifies the minimum width of an inverted zone. ScanFix will search for a horizontal black line or area of at least this width. The default is 300 pixels, or one inch at 300 DPI.

### ***Example -- Inverted Text Correction***

For this example, we will be linking the ScanFix control to the TMSFile control:

```

' link the ScanFix control to a file control
ScanFix1.ImageDataSource = TMSFile1.Link

' set InputFrom to memory, since the file
' controls will load the image from disk
ScanFix1.InputFrom = 0

' set the input file name
TMSFile1.InputFileName = "c:\newfiles\tiff0001.tif"

' set OutputTo to file to use the control's
' file I/O libraries
ScanFix1.ImageDesination = 1

' set the OutputFileName
ScanFix1.OutputFileName = "c:\processed\img_001.tif"

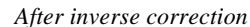
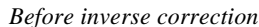
' activate auto-detection and correction of
' inverted text
ScanFix1.InverseActivate = True

' find only inverted regions 300 pixels high (1"
' at 300 DPI)
ScanFix1.InverseHeight = 300

' find only inverted regions 1200 pixels high (4"
' at 300 DPI)

```

The following images demonstrate the results of applying the property settings show immediately above. As you can see, the inverted region from the image on the left have been corrected, allowing the text to be read by OCR engines.



---

## Image Registration and Positioning

The automated processing of images frequently includes the selection of specific regions within each image for further processing such as OCR or operator review. Because the coordinates for these regions are commonly hard-coded into an application, any change in the position of a document as it is being scanned will result in the incorrect region being selected by those coordinates. Registration is one of the most important features that enables the application to correctly position all incoming images. Registration positions the top and left margins of the image data a specific distance from the top and left edges, respectively, of the image.

Horizontal and Vertical Registration can be independently activated, allowing optimum processing for each application. If you require only a consistent left margin, horizontal registration can be enabled by itself. For applications that require the precise location of text in the image, both horizontal and vertical registration should be enabled.

### Horizontal Registration Properties

Horizontal registration automatically positions image data, providing a consistent left margin. ScanFix locates the left edge of text or a vertical line and positions the entire image so that text or line is a specified number of pixels from the left edge. It is strongly advised that you enable Skew Detection and Correction whenever registration is performed.

Horizontal registration is enabled by setting the **HorzRegActivate** property to True. The parameters to the registration process specify details such as the size of the resulting left margin, whether to register to a line or to text, the size of the line to register to, and whether to ignore binder holes and letterheads. Following is a list of the Horizontal Registration properties.

HorzRegActivate	If True, the control will attempt to register the image horizontally. A vertical line must be used for registration, and the following properties set the parameters to this process.
HorzRegAddOnly	Set to True to restrict ScanFix to adding margin. This assures that no part of your image is inadvertently lost, particularly if text extends beyond the line used for registration.
HorzRegCentralFocus	If True, the control searches for lines only in the middle portion of the image border, possibly ignoring letterhead, page numbers, logos, footnotes, etc.
HorzRegFindLine	Specifies the minimum line length that will be used for horizontal registration. Any lines shorter than this number of pixels will be ignored.

HorzRegIgnoreHoles	If True, the control will ignore binder holes and other left-edge noise when searching for a line to register to.
HorzRegLeftMargin	Specifies the final distance from the left edge of the image to the line used for registration. The default is 150 (pixels), or 1/2 inch at 300 DPI.

### ***Example -- Horizontal Registration***

In this example, we are linking the ScanFix control to the file control. It will also receive and send information to this control:

```

' link the ScanFix control to a file control for
' input and to another file control for output
ScanFix1.ImageDataSource = TMSFile1.Link
TMSFile2.ImageDataSource = ScanFix1.Link

' set InputFrom and OutputTo to memory,
' since the file controls will load and save the
' image to disk
ScanFix1.InputFrom = 0           ' ImageDataSource
ScanFix1.OutputTo = 0           ' Memory

' set the input file name
TMSFile1.InputFileName = "c:\newfiles\tiff0001.tif"

' activate registration to the left edge of the
' page
ScanFix1.HorzRegActivate = True

' register to the leftmost vertical line at least
' 200 pixels, or 2/3rds" at 300 DPI, long
ScanFix1.HorzRegFindLine = 200

' leave a margin on the left edge of 30 pixels, or
' 0.1" at 300 DPI
ScanFix1.HorzRegLeftMargin = 30

```

< Continued >



```

' search only the center of the left edge for the
' line
ScanFix1.HorzRegCentralFocus = True

' register the image to the line, even at the
' possible cost of loosing data outside the line
ScanFix1.HorzRegAddOnly = False

' ignore the binder holes when finding a vertical
' line
ScanFix1.HorzRegIgnoreHoles = True

```

The two images below show the horizontal registration performed with the values shown in the sample above. Notice that the left edge of the table was used for the registration, and that it was placed 100 pixels from the edge of the image. This resulted in the loss of several characters in the footer.

**Example – Horizontal Registration, or Registration to a Vertical Line**

The following example shows the interaction of the properties controlling horizontal registration. Following the code is a sample of the results obtained using these property settings.

```

' enable registration to the left edge of the page
FFix1.HorzReg_Activate = True

' find the leftmost vertical line that is at least 100 pixels long
FFix1.HorzReg_FindLine = 100

' leave a margin on the left edge of 100 pixels
FFix1.HorzReg_LeftMargin = 100

' always create the 100 pixel margin, even at the possible cost of
' losing data outside the line
FFix1.HorzReg_AddOnly = False

' search only the center of the left edge for the line
FFix1.HorzReg_CentralFocus = True

' ignore binder holes when finding a vertical line
FFix1.HorzReg_IgnoreHoles = True

' begin enhancement
FFix1.Action = FF_ENHANCE

```

**Vertical Registration**

Table 6. Vertical Registration Properties

Property Name	Description
VertReg_Activate	Enables the registration to the upper and lower margins of the image.
VertReg_AddOnly	Restricts FormFix to adding a margin.
VertReg_FindLine	The minimum length of the line, in pixels, that FormFix will use for registration.
VertReg_TopMargin	Sets the distance, in pixels, the line will be from the top of the page.

**Example – Vertical Registration, or Registration to a Horizontal Line**

```

' activate registration to the upper edge of the page
FFix1.VertReg_Activate = True

' register to the topmost horizontal line that is at least 100
' pixels, or 1.0" at 300 DPI, long
FFix1.VertReg_FindLine = 100

' leave a top margin of 100 pixels, or 0.5" at 300 DPI
FFix1.VertReg_TopMargin = 100

```

User's Guide to the FormFix SDK

Chapter 4: Image Enhancement Options • 23

Before registration

**Example – Horizontal Registration, or Registration to a Vertical Line**

The following example shows the interaction of the properties controlling horizontal registration. Following the code is a sample of the results obtained using these property settings.

```

' enable registration to the left edge of the page
FFix1.HorzReg_Activate = True

' find the leftmost vertical line that is at least 100 pixels long
FFix1.HorzReg_FindLine = 100

' leave a margin on the left edge of 100 pixels
FFix1.HorzReg_LeftMargin = 100

' always create the 100 pixel margin, even at the possible cost of
' losing data outside the line
FFix1.HorzReg_AddOnly = False

' search only the center of the left edge for the line
FFix1.HorzReg_CentralFocus = True

' ignore binder holes when finding a vertical line
FFix1.HorzReg_IgnoreHoles = True

' begin enhancement
FFix1.Action = FF_ENHANCE

```

**Vertical Registration**

Table 6. Vertical Registration Properties

Property Name	Description
VertReg_Activate	Enables the registration to the upper and lower margins of the image.
VertReg_AddOnly	Restricts FormFix to adding a margin.
VertReg_FindLine	The minimum length of the line, in pixels, that FormFix will use for registration.
VertReg_TopMargin	Sets the distance, in pixels, the line will be from the top of the page.

**Example – Vertical Registration, or Registration to a Horizontal Line**

```

' activate registration to the upper edge of the page
FFix1.VertReg_Activate = True

' register to the topmost horizontal line that is at least 100
' pixels, or 1.0" at 300 DPI, long
FFix1.VertReg_FindLine = 100

' leave a top margin of 100 pixels, or 0.5" at 300 DPI
FFix1.VertReg_TopMargin = 100

```

User's Guide to the FormFix SDK

Chapter 4: Image Enhancement Options • 23

After registration

# Vertical Registration

Vertical registration automatically positions image data, providing a consistent top margin. ScanFix locates the top edge of text or a horizontal line and positions the entire image so that text or line is a specified number of pixels from the top edge.

It is strongly advised that you enable Skew Detection and Correction whenever registration is performed.

Vertical registration is enabled by setting the **VertRegActivate** property to True. The parameters to the registration process specify details such as the size of the resulting top margin, whether to register to a line or to text, the size of the line to register to, and whether to limit registration to adding to the existing margin.

VertRegActivate	If True, the control will attempt to register the image vertically. A horizontal line must be used for registration, and the following properties set the parameters to this process.
VertRegAddOnly	Set to True to restrict ScanFix to adding margin. This assures that no part of your image is inadvertently lost, particularly if text extends beyond the line used for registration.
VertRegCentralFocus	If True, the control searches for lines only in the middle portion of the image border, possibly ignoring letterhead, page numbers, logos, footnotes, etc.
VertRegFindLine	Specifies the minimum line length that will be used for vertical registration. Any lines shorter than this number of pixels will be ignored.
VertRegTopMargin	Specifies the final distance from the top edge of the image to the line used for registration. The default is 150 (pixels), or 1/2 inch at 300 DPI.

### ***Example -- Vertical Registration***

For this example, we will link ScanFix to the display control and set the InputFrom and OutputTo to the display

```

' link the ScanFix control to one display control
' for input and another for output
ScanFix1.ImageDataSource = TMSDispl.Link
TMSDispl2.ImageDataSource = ScanFix1.Link

' set the InputFrom and OutputTo
' properties to receive and give information from
' the display
ScanFix1.InputFrom = 0           ' ImageDataSource
ScanFix1.OutputTo = 0           ' Memory

< Continued >

```

```

' activate registration to the upper edge of the page
ScanFix1.VertRegActivate = True

' register to the topmost horizontal line that is
' at least 300 pixels, or 1.0" at 300 DPI, long
ScanFix1.VertRegFindLine = 300

' leave a top margin of 150 pixels, or 0.5" at 300
' DPI
ScanFix1.VertRegTopMargin = 150

' ignore the logo at the top left of the image
ScanFix1.VertRegCentralFocus = True

' register to the line, even at the possible cost
' of loosing data above the line
ScanFix1.VertRegAddOnly = False

' begin processing
ScanFix1.ScanFix

```

The pair of images below demonstrates the results of vertical registration. The image on the left was the original image, and the image on the right was created by vertical registration, using the parameters shown in the preceding sample. The output image is not as tall as the original because the large top margin in the original has been removed.

Licensing Utility Programs	
<p><b>Licensing Configuration Manager (LCM)</b></p> <p>The LCM is installed with all installations of ImageBASIC. This utility is used by our customers to:</p> <ol style="list-style-type: none"> <li>1. create licensing databases</li> <li>2. add new licenses to licensing databases or hardware keys</li> <li>3. review the contents of either a licensing database or hardware key</li> </ol> <p>The LCM allows the person who is running it to select the type and number of licenses that will be installed, and then requests the input of an Authorization Code. This authorization code can be obtained either from DMS or, for runtime licenses, from any developer who is using the RLA (see below).</p> <p>The LCM can be distributed by DMS customers to their clients, allowing the end-user to install and configure runtime licenses.</p> <p><b>LCM Use</b></p> <p>The LCM can be installed and upgrade licensing databases and runtime hardware keys. It differs from the standard LCM only in being less complex by offering fewer options to the user. Instead of allowing the user to select the type and number of licenses to install, the LCM file reads this information from a Pricing File. These products files are created by the RLA (see below) and downloaded to the end user along with the application and with ImageBASIC.</p> <p>For example, the developer might distribute two license pricing files:</p> <ol style="list-style-type: none"> <li>1. Single-user package             <ul style="list-style-type: none"> <li>1. ImageWin32key.txt</li> </ul> </li> <li>2. Multi-user package             <ul style="list-style-type: none"> <li>1. ImageWin32key.txt</li> <li>2. ImageWin32key.txt</li> <li>3. ImageWin32key.txt</li> <li>4. ImageWin32key.txt</li> </ul> </li> </ol> <p>The client can install the single user package for testing and then upgrade to the multi-user package when moving to full production use of the application. The client will upgrade the LCM file and select the runtime license package, and the developer can then provide the necessary authorization code.</p> <p><b>LCM vs. LCM Lite</b></p> <ul style="list-style-type: none"> <li>• The LCM allows the user to select the license that will be installed, but cannot use License Pricing Files.</li> <li>• The LCM Lite can only install licenses from a Pricing File.</li> </ul>	<p><b>Software License (SWL) Viewer</b></p> <p>After the above utilities are used to install a licensing database, the user will naturally want to check the results of the database. Two methods are available to do this. First, the Licensing server (dbscan) can create a log file that requests, all licensing requests by host machine and the results of those requests. Second, the SWL Viewer utility can be run to read the database and report which licenses are currently being used by all client machines accessing the database. This utility can also be instructed to generate hierarchical reports on the state of the database, allowing an administrator to keep track of exactly how many licenses are in use at any given time.</p> <p>The LCM can be distributed by DMS customers to their clients, allowing the end-user to install and configure runtime licenses.</p> <p><b>Runtime Licensing Authorization (RLA)</b></p> <p>The RLA is used to generate authorization codes required for the distribution of runtime licenses. We provide this utility to our customers, allowing them to distribute runtime licenses to their customers. DMS is therefore not involved with the end user, who may not even know that we exist.</p> <p><b>RLA Use</b></p> <p>The RLA can be installed and upgrade licensing databases and runtime hardware keys. It differs from the standard LCM only in being less complex by offering fewer options to the user. Instead of allowing the user to select the type and number of licenses to install, the LCM file reads this information from a Pricing File. These products files are created by the RLA (see below) and downloaded to the end user along with the application and with ImageBASIC.</p> <p>For example, the developer might distribute two license pricing files:</p> <ol style="list-style-type: none"> <li>1. Single-user package             <ul style="list-style-type: none"> <li>1. ImageWin32key.txt</li> </ul> </li> <li>2. Multi-user package             <ul style="list-style-type: none"> <li>1. ImageWin32key.txt</li> <li>2. ImageWin32key.txt</li> <li>3. ImageWin32key.txt</li> <li>4. ImageWin32key.txt</li> </ul> </li> </ol> <p>The client can install the single user package for testing and then upgrade to the multi-user package when moving to full production use of the application. The client will upgrade the LCM file and select the runtime license package, and the developer can then provide the necessary authorization code.</p> <p><b>RLA vs. DMS</b></p> <ul style="list-style-type: none"> <li>• The RLA can only authorize runtime licenses that are being installed by the LCM or LCM Lite.</li> <li>• The RLA can only authorize runtime licenses that are being installed by the LCM or LCM Lite.</li> </ul>

Before registration

After registration

---

## Line Detection and Removal

Line detection and removal is the process by which ScanFix automatically detects, optionally reports, and removes horizontal and vertical lines from an image. Horizontal and vertical lines are managed separately. If a line intersects a character, and it is removed, the character will be distorted. For this reason, ScanFix includes the option to rebuild any characters that are impacted by the line removal. This reconstruction feature does add to the processing time, and should therefore be enabled only when necessary.

### Horizontal Line Detection and Removal Properties

The detection and removal of horizontal lines is enabled by setting the **HorzLine\_Activate** property to True. Several additional properties are then available to optimize the detection process. The following table describes these parameter properties. The following properties control the line detection and removal feature of the ScanFix control:

HorzLineActivate	Set to True to perform removal of horizontal lines in an image. The following <b>HorzLine...</b> properties specify the characteristics of the lines to remove.
HorzLineClean	Specifies the distance in pixels the line removal algorithm will search above and below the line itself. Any stray pixels in this region will be removed with the line. The default is 2.
HorzLineMaxGap	<p>Specifies the maximum gap along the length of the line that is allowed. Any gap above this size will be interpreted as a division between two lines. Poor quality lines will generally have gaps which prevent their recognition.</p> <p><b>Note:</b> For extremely poor-quality images, such as dot-matrix and microfilm documents, you may want to set this value as high as 6. However, if the value is set too high, portions of text may be removed as well.</p>
HorzLineMaxWidth	<p>Specifies the maximum width of lines the control will remove. Default is 22 pixels.</p> <p><b>Note:</b> The capital letter <b>T</b> in a large title or headline has a horizontal component which could be mistaken for a line and removed. <i>To avoid erroneous line removal, set <b>HorzLineMaxWidth</b> larger than the largest text.</i></p>

HorzLineMinLength	Specifies the minimum length of lines the control will remove. Default is 150, or 1/2 inch at 300 DPI.
HorzLineReconstruct	Specifies the size of the largest character that will be repaired after an intersecting line is removed. It sets the maximum pixel width of characters to be repaired.
HorzLineReport	Specifies whether the control will locate and/or remove the lines it finds. The values for <b>HorzLineReport</b> are as follows: <ul style="list-style-type: none"> <li>0        Omit (Omits reports and removes lines)</li> <li>1        Locate and Remove</li> <li>2        Locate Only (Reports but does not remove)</li> </ul>

Many of the common problems encountered in line removal, most notably a failure to identify lines because of multiple small gaps in the lines, can usually be remedied by the application the premium feature Character Smoothing. The **CompletionSmooth** property controls this feature and is detailed in the next section.

### ***Example -- Horizontal Line Detection and Removal***

In a 300 DPI image, the following settings of the line detection and removal parameters will cause all vertical lines at least one (1) inch long and all horizontal lines at least two (2) inches long to be removed. Compensation is made for small gaps in the lines, and any degraded characters are repaired after the removal is complete. For this example, we will be linking the ScanFix control to the TMSFile control:

```

' set InputFrom to file to read directly from file
ScanFix1.InputFrom = 1

' set the input file name
ScanFix1.InputFileName = "c:\newfiles\tiff0001.tif"

' set OutputTo to file to use the control's
' file I/O libraries
ScanFix1.OutputTo = 1          ' File

' set the OutputFileName
ScanFix1.OutputFileName = "c:\processed\img_001.tif"

' enable the auto detect and the horizontal line
' parameter properties
ScanFix1.HorzLineActivate = True

' removal isolated specks up to 2 pixels from the
' line

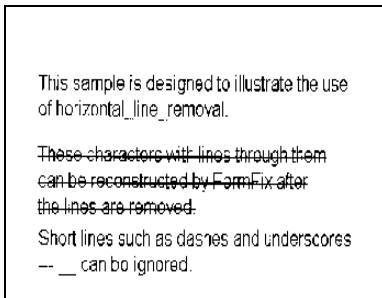
```

```

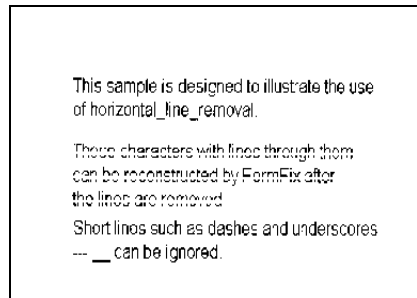
ScanFix1.HorzLineClean = 2
    ' allow for gaps of up to 5 pixels, or 1/60th at
    ' 300 DPI
ScanFix1.HorzLineMaxGap = 5
    ' detect only horizontal lines at least 20 pixels
    ' wide, or 1/15th at 300 DPI
ScanFix1.HorzLineMaxWidth = 10
    ' detect only horizontal lines at least 300 pixels
    ' long, or 1" at 300 DPI
ScanFix1.HorzLineMinLength = 300
    ' repair characters up to 50 pixels wide, or 1/6th
    ' at 300 DPI intersected by horizontal line
ScanFix1.HorzLineReconstruct = 50
    ' do not send a report to the Message event
ScanFix1.HorzLineReport = 0

```

The following images illustrate the removal of horizontal lines and the reconstruction of characters after an intersecting line is removed. The two images on the left contrast line removal with (top image) and without (bottom image) character reconstruction. Note also that the short underscores are not removed because the minimum length was set to 300 pixels, or one inch (1") in this image.



*Before line removal*



*After line removal without reconstruction*

This sample is designed to illustrate the use of horizontal\_line\_removal.

These characters with lines through them can be reconstructed by FormFix after the lines are removed.

Short lines such as dashes and underscores --- \_ can be ignored.

*After line removal with reconstruction*

## Vertical Line Detection and Removal

The detection and removal of vertical lines is enabled by setting the **VertLineActivate** property to True. The detection and removal of vertical lines is controlled separately. After line removal is enabled, several additional properties are then available to optimize the detection process. The following table describes these parameter properties.

VertLineActivate	Set to True to perform removal of vertical lines in an image. The following <b>VertLine...</b> properties specify the characteristics of the lines to remove.
VertLineClean	Specifies the distance in pixels the line removal algorithm will search to the right and left of the line itself. Any stray pixels in this region will be removed with the line. The default is 2.
VertLineMaxGap	Specifies the maximum gap along the length of the line that is allowed. Any gap above this size will be interpreted as a division between two lines. Poor quality lines will generally have gaps which prevent their recognition.
VertLineMaxWidth	Specifies the maximum width of lines the control will remove. Default is 22 pixels.

**Note:** The capital letter **T** in a large title or headline has a vertical component which could be mistaken for a line and removed. *To avoid erroneous line removal, set **VertLineMaxWidth** larger than the largest text.*

VertLineMinLength	Specifies the minimum length of lines the control will remove. Default is 150, or 1/2 inch at 300 DPI.
VertLineReconstruct	Specifies the size of the largest character that will be repaired after an intersecting line is removed. Sets

the maximum pixel height of characters to be repaired.

VertLineReport

Specifies whether the control will locate and/or remove the lines it finds. The values for

**HorzLineReport** are as follows:

- 0      Omit (Omits reports and removes lines)
- 1      Locate and Remove
- 2      Locate Only (Reports but does not remove)

### ***Example -- Vertical Line Detection and Removal***

For this example, we will be linking the ScanFix control to the TMSFile control:

```
' link the ScanFix control to a file control
ScanFix1.ImageDataSource = TMSFile1.Link

' set InputFrom to memory, since the file
' controls will load the image from disk
ScanFix1.InputFrom = 0

' set the input file name
TMSFile1.InputFileName = "c:\newfiles\tiff0001.tif"

' set OutputTo to file to use the control's
' file I/O libraries
ScanFix1.OutputTo = 1

' set the OutputFileName
ScanFix1.OutputFileName = "c:\processed\img_001.tif"

' activate the vertical line detection and the
' vertical line parameter properties
ScanFix1.VertLineActivate = True

' removal isolated specks up to 3 pixels from the
' line
ScanFix1.VertLineClean = 3

' allow for gaps of up to 10 pixels, or 1/30th at
' 300 DPI
ScanFix1.VertLineMaxGap = 10

' detect only vertical lines at least 20 pixels
' wide, or 1/15th at 300 DPI
ScanFix1.VertLineMaxWidth = 20
```

< Continued >



```

' detect only horizontal lines at least 600 pixels
' long, or 2" at 300 DPI
ScanFix1.VertLineMinLength = 600

' do not perform reconstruction on intersected
' characters
ScanFix1.VertLineReconstruct = 0

' do not send a report to the Message event
ScanFix1.VertLineReport = False

' begin processing
ScanFix1.ScanFix

```

The image on the right, below, was the result of applying the property settings shown above to the image on the left. Note that the dotted line on the left was not removed because it contained gaps larger that 10 pixels. The dotted line on the right was removed because its gaps are smaller than 10 pixels.

Table 8: Vertical Line Detection and Removal Properties

Property Name	Description
VertLine_Activate	If True, enables vertical line detection and removal during image enhancement.
VertLine_Clean	When FormFix is removing vertical lines, it can also locate and remove random noise near the line. This property specifies how far from the line, in pixels, the engine will search for the noise. The default is 2.
VertLine_MaxGap	Specifies the largest gap, in pixels, that can be present along a line that FormFix will accept and consider both fragments part of a single line.
VertLine_MaxWidth	As an aid to the protection of large type in documents, this property specifies the maximum line height, in pixels, that will be removed. Because letters are generally thicker than normal lines, the removal process can be optimized to ignore the thicker letters.
VertLine_MinLength	Specifies the minimum length, in pixels, that a line must reach in order to be removed. VertLine_MaxGap limits the size of spaces along this line.
VertLine_Reconstruct	When a line intersects a character and the line is removed, FormFix can reconstruct the character. This property specifies the height of the largest character that will be repaired after an intersecting line is removed.
VertLine_Report	If set to <code>True</code> (Locate & Report), FormFix reports the pixel coordinates of the lines that it detects. The report is sent to the debug window that is enabled by the <code>MessageFlag</code> property.

*Before vertical line removal*

Table 8: Vertical Line Detection and Removal Properties

Property Name	Description
VertLine_Activate	If True, enables vertical line detection and removal during image enhancement.
VertLine_Clean	When FormFix is removing vertical lines, it can also locate and remove random noise near the line. This property specifies how far from the line, in pixels, the engine will search for the noise. The default is 2.
VertLine_MaxGap	Specifies the largest gap, in pixels, that can be present along a line that FormFix will accept and consider both fragments part of a single line.
VertLine_MaxWidth	As an aid to the protection of large type in documents, this property specifies the maximum line height, in pixels, that will be removed. Because letters are generally thicker than normal lines, the removal process can be optimized to ignore the thicker letters.
VertLine_MinLength	Specifies the minimum length, in pixels, that a line must reach in order to be removed. VertLine_MaxGap limits the size of spaces along this line.
VertLine_Reconstruct	When a line intersects a character and the line is removed, FormFix can reconstruct the character. This property specifies the height of the largest character that will be repaired after an intersecting line is removed.
VertLine_Report	If set to <code>True</code> (Locate & Report), FormFix reports the pixel coordinates of the lines that it detects. The report is sent to the debug window that is enabled by the <code>MessageFlag</code> property.

*After vertical line removal*

## Skew Detection and Correction

ScanFix can automatically straighten image data (text and lines) relative to the page. Deskewing is invaluable for applications which require precise and consistent placement of text, such as Forms Processing and the intelligent OCR of data. Even at 1% to 2% skew, many OCR engines begin to confuse characters. ScanFix automatically detects and corrects up to 20% skew.

Lines are not necessary for ScanFix to detect and correct text skew. ScanFix will use lines where they are available (and representative of the document as a whole), but

will also detect and correct skew from the text itself. It is recommended that you perform deskewing for most application.

Deskewing an image can also improve the performance of compression algorithms by ordering the white space between lines. Skewed TIFF Group 4 files can compress up to an additional 25% when deskewed.

## Deskewing Properties

The deskewing feature of ScanFix is controlled through the following properties:

DeskewActivate	If True, the image data will be deskewed based on either horizontal lines or text.
DeskewCharacterProtect	May be set to True to minimize the distortion of characters that is possible when ScanFix corrects a large degree of skew. This is only necessary when the original image text is skewed over about 15%.
DeskewMaxAcceptableSkew	<p>Specifies the maximum amount of skew considered acceptable. Larger values of <b>DeskewMaxAcceptableSkew</b> result in more precise the Deskewing. The default setting of 150 means that the resultant image will have no more than one pixel of vertical skew for each 150 pixels horizontally. This corresponds to 0.67% skew, and produces good OCR results.</p> <p>A setting of 100 corresponds to 1% skew; a setting of 200 corresponds to 0.5% skew. Set to lower values for higher speed.</p>
DeskewMinDetectLength	<p>Specifies the minimum length of a run of text or a line used to detect skew. The default setting is 300 pixels, or 1" at 300 DPI. When a length of text or a line is detected that exceeds this limit, the <b>DeskewMaxAcceptableSkew</b> property is consulted to determine whether the image should be straightened.</p>

## Example -- Deskewing

The following code segment will identify skew based on a run of characters at least 600 pixels long (2" at 300 DPI) and will perform deskewing if more than a 0.5% skew is detected. The **DeskewCharacterProtect** property is True to allow for the possibility of very badly skewed text, even though protection will cause the process to take a little longer. In this example, we are linking the ScanFix control to the file control. It will also receive and send information to this control:

```
' link the ScanFix control to a file control for
' input and to another file control for output
ScanFix1.ImageDataSource = TMSFile1.Link
TMSFile2.ImageDataSource = ScanFix1.Link

' set InputFrom and OutputTo to memory,
' since the file controls will load and save the
' image to disk
ScanFix1.InputFrom = 0
ScanFix1.OutputTo = 0

' set the input file name
TMSFile1.InputFileName = "c:\newfiles\tiff0001.tif"

' activate the deskewing feature of the control
ScanFix1.DeskewActivate = True

' find a run of text or a horizontal line at least
' 600 pixels long in order to determine image skew
ScanFix1.DeskewMinDetectLength = 600

' accept no more than one (1) pixel of skew over
' 200 pixels
ScanFix1.DeskewMaxAcceptableSkew = 200

' correct any characters distorted by straightening
' a badly skewed image
ScanFix1.DeskewCharacterProtect = True

' begin processing
ScanFix1.ScanFix
```

The following images illustrate the correction of skew in a typical image.

... / to True. The ...  
 ... Line Detection and Removal ...  
 ... several additional properties are then available ...  
 ... following table describes these parameter properties.

Property Name	Description
Line_Activate	If True, enables vertical line detection and removal during image enhancement.
Line_Clean	When FormFix is removing vertical lines, it can also locate and remove random noise near the line. This property specifies how far from the line, in pixels, the engine will search for the noise. The default is 2.
VertLine_MaxGap	Specifies the largest gap, in pixels, that can be present along a line that FormFix will accept and consider both fragments part of a single line.
VertLine_MaxWidth	As an aid to the protection of large type in documents, this property specifies the maximum line height, in pixels, that will be removed. Because letters are generally thicker than normal lines, the removal process can be optimized to ignore the thicker letters.
VertLine_MinLength	Specifies the minimum length, in pixels, that a line must reach in order to be removed. VertLine_MaxGap limits the size of spaces along this line.
VertLine_Reconstruct	When a line intersects a character and the line is removed, FormFix can reconstruct the character. This property specifies the height of the largest character that will be repaired after an intersecting line is removed.
VertLine_Report	If set to 1-Locate & Report, FormFix reports the pixel coordinates of the lines that it detects. The report is sent to the debug window that is enabled by the MessageFlag

*Before skew correction*

... / to True. The ...  
 ... Line Detection and Removal ...  
 ... several additional properties are then available ...  
 ... following table describes these parameter properties.

Property Name	Description
Line_Activate	If True, enables vertical line detection and removal during image enhancement.
Line_Clean	When FormFix is removing vertical lines, it can also locate and remove random noise near the line. This property specifies how far from the line, in pixels, the engine will search for the noise. The default is 2.
VertLine_MaxGap	Specifies the largest gap, in pixels, that can be present along a line that FormFix will accept and consider both fragments part of a single line.
VertLine_MaxWidth	As an aid to the protection of large type in documents, this property specifies the maximum line height, in pixels, that will be removed. Because letters are generally thicker than normal lines, the removal process can be optimized to ignore the thicker letters.
VertLine_MinLength	Specifies the minimum length, in pixels, that a line must reach in order to be removed. VertLine_MaxGap limits the size of spaces along this line.
VertLine_Reconstruct	When a line intersects a character and the line is removed, FormFix can reconstruct the character. This property specifies the height of the largest character that will be repaired after an intersecting line is removed.
VertLine_Report	If set to 1-Locate & Report, FormFix reports the pixel coordinates of the lines that it detects. The report is sent to the debug window that is enabled by the MessageFlag

*After skew correction*

## SubImage Selection

The ScanFix control can extract an area from the image after deskewing and registering, but before other ScanFix processing. This feature is rarely used by most users and is useful for applications where only part of the document is needed at a time. It is strongly recommended that deskewing and registration be activated when a sub-image is being defined. The sub-image will be extracted after the entire image has been straightened.

## SubImage Selection Properties

The following properties are used to control the identification and extraction of a region within the image being processed. The region specified here will be the entire image returned by the ScanFix control.

- |                    |  |
|--------------------|--|
| SubImageActivate   | If True, the values of the following properties will be applied to select and extract a region within the image being processed.   |
| SubImageBottomEdge | Specifies the lower edge of the region that will be extracted. The value in this property is the number of pixels from the top of the image to the bottom of the region. |
| SubImageLeftEdge   | Specifies the left edge of the region that will be extracted. The value in this property is the number of pixels from the left of the image to the left of the region.   |

SubImagePad	After extraction, the ScanFix control can add a white border to the perimeter of the extracted region. If the <b>SubImagePad</b> property is set to any non-zero integer value, a white border will be added of the specified width.
SubImageRightEdge	Specifies the right edge of the region that will be extracted. The value in this property is the number of pixels from the left edge of the image to the right of the region.
SubImageTopEdge	Specifies the top edge of the region that will be extracted. The value in this property is the number of pixels from the top of the image to the top of the region.

### ***Example -- SubImage Selection***

The following segment of code will extract the top half of an image read from the Image Display Window and place a border 100 pixels wide around it. Refer to the *ImageBASIC User's Guide* for details on the Image Display Window properties used in this example. For this example, we will be linking the ScanFix control to the TMSFile control:

```

' link the ScanFix control to a file control
ScanFix1.ImageDataSource = TMSFile1.Link

' set InputFrom to memory, since the file
' controls will load the image from disk
ScanFix1.InputFrom = 0

' set the input file name
TMSFile1.InputFileName = "c:\newfiles\tiff0001.tif"

' set OutputTo to file to use the control's
' file I/O libraries
ScanFix1.OutputTo = 1

' set the OutputFileName
ScanFix1.OutputFileName = "c:\processed\img_001.tif"

' activate the SubImage selection and extraction
' feature
ScanFix1.SubImageActivate = True

' start the region at the left edge of the image
ScanFix1.SubImageLeftEdge = 0

' start the region at the top edge of the image

```

```
ScanFix1.SubImageTopEdge = 0

' make the SubImage region the entire image width
ScanFix1.SubImageRightEdge = TMSFile1.PageWidth

' make the SubImage region one-half the image
' height
ScanFix1.SubImageBottomEdge = (TMSFile1.PageHeight / 2)

' draw a white border 100 pixels wide around the
' extracted region
ScanFix1.SubImagePad = 100

' begin processing
ScanFix1.ScanFix
```

The following images illustrate the application of the property settings shown in the code above. Note that only the top half of the original image is present in the enhanced image.

**Vertical Line Detection and Removal**

The detection and removal of vertical lines is enabled by setting the `VertLine_Activate` property to `True`. The detection and removal of vertical lines is controlled separately (see "Horizontal Line Detection and Removal" on page 31 for details). After line removal is enabled, several additional properties are then available to optimize the detection process. The following table describes these parameter properties.

**Table 8. Vertical Line Detection and Removal Properties**

Property Name	Description
<code>VertLine_Activate</code>	If <code>True</code> , enables vertical line detection and removal during image enhancement.
<code>VertLine_Clean</code>	When <code>FormFix</code> is removing vertical lines, it can also locate and remove random noise near the line. This property specifies how far from the line, in pixels, the engine will search for the noise. The default is 2.
<code>VertLine_MaxGap</code>	Specifies the largest gap, in pixels, that can be present along a line that <code>FormFix</code> will accept and consider both fragments part of a single line.
<code>VertLine_MaxWidth</code>	As an aid to the protection of large type in documents, this property specifies the maximum line height, in pixels, that will be removed. Because letters are generally thicker than normal lines, the removal process can be optimized to ignore the thicker letters.
<code>VertLine_MinLength</code>	Specifies the minimum length, in pixels, that a line must reach in order to be removed. <code>VertLine_MaxGap</code> limits the size of spaces along the line.
<code>VertLine_Reconstruct</code>	When a line intersects a character and the line is removed, <code>FormFix</code> can reconstruct the character. This property specifies the height of the largest character that will be repaired after an intersecting line is removed.
<code>VertLine_Report</code>	If set to <code>1-Locate &amp; Report</code> , <code>FormFix</code> reports the pixel coordinates of the lines that it detects. The report is sent to the debug window that is enabled by the <code>MessageFlag</code> property.

**Example - Vertical Line Detection and Removal**

The following code segment illustrates the interaction of the vertical line detection and removal properties. In this example, removal is first enabled, and then the details of the removal process are set. First, the control is linked to removing only lines at least 600 pixels long, but no more than 20 pixels wide. When determining if a line reaches the minimum length, gaps of up to 10 pixels will be allowed.

When the line is removed, the control is instructed to find and remove isolated specks within 3 pixels of the line. Character reconstruction is very important if any characters are intersected by removed lines, but in this example, it is assumed that the form does not contain any such characters. The last two commands instruct the control not to output location information to the debug window and then begin enhancement. Following the code segment you will find images that illustrate the application of these property settings.

User's Guide to the FormFix BEX Chapter 4: Image Enhancement Options • 31

Before subimage processing

**Vertical Line Detection and Removal**

The detection and removal of vertical lines is enabled by setting the `VertLine_Activate` property to `True`. The detection and removal of vertical lines is controlled separately (see "Horizontal Line Detection and Removal" on page 31 for details). After line removal is enabled, several additional properties are then available to optimize the detection process. The following table describes these parameter properties.

**Table 8. Vertical Line Detection and Removal Properties**

Property Name	Description
<code>VertLine_Activate</code>	If <code>True</code> , enables vertical line detection and removal during image enhancement.
<code>VertLine_Clean</code>	When <code>FormFix</code> is removing vertical lines, it can also locate and remove random noise near the line. This property specifies how far from the line, in pixels, the engine will search for the noise. The default is 2.
<code>VertLine_MaxGap</code>	Specifies the largest gap, in pixels, that can be present along a line that <code>FormFix</code> will accept and consider both fragments part of a single line.
<code>VertLine_MaxWidth</code>	As an aid to the protection of large type in documents, this property specifies the maximum line height, in pixels, that will be removed. Because letters are generally thicker than normal lines, the removal process can be optimized to ignore the thicker letters.
<code>VertLine_MinLength</code>	Specifies the minimum length, in pixels, that a line must reach in order to be removed. <code>VertLine_MaxGap</code> limits the

After subimage processing

---

## Removing Runs of Periods

The ScanFix control can remove runs of periods ( ..... ), also called dot leaders, from forms, tables of contents, etc. Periods interfere with OCR engines and prevent accurate formatting of text.

PeriodRemovalExpectedFrequency	Specifies the expected number of pixels between each dot in the run. If set to any positive non-zero integer, the control will attempt to locate a series of dots approximately that distance apart.
PeriodRemovalMaxHeight	Specifies the maximum expected height of each dot in a run of periods. Any dots larger than this value will be ignored by the control when it is identifying runs of dots.

### ***Example -- Period Removal***

The following routine will cause the ScanFix control to find all runs of periods composed of dots at most 6 pixels high, or .05" at 300 DPI. All identified runs will be removed. In this example, we are linking the ScanFix control to the file control. It will also receive and send information to this control:

```
' link the ScanFix control to a file control for
' input and to another file control for output
ScanFix1.ImageDataSource = TMSFile1.Link
TMSFile2.ImageDataSource = ScanFix1.Link

' set InputFrom and OutputTo to memory,
' since the file controls will load and save the
' image to disk
ScanFix1.InputFrom = 0
ScanFix1.OutputTo = 0

' set the input file name
TMSFile1.InputFileName = "c:\newfiles\tiff0001.tif"

' specify the expected distance between dots, in
' pixels
ScanFix1.PeriodRemovalExpectedFrequency = 10

' specify the maximum height of dots to remove, in
' pixels
ScanFix1.PeriodRemovalMaxHeight = 6

' begin processing
ScanFix1.ScanFix
```

The following images illustrate the removal of dot leaders in a table of contents. The removal of these runs allows for more accurate OCR and text positioning in the OCR output.

Contents	
Chapter 1: Introduction to the FormFix iBX	1
Welcome	1
Adding the FormFix iBX to Your Application	1
Installing the FormFix iBX	1
Adding FormFix to An Application	1
Loading Verification	2
Using the FormFix iBX	3
System Requirements	3
Programming Considerations	3
Error Handling and Debugging	4
Chapter 2: Forms Processing	7
Form Processing Processes	7
Creating a Master Form	7
Deleting a Master Form	11
Form Identification	13
Form Extraction	14
Chapter 3: Image Enhancement	17
Image Enhancement Options	17
Enhancing an Image	17
Specify the Image Source for Enhancement	17
Specify the Image Destination for Enhancement	18
Advanced Image Enhancement Options	18
Start Image Enhancement	18
Chapter 4: Image Enhancement Options	19
Enabling Image Enhancement	19
Specck Removal	20
Dot Clipping Detection and Removal	22
Inverted Text Detection and Correction	25
Image Registration and Positioning	27
Horizontal Registration	27
Vertical Registration	29
Line Detection and Removal	31
Bar Detection and Correction	34
Subimage Processing	37
Dot Leader Detection and Removal	38
Premium Enhancement Options	39
Character Smoothing	39
Automatic Orientation and Rotation	40
Removing Borders - Intelligent Crop	41
Controlling Property	42
Outcropping Symbols	42
Processing Mask Options	42
Image Cropping Options	43
Autotriple Specific Options	44
Debugging Options	45
Desktop Control Options	45
Dot Shading Options	46
User's Guide to the FormFix iBX	
Contents v i	

Before dot leader removal

Contents	
Chapter 1: Introduction to the FormFix iBX	1
Welcome	1
Adding the FormFix iBX to Your Application	1
Installing the FormFix iBX	1
Adding FormFix to An Application	1
Loading Verification	2
Using the FormFix iBX	3
System Requirements	3
Programming Considerations	3
Error Handling and Debugging	4
Chapter 2: Forms Processing	7
Form Processing Processes	7
Creating a Master Form	7
Deleting a Master Form	11
Form Identification	13
Form Extraction	14
Chapter 3: Image Enhancement	17
Image Enhancement Options	17
Enhancing an Image	17
Specify the Image Source for Enhancement	17
Specify the Image Destination for Enhancement	18
Advanced Image Enhancement Options	18
Start Image Enhancement	18
Chapter 4: Image Enhancement Options	19
Enabling Image Enhancement	19
Specck Removal	20
Dot Clipping Detection and Removal	22
Inverted Text Detection and Correction	25
Image Registration and Positioning	27
Horizontal Registration	27
Vertical Registration	29
Line Detection and Removal	31
Bar Detection and Correction	34
Subimage Processing	37
Dot Leader Detection and Removal	38
Premium Enhancement Options	39
Character Smoothing	39
Automatic Orientation and Rotation	40
Removing Borders - Intelligent Crop	41
Controlling Property	42
Outcropping Symbols	42
Processing Mask Options	42
Image Cropping Options	43
Autotriple Specific Options	44
Debugging Options	45
Desktop Control Options	45
Dot Shading Options	46
User's Guide to the FormFix iBX	
Contents v i	

After dot leader removal

## Premium Enhancement Options

In addition to the standard enhancement options detailed above, several premium enhancement options are also available. Enabling and applying these options follows the same procedure as the standard options. For more information on this procedure, please refer to Enhancing an Image on page 7.

The premium enhancement options are as following:

- Character Smoothing and Advanced Line Removal
- Automatic Orientation and Rotation
- Intelligent Cropping



---

## Character Completion

Many times, images are presented for processing that are distorted through faxing or the scanning process. The text in these images may be stretched or shrunk, resulting in an unintentionally bold or faint characters. The ScanFix control allows the Visual Basic developer to correct these problems by adding to or removing from the individual characters. The Completion properties of the control allow the custom application of either adding extra pixels to, or removing pixels from, each image object.

For example, an image or a region of an image may contain text that has been accidentally made very bold during the digitizing process. By the removal of one or two pixels from the entire perimeter of the characters, they can be returned to nearly their original appearance.

### ***Character Completion Properties***

The following properties control the Character Completion features of the ScanFix control:

CompletionErode	Specifies the number of pixels that will be removed from the perimeter of each individual object in an image. For example, if this property is set to 2, a dot measuring 6 by 6 pixels would be reduced to a 2 by 2 pixel dot. <b>CompletionErode</b> is an integer property that defaults to 0.
CompletionGrow	Specifies the number of pixels that will be added to the perimeter of each individual object in an image. For example, if this property is set to 2, a dot measuring 6 by 6 pixels would be increased to a 10 by 10 pixel dot. <b>CompletionGrow</b> is an integer property that defaults to 0.
CompletionCustom	Allows the individual setting of erosion and growth for the horizontal and vertical planes. For example, the control may be instructed to grow characters in the vertical plane but erode them horizontally.
H#	Grows the image by # pixels horizontally
V#	Grows the image by # pixels vertically
h#	Erodes the image by # pixels horizontally
v#	Erodes the image by # pixels vertically

## Example -- Character Completion

The following segment of code uses the CompletionCustom property to *increase* the width of all image objects while *decreasing* their height. For this example, we will link ScanFix to the display control and set the InputFrom and OutputTo to the display

```
' link the ScanFix control to one display control
' for input and another for output
ScanFix1.ImageDataSource = TMSDispl.Link
TMSDisp2.ImageDataSource = ScanFix1.Link

' set the InputFrom and OutputTo
' properties to receive and give information from
' the display
ScanFix1.InputFrom = 0
ScanFix1.OutputTo = 0

' set the amount for horizontal and vertical
' completion (add two (2) pixels horizontally and
' remove one (1) vertically)
ScanFix1.CompletionCustom = "H2 v1"

' begin processing
ScanFix1.ScanFix
```

---

## Character Smoothing

A feature that is related to the Character Completion detailed above, Character Smoothing can be used to improve character quality. The advantages of Character Smoothing are most obvious when considering dot matrix text. Character Smoothing performs a "sand and fill" operation, smoothing out the edges of characters, improving their appearance, and reducing image file size. This smoothing also enhances OCR accuracy, particularly if the image was badly skewed or printed at a low resolution.

## Character Smoothing Properties

Only one property enables and configures the Character Smoothing action of the ScanFix control:

CompletionSmooth	Integer value specifies the extent of "sand and fill" smoothing. Larger values result in coarser smoothing. Numbers between 1 and 5 work well, but coarser smoothing (i.e., larger values) may be required for some images.
------------------	---

Enhanced Line Removal, part of the Character Smoothing option, removes low resolution or distorted lines that the standard line management options may miss. To perform enhance line removal on an image, set the **CompletionSmooth** property and the "Line Detection and Removal" on page 24 properties to the appropriate values.

### ***Example -- Character Smoothing***

The following segment of code will correct dot matrix print, making the characters more regular and improving OCR accuracy. For this example, we will be linking the ScanFix control to the TMSFile control:

```
' link the ScanFix control to a file control
ScanFix1.ImageDataSource = TMSFile1.Link

' set InputFrom to memory, since the file
' controls will load the image from disk
ScanFix1.InputFrom = 0

' set the input file name
TMSFile1.InputFileName = "c:\newfiles\tiff0001.tif"

' set OutputTo to file to use the control's
' file I/O libraries
ScanFix1.OutputTo = 1

' set the OutputFileName
ScanFix1.OutputFileName = "c:\processed\img_001.tif"

' specify the amount for character smoothing
ScanFix1.CompletionSmooth = 3

' begin processing
ScanFix1.ScanFix
```

---

## Intelligent Cropping, or Removing Borders

ScanFix offers two options for the automatic detection and removal of solid borders from an image. With the Intelligent Cropping feature, black or white borders can be automatically removed. Intelligent Cropping, as controlled by the **CropWhite** and **CropBlack** properties, is not included in the base ScanFix Control package, but is available for an additional license fee

### Intelligent Cropping Properties

The following properties are used to control the removal of borders from images:

CropBlack	If True, any regular or irregular black borders around an image will be "bleached" to white. Only the region outside an rectangle drawn around the text will be effected.
CropWhite	If True, the image will be stripped of any white border that is outside the area defined by a regular rectangle drawn around the text in an image. This removal registers the image data to the top and left.

Certain aspects of the identification and removal of borders can be controlled through the **OptionString** property. This property may be used to allow for noise within the border or to remove only vertical or horizontal borders. Intelligent Cropping is a premium feature that requires an additional license fee.

### ***Example -- Intelligent Cropping***

The following code segment will first bleach any black border to match the surround white border, and it will then remove the white border, making the image as small as possible, registering it to the top and left of the page. For this example, we will be linking the ScanFix control to the TMSFile control:

```
' link the ScanFix control to a file control
ScanFix1.ImageDataSource = TMSFile1.Link

' set InputFrom to memory, since the file
' controls will load the image from disk
ScanFix1.InputFrom = 0

' set the input file name
TMSFile1.InputFileName = "c:\newfiles\tiff0001.tif"

< Continued >

' set OutputTo to file to use the control's
```

```

' file I/O libraries
ScanFix1.OutputTo = 1

' set the OutputFileName
ScanFix1.OutputFileName = "c:\processed\img_001.tif"

' enable the Intelligent Cropping
ScanFix1.CropBlack = True
ScanFix1.CropWhite = True

' begin processing
ScanFix1.ScanFix

```

---

## Orientation and Rotation

ScanFix offers both manual and automatic image orientation options. The manual options are included with the standard ScanFix package, while the automatic options are available at an additional fee.

The standard options allow you to rotate the image in 90 increments -- 0°, 90°, 180°, 270°. This rotation may be applied either before or after all other processing is performed.

The automatic options analyze the image and determine its current orientation, correcting it to upright or portrait, or allowing you to return the image to its original orientation after processing is complete.

### Orientation and Rotation Properties

Standard orientation control is performed through two properties:

TurnAfter

Enumerated property that will rotate the image to the specified orientation *after* processing is performed. The degree of rotation is based on the input image, not necessarily on a portrait image. The valid options for this property are as follows:

Property Value	Rotation
0	No rotation
1	90°
2	180°
3	270°

TurnBefore

Enumerated property that will rotate the image to the specified orientation *before* any other processing is

performed. The degree of rotation is based on the input image, not necessarily on a portrait image. The valid options for this property are as follows:

Property Value	Rotation
0	No rotation
1	90°
2	180°
3	270°

Automatic orientation control is performed through three additional properties:

AutoRotatePortrait	If True, the image will be automatically rotated to portrait orientation. This means only that the long side of the image will be vertical, which may be upside-down.
AutoRotateRevert	If True, the image will be returned to its original orientation after processing is complete.
AutoRotateUpright	If True, the image will be automatically rotated so that the image is upright. Unlike <b>AutoRotatePortrait</b> , this property will always output upright images.

### ***Example -- Orientation and Rotation***

The following routine will read image data from an Image Display Window and rotate it so that all output images are upright. . In this example, we are linking the ScanFix control to the file control. It will also receive and send information to this control:

```

' link the ScanFix control to a file control for
' input and to another file control for output
ScanFix1.ImageDataSource = TMSFile1.Link
TMSFile2.ImageDataSource = ScanFix1.Link

' set InputFrom and OutputTo to memory,
' since the file controls will load and save the
' image to disk
ScanFix1.InputFrom = 0

                                < Continued >

ScanFix1.OutputTo = 0

' set the input file name

```

```

TMSFile1.InputFileName = "c:\newfiles\tiff0001.tif"

' activate the Automatic Upright feature
ScanFix1.AutoRotateUpright = True

' begin processing
ScanFix1.ScanFix

```

---

## The OptionString Property

The **OptionString** property permits the user to add custom processing options to the ScanFix process. This version of ScanFix has several additional features which are not otherwise accessible. These features are controlled through **OptionString** parameters passed to the underlying ScanFix DLL's, as specified below.

### OptionString Syntax

When setting the **OptionString** property, the general format is **AAAA=#** or **AAAA=xx**, where **AAAA** is the option name in capital letters, **#** is a numeric parameter to be passed to the option, and **xx** is a string of characters (which may be random) which cause the option string to evaluate as ON:

```

SFix1.OptionString = "BATCHMODE=ON"

or

SFix1.OptionString = "SFTIME=20 SFWHITE=20"

```

If you need to add options to the **OptionString** property during run time, you could add an option with the code

```

SFix1.OptionString = SFix1.OptionString + " APPEND=ON"

```

If you are adding options this way, it is vital to include a space before the new option. It is also important that you do not include spaces around the equal sign. Removing a single option from an existing **OptionString** at run time requires that you repopulate the property with a string excluding that particular option. If you are setting the **OptionString** property at design time in the Properties window, multiple string options may be entered to the property, separated by spaces and/or commas.

## Processing Mode Options

### ***BATCHMODE=ON***

This option adjusts how ScanFix responds to fatal errors when processing multiple files. With this feature enabled, ScanFix will continue to the next file.

### ***SFSAFE=ON***

This option causes ScanFix to load and free the library between each image. If per chance ScanFix has failed to clean up memory between images, this will allow Windows to do a 'perfect' clean up job. The result is that ScanFix will be slightly slower but slightly more robust when processing batches of images.

### ***SFYIELD=ON***

This option causes ScanFix to frequently yield control of the operating system. This causes ScanFix to run slightly slower, but makes it useful as a background batch processor.

## Image Cropping Options

The cropping options detailed here are available only when the Intelligent Cropping features have been purchased. Intelligent Cropping is a premium feature, available for an additional license fee.

### ***SFBLACK=#***

This option allows you to adjust the amount of noise avoidance that Black crop will use. The number indicates the thickness in pixels of white space that will be considered as noise. Because ScanFix has other 'edge to info' detection algorithms, a rather large number (up to 100 or so) can usually be used without damaging text. The default value is 20. This option is available only if the premium feature Intelligent Cropping has been purchased.

### ***SFHCROP=#***

This option allows you to request only horizontal black cropping (removing black from the left and right edge of the image.) It also allows you to set a different noise avoidance value for horizontal cropping as for vertical. Noise avoidance values are discussed in SFBLACK=# above. This option is available only if the premium feature Intelligent Cropping has been purchased.



## **SFVCROP=#**

This option, which is only available if you have purchased the Cropping option, allows you to request only vertical black cropping (removing black from the top and bottom of the image.) It also allows you to set a different noise avoidance value for horizontal cropping as for vertical. Noise avoidance values are discussed in SFBLACK=# above. This option is available only if the premium feature Intelligent Cropping has been purchased.

## **SFWHITE=#[,#]**

The first # indicates the amount of noise to ignore. The default value of 10 causes ScanFix to ignore noise that is less than 10 pixels in either direction (approximately). The second # (optional, as indicated by the []'s) indicates which planes to crop. The plane values are 1 = TOP, 2 = RIGHT, 4 = BOTTOM, 8 = LEFT. To get multiple planes add the plane values together. The default value of 15 means do all planes.

Note that registration happens after white crop, so it is frequently more efficient to use the value of 6 ( RIGHT + BOTTOM ). Further, right and left edge detection are SIGNIFICANTLY slower than top and bottom detection, so you may want to avoid using them in speed-critical applications.

So, to avoid heavier noise during white crop, use string option:

```
SFix1.OptionString = "SFWHITE=20"
```

And to avoid less noise, while just cropping the top and bottom of the image, not the left and right edge, use string option:

```
SFix1.OptionString = "SFWHITE=5,5"
```

This option is available only if the premium feature Intelligent Cropping has been purchased.

## **SF\_RMBLOB=#[,#]**

This feature locates large black objects in your document after the black cropping has completed. It is useful if the edge of your document is too noisy to be effectively dealt with using the **CropBlack** property and SFBLACK parameter. This feature finds large black objects anywhere in your image, and removes them.

The first number indicates the number of pixels of black in a contiguous field of black that is the minimum to remove. High numbers are usual. 65535 is the highest permitted number. The second number indicates the required black density, defined as the number of pixels of black compared with the number of pixels in the bounding box of the blob. The default for the second number is 50 (50%).

```
SFix1.OptionString = "SF_RMBLOB=2250"
```

will remove all blobs with 2250 or more black pixels (1/2" \* 1/2") where the blob has at least 50% density (a 2250 pixel blob does not extend over an area greater than 1/2" \* 1".)

```
SFix1.OptionString = "SF_RMBLOB=2250,25"
```

will remove blobs with 2250 or more pixels, but will remove 2250 pixel blobs which extend over a 1" \* 1" area. This option is available only if the premium feature Intelligent Cropping has been purchased.

## AutoRotate Specific Options

The rotation options detailed here are available only when the AutoRotation features have been purchased. AutoRotation is a premium feature, not included in the standard version of ScanFix, that is available for an additional license fee.

### **SFPORTRAIT=#**

When you enable the **AutoRotatePortrait** property, it seeks 20 indicators that an image is a portrait image. If it finds those 20 indicators, it concludes that the image is actually a portrait image. This option is available only if the premium feature AutoRotation has been purchased.

If it fails to find 20, it then rotates the image and seeks 20 indicators that the rotated aspect is correct. If it can't find 20 indicators in the rotated image, it uses the indicator count to decide which aspect is the most 'portrait'. The # parameter affects the number of indicators required for a quick decision by the Auto Portrait algorithm. If you set the number to a smaller value such as 10, it will conclude that the image is portrait without checking the rotated perspective if it finds only 10 indicators.

If the parameter is very large, say 1000, it will almost always rotate the image and check for indicators in the rotated perspective. This, of course, is slower. If it fails to establish the desired aspect at such a setting, then the algorithm never will. You may wish to talk to Sequoia Data about such an image to seek a custom solution.

### **SFUPRIGHT=#**

When you enable the **AutoRotateUpright** property, ScanFix seeks image characteristics that indicate whether an image is upside down or right side up. The indicators that it seeks are: the envelope of paragraphs (paragraphs are usually Completion\_Smooth on the left and rough on the right.), and the ascenders and descenders of lower case print (There are many more ascenders than descenders in lower case print.). Setting the # to a higher value causes ScanFix to require a more clear determination of upsidedownness before it will invert an image. If it does not

get that indication, it will report an UNSURE message. If you set # to a smaller value, it may conclude that an image is upside down with very little evidence, and may get it wrong. This option is available only if the premium feature AutoRotation has been purchased.

## Debugging Options

### ***ENTRYDB=ON***

This option causes the ScanFix engine to report all of the parameters as it sees them. This is particularly useful for cross-checking whether string options are being implemented correctly. It may be requested by our technical support engineers as a reliable way of assuring what parameters are set. It also reports the link date of your ScanFix engine.

### ***SFNOCHECK=ON***

Normally the ScanFix engine cross-checks all string options against a valid table, reporting misspelled string options etc. Setting this string option causes ScanFix to avoid this check, improving speed slightly, and reducing warnings on undocumented custom string options.

### ***SFTIME=#***

Set # to the number of seconds maximum allocated to process an image, and if ScanFix cannot enhance an image in that amount of time, it will give up with a message. This is useful for rejecting unmanageably complex images in batch processing.

### ***SF\_IMGINFO=ON***

If SF\_IMGINFO is enabled, then ScanFix will report the image width, height and horizontal and vertical resolution to the Message event in the following format:

```
IMAGE SIZE=width,height RES=horizontal,vertical
```

This expresses the established image size after all ScanFix enhancement is complete. Note that the horizontal and vertical resolution reports can be useful for determining if an image is a low or high resolution fax.

## Deskew Control Options

### **~~SFFACTOR=#~~**

The default value for SFFACTOR is image resolution dependent, and is 70 on 300 dpi images. If your document has small dense print, and fails to establish the image's skew, especially if it has vertical lines, try a smaller number such as 40 or 20. If your document has only very large print, and ScanFix fails to establish the image's skew, use a higher number. Try 200, 400, or 500. If you use a very small number such as 1 or 2, then ScanFix will require solid horizontal lines for skew detection.

### **~~SF\_SEESKEW=ON~~**

If this string option is set, after the skew is reported to the Message event in the usual fashion: SKEW: 1 in 23, the skew will not be corrected. This is useful if you want to know the skew but don't want to correct it.

### **~~SF\_SKEWTO=#~~**

SF\_SKEWTO causes the image to be rotated to the amount stated in #. The value (#) is the same as that reported by the skew detector, one pixel rotation per # pixels horizontal. Negative values produce a counterclockwise rotation. Note that this happens after any automatic skew correction has been done.

## Dot Shading Options

### **~~SFSHADE\_AFTER=ON~~**

This option forces shade detection to occur late in the enhancement process, after line detection and isolated despeck removal. This feature can be priceless when shade is surrounded by lines.

### **~~SF\_DENSITY=#~~**

This string option allows for an adjustment of the shaded zone detection algorithm. If your images have very sparse shading (the shade dots are much smaller than the gaps between dots), set # to a small negative number. If your images have very dense shading (the shade dots are much larger than the gaps between dots, or the dots are touching each other), set # to a small positive number. Numbers between -5 and 5 can be useful.

### ***SF\_DOTVERIFY=#***

The dot shading remover does a 'dot count' analysis which assures a zone is actually dot shaded. This is especially valuable in small shaded zones and areas of small print, but may be invasive, especially with fine line shading. It also may affect speed slightly. Normally all shaded zones are checked for dot count. If SF\_DOTVERIFY is set, however, the larger zones are not counted. If any zone is taller (involves more scan lines) than #, then it is not counted. If SF\_DOTVERIFY=0 is set, dot verification will be fully disabled.

## **Fax Management Options**

### ***SF\_DOUBLE=#***

Causes the image size to be doubled. If # is 1, then it will double the number of scan lines (height) but will not double the image width. This is effective for simulating high resolution faxes from low resolution faxes. If the # is 2, then the width of the image is doubled, but the height of the image remains the same. If # is 3 (or any other value), then the width and height of the image is doubled. This is often useful for improving OCR results with very small print.

The doubling happens late in the image enhancement product and does not change the stated resolution of the image. I.E.: if a 300 dpi image was read in, the written image will still state 300 dpi with this feature active.

### ***SFTIFF\_F=ON***

This option biases tiff output to conform to the TIFF class F (fax) file standard. Note that this feature does not assure that the image widths and heights will conform to the standard. Other facilities within ScanFix (such as the SUB IMAGE feature, and the SIZE\_SAMEstring option) can assist you in this regard.

### ***SF\_FAXHDR=#***

This option exempts the top scan lines of your document from all enhancing. This was designed specifically for the headers which are commonly found on the top of faxed documents. The value of # indicates the number of scan lines at the top of the document that are to be treated as fax header.

### ***SF\_FAXHDR=I#***

Does as above but assumes that the top # 100ths of an inch are fax header. We included the latter so that high and low resolution images could be passed without regard for their resolution. The capital I stands for inches.

## File Reading Options

### **SET\_HRES=#**

Causes ScanFix to treat an image as having the horizontal resolution stated in #. This is done prior to any enhancement, and the supplied resolution is written to any file.

### **SET\_RES=#**

Causes ScanFix to treat an image as having the resolution stated in #. This is done prior to any enhancement, and the supplied resolution is written to any file.

### **SET\_VRES=#**

Causes ScanFix to treat an image as having the vertical resolution stated in #. This is done prior to any enhancement, and the supplied resolution is written to any file.

### **SFPAGE=#**

Causes the #'th page of a multi-page tiff file to be enhanced. ScanFix numbers the pages in a multi-page file starting at 1, just as the Image Display Window does. To use this option, **InputFrom** should be set to 1--File.

## File Writing Options

The File Writing Options are case sensitive, and do not use an option name followed by a '=' as other string options do. To use two or more, enter each separately, and separate them by a space:

```
SFix1.OptionString = " -t -3"
```

Note: ScanFix will output a great number of different files -- including file types that cannot be read by ImageBASIC. If you want to be able to read the output file, make sure that it is a compatible format.

File Writing Option Strings are as follows:

- h            output file in Hayes JTFax format
- l            output file in Laserdata's Group 4 format
- t            output file in tiff format (defaults to Group 4 compressed)
- p            output file in Calera's .PDA Group 4 format
- r            output file in Sequoia Data's runlength format (Pixtrans RLE)
- s            output file in Sequoia Data's pixelmap format

-v	output file in Viewstar format
-x	output file in PCX format
-1	output tiff format files uncompressed
-2	output tiff files using type 2 (modified Group 3) compression
-3	output tiff files using type 3 (true Group 3) compression
-4	output tiff files using type 4 (group 4) compression
-I	output tiff files using 1 line strips (Usually undesirable)
-S	output tiff format files using 32 line strips
-N	output tiff format files using no strips (Most common)
-F	output tiff format files using image order 1
-f	output tiff format files using image order 2
-i	output tiff format files using the INTEL file type
-m	output tiff format files using MOTOROLA file type

NOTE: ScanFix will output the same file format as it read in, unless otherwise instructed by the above flags. Unless a TIFF format was read, if TIFF format is being output it will be Group 4, Unstripped, Image order 1, and Intel file type. The only exceptions are due to responses to string options. ScanFix can read the formats that it can write, so the above list indicates what formats it can read.

### ***APPEND=ON***

Causes the output tiff image to be appended to any existing tiff image when it is written, producing or extending a multi-page tiff image.

### ***INVERTCOLOR=ON***

Causes TIFF format files to output with the opposite color rendition from what ScanFix normally produces. Various programs interpret the color rendition tag (Photometric Interpretation) differently. This option allows ScanFix to produce files that work well with any of these programs.

### ***SF\_NOSAVE=ON***

This option causes ScanFix to produce no output when **OutputTo** is set to 1 (OutputFileName.)

### ***SF\_NOTAG=ON***

This option causes ScanFix to output tiff files without placing its mark (that it has enhanced the image) in the tiff header. If the file read had ScanFix's mark in it, this will cause that mark to be removed.

### ***SF\_NOWORK=ON***

This option causes ScanFix to produce no output unless there was some enhancement done to the file. It is useful to increase the speed of processing, also as a way of indicating those files which needed no enhancement (their output file will not exist if they were clean.)

### ***SF\_ONCE=ON***

This option causes ScanFix to refuse to read tiff files which have ScanFix's mark in them. (See SF\_NOTAG option above.) This is useful to assure that you are not doubling your work.

### ***SIZE\_MAX=#***

This option causes ScanFix to assure that the width and height of a file will not exceed the pixel count supplied in #. This is useful for assuring that file sizes will not exceed another program's capacities. The minimum acceptable value for this option is 1000.

### ***SIZE\_MAXH=#***

This option is like SIZE\_MAX=# above, but only limits the height of the file.

### ***SIZE\_MAXW=#***

This option is like SIZE\_MAX=# above, but only limits the width of the file.

### ***SIZE\_SAME=ON***

This option causes ScanFix to remember the original pixel size of the input file, and cause the output file to be the same pixel size. This function can cause data loss, and causes unusual interactions with the **SubImageActivateProperty**.

### ***SIZE\_SAMEH=ON***

This option is like SIZE\_SAME=ON except that it only affects the height of the image.



### ***SIZE\_SAME~~W~~=ON***

This option is like SIZE\_SAME=ON except that it only affects the width of the image.

## **Inverse Text Management Options**

### ***SFINV\_DENSITY=#***

This string option allows control of the invert manager's analysis of what ratio of white to black should be found in an image. The value passed represents the number of white pixels per 100 black pixels above which the inverter decides that the zone is not an invert zone. The default is 200. Use a higher value to detect lighter zones.

## **Other String Options**

### ***SFSPECK\_AFTER~~ON~~***

Enabling this string option causes horizontal and vertical noise removal to occur very late in analysis. It will happen after isolated noise removal, and even after dot shading detection if SFSHADE\_AFTER is on.

## **Registration Control Options**

### ***SFREG\_GAP=#***

This string option controls the amount of white space that may be found between bits of edge noise. A larger number will cause more data to be recognized as edge noise. The default value of 5 works well in most cases.

### ***SFREG\_MIN=#***

SFREG\_MIN adjusts the noise threshold for registration. If you want the registration activity to be more willing to treat image activity as noise (and ignore it for registration) try a higher value. The default is 1, which is maximally sensitive. If our register is adjusting your margins to noise rather than to the image, and if register to a line is not a solution, setting this parameter to a number greater than 1 is frequently useful.

### ***SFREG\_WIDTH=#***

Adjusts the width of consistent activity (in scan lines) which indicates the edge of information. If there is significant activity for less than # of vertical scan lines, the

activity is considered to be random noise. The default is 5. Values around 20 (for a 300 dpi image) frequently work well and consistently ignore perpendicular lines because individual characters are usually about 20 pixels wide. Smaller values down to 1 are recommended when the image edge is dot matrix print.

### ***SF\_SEEREG=ON***

If enabled, image registration will not be adjusted, only detected and reported to the **Message** event. The report: 'REGISTERING # & #' indicate the number of pixels adjustment would have been made to the top and left of the image respectively.

## **SubImage Control Options**

### ***RE\_SIZE=#,#,#,#***

Similar to the **SubImagePad** property but uses relative parameters. The first # is for the top, the 2nd is for the right, the 3rd is for the bottom, and the 4th is the left side of the image. A positive number broadens the image in the stated direction. A negative number removes image in the stated direction.

```
SFix1.OptionString = "RE_SIZE=100,0,0,0"
```

This places a 100 pixel white pad on the top of the image.

# Chapter 3: ScanFix Control Reference

---

## Properties, Methods and Events

### **AboutBox Method**

<b>Definition:</b>	Displays a message box containing the name of the control, a copyright message and an OK button. Pressing the button will unload the message box.
<b>Parameters:</b>	None
<b>Syntax:</b>	<code>ScanFix1.AboutBox</code>
<b>Return Value:</b>	None
<b>Comments:</b>	This message box provides information about the ScanFix control.

### **Active Property**

<b>Definition:</b>	<p>If set to True at design time, the control will fully initialize and verify licensing immediately upon initialization of the runtime application.</p> <p>If set to False at design time, full initialization of the control will be delayed at initialization of the runtime application. In this case, this property must be explicitly set to True at runtime before the control is used.</p>
<b>Data Type:</b>	Boolean
<b>Design Access:</b>	Read/Write
<b>Runtime Access:</b>	Read/Write (see limits below)
<b>Comments:</b>	<p>If this property is set to True (the default) at design time, the control is fully initialized and licensing is verified immediately upon initialization of the application at runtime. The related technology libraries are loaded and the control is ready to be used.</p> <p>If this property is set to False at design time, the control will only partially initialize when the application loads at runtime. By delaying these two actions, the application should be able to load more quickly:</p> <ol style="list-style-type: none"><li>1) The related technology libraries for the control will not be loaded.</li><li>2) The licensing server will not verify an available token for the control.</li></ol>

If the control initializes with Active set to False, this property must be explicitly set to True by the application. Until Active is set to True, the control will ignore all instructions to it.

If the control fails to find a license token, the Active property will be automatically set to False. The application can check this value on Form Load to determine if each control is licensed and can be used.

### ***AutoRotatePortrait Property***

- Definition:** If True, the image that is being enhanced will be automatically rotated to portrait. The page may remain upside-down. Available only as part of the Premium AutoRotation option.
- Data Type:** Boolean
- Design Access:** Read/Write
- Runtime Access:** Read-only
- Possible Values:** True  
False
- See Also:** AutoRotateRevert Property, AutoRotateUpright Property
- Comments:** This property can also be configured through the **OptionString** property. When you enable the **AutoRotatePortrait** property, it seeks 20 indicators that an image is a portrait image. If it finds those 20 indicators, it concludes that the image is actually a portrait image. If it fails to find 20, it then rotates the image and seeks 20 indicators that the rotated aspect is correct. If it can not find 20 indicators in the rotated image, it uses the indicator count to decide which aspect is the most 'portrait'.
- The # parameter in the **OptionString** affects the number of indicators required for a quick decision by the Auto Portrait algorithm. If you set the number to a smaller value such as 10, it will conclude that the image is portrait without checking the rotated perspective if it finds 10 indicators.

### ***AutoRotateRevert Property***

<b>Definition:</b>	Reverts the image to its original orientation after all other processing.
<b>Data Type:</b>	Boolean
<b>Design Access:</b>	Read/Write
<b>Runtime Access:</b>	Read-only
<b>Possible Values:</b>	True False
<b>See Also:</b>	AutoRotatePortrait Property, AutoRotateUpright Property
<b>Comments:</b>	This is a premium property, available as a part of the Automatic Rotation option. This activity is can also be configured through the <b>OptionString</b> property.

### ***AutoRotateUpright Property***

<b>Definition:</b>	If True during image enhancement, all incoming images will be rotated to upright before any other enhancement.
<b>Data Type:</b>	Boolean
<b>Design Access:</b>	Read/Write
<b>Runtime Access:</b>	Read-only
<b>Possible Values:</b>	True False
<b>See Also:</b>	AutoRotateRevert Property, AutoRotatePortrait Property
<b>Comments:</b>	This is a premium property, available as a part of the Automatic Rotation option. The determination of upright is based largely on the relative abundance of ascenders on letters. The <b>OptionString</b> Property can be used to configure this determination.

## ***CompletionCustom Property***

**Definition:** Allows for fine control over the "growth" and "erosion" of an image. It grows or erodes an image vertically and horizontally.

**Data Type:** String

**Design Access:** Read/Write

**Runtime Access:** Read-only

**See Also:** CompletionErode Property, CompletionGrow Property

**Comments:** The image is processed in the same order as this property is set. The following table shows how to set this property:

H#	Grows the image by # horizontally
V#	Grows the image by # vertically
h#	Erodes the image by # horizontally
v#	Erodes the image by # vertically

Example:

```
ScanFix1.CompletionCustom = H3, V3, h0, v0
```

## ***CompletionErode Property***

**Definition:** Specifies the number of pixels to remove from the outside border of all image objects.

**Data Type:** Integer

**Design Access:** Read/Write

**Runtime Access:** Read-only

**Default Value** 0

**See Also:** CompletionGrow Property, CompletionSmooth Property

**Comments:** If set to a positive non-zero integer, ScanFix will remove the specified number of pixels from each uniquely identifiable image object. Image objects are generally letters and lines, but also include specks of random noise and dots on i's. Erosion is performed in both the vertical and horizontal planes, so that a dot that is 3 by 3 pixels will be reduced to a single pixel with this property set to two (2).

### ***CompletionGrow Property***

<b>Definition:</b>	Specifies the number of pixels to add to the outer border of all image objects.
<b>Data Type:</b>	Integer
<b>Design Access:</b>	Read/Write
<b>Runtime Access:</b>	Read-only
<b>Default Value</b>	0
<b>See Also:</b>	CompletionErode Property, CompletionSmooth Property, CompletionCustom Property
<b>Comments:</b>	If set to a positive integer, ScanFix will add the specified number of pixels to each uniquely identifiable image object. Image objects are generally letters and lines, but also include specks of random noise and dots on <i>i</i> 's. Erosion is performed in both the vertical and horizontal planes, so that a dot that is 3 by 3 pixels will be grown to 5 by 5 pixels with this property set to two (2).

### ***CompletionSmooth Property***

<b>Definition:</b>	Specifies the maximum number of pixels to add or remove in a "sand and fill" operation that is performed to smooth the surfaces of characters.
<b>Data Type:</b>	Integer
<b>Design Access:</b>	Read/Write
<b>Runtime Access:</b>	Read-only
<b>Default Value</b>	0
<b>See Also:</b>	CompletionErode Property, CompletionGrow Property
<b>Comments:</b>	If set to a positive integer, ScanFix will add or remove up to the specified number of pixels to each uniquely identifiable image object in an attempt to make the objects border as even as possible. This process is designed for use on dot matrix and other low resolution print. The resulting text is more smooth and gap-free, making it more OCR friendly. Values in this property from 1 to 5 generally offer the best results.

## ***CropBlack Property***

<b>Definition:</b>	If True during image enhancement, any black border on the image will be "bleached" to white.
<b>Data Type:</b>	Boolean
<b>Design Access:</b>	Read/Write
<b>Runtime Access:</b>	Read-only
<b>Possible Values:</b>	True False (Default)
<b>See Also:</b>	CropWhite Property
<b>Comments:</b>	The <b>CropBlack</b> property removes black borders from an image prior to deskew analysis. The related <b>OptionString</b> property sets the amount of white noise which will be ignored in black edge analysis.

## ***CropWhite Property***

<b>Definition:</b>	If True during image enhancement, any white border will be removed.
<b>Data Type:</b>	Boolean
<b>Design Access:</b>	Read/Write
<b>Runtime Access:</b>	Read-only
<b>Possible Values:</b>	True False (Default)
<b>See Also:</b>	CropBlack Property
<b>Comments:</b>	The removal of white borders from an image is performed prior to deskew analysis. The border that is removed is everything that is outside an upright rectangle that completely bounds the desirable image data. Removal of the border registers the image to the top and left and reduces its overall width and height.



### ***DeskewActivate Property***

<b>Definition:</b>	If True during image enhancement, Skew Detection and Correction is enabled.
<b>Data Type:</b>	Boolean
<b>Design Access:</b>	Read/Write
<b>Runtime Access:</b>	Read-only
<b>Possible Values:</b>	False (Default) True
<b>See Also:</b>	DeskewCharacterProtect Property, DeskewMaxAcceptableSkew Property, DeskewMinDetectLength Property
<b>Comments:</b>	Skew Detection and Correction is performed prior to registration and line removal, and is recommended for most applications.

### ***DeskewCharacterProtect Property***

<b>Definition:</b>	If True, minimizes the distortion of characters that is possible when correcting a large degree of skew.
<b>Data Type:</b>	Integer
<b>Design Access:</b>	Read/Write
<b>Runtime Access:</b>	Read-only
<b>Default Value</b>	0
<b>See Also:</b>	DeskewActivate Property
<b>Comments:</b>	Character protection is necessary only when the original image text is skewed over about 15%.

### ***DeskewMaxAcceptableSkew Property***

<b>Definition:</b>	Specifies the horizontal run of text over which a single pixel of vertical skew is acceptable.
<b>Data Type:</b>	Integer
<b>Design Access:</b>	Read/Write
<b>Runtime Access:</b>	Read-only
<b>Default Value</b>	0
<b>See Also:</b>	DeskewActivate Property
<b>Comments:</b>	Larger values result in more precise deskewing. The default setting of 150 means that the resultant image will have no more than one pixel of vertical skew for each 150 pixels horizontally. This corresponds to 0.67% skew, and produces good OCR results. A setting of 100 corresponds to 1% skew; a setting of 200 corresponds to 0.5% skew. Set to lower values for higher speed.

### ***DeskewMinDetectLength Property***

<b>Definition:</b>	Specifies the minimum horizontal pixel length of a run of text or line used to detect skew.
<b>Data Type:</b>	Integer
<b>Design Access:</b>	Read/Write
<b>Runtime Access:</b>	Read-only
<b>Default Value</b>	0
<b>See Also:</b>	DeskewActivate Property
<b>Comments:</b>	When a length of text or a line is detected that exceeds this limit, the <b>DeskewMaxAcceptableSkew</b> property is consulted to determine whether the image should be straightened.

### ***DespeckHorizontal Property***

<b>Definition:</b>	Specifies the maximum width of an image object to remove.
<b>Data Type:</b>	Integer
<b>Design Access:</b>	Read/Write
<b>Runtime Access:</b>	Read-only
<b>See Also:</b>	DespeckVertical Property, DespeckIsolated Property
<b>Comments:</b>	This property limits only the horizontal dimension, so that a vertical line that is hundreds of pixels long will be removed if it is only one or two pixels wide. Values larger than 1 or 2 tend to degrade text and lines. Use negative values (-1 or -2) to despeck with character protection, an option that requires more time but can result in text that is more easily OCR'ed.

### ***DespeckIsolated Property***

<b>Definition:</b>	Specifies the maximum width and height, in pixels, of image objects to remove.
<b>Data Type:</b>	Integer
<b>Design Access:</b>	Read/Write
<b>Runtime Access:</b>	Read-only
<b>Default Value</b>	0
<b>See Also:</b>	DespeckHorizontal Property, DespeckVertical Property
<b>Comments:</b>	This property defines the maximum width and height of a clump of pixels that will be removed in the speck removal process. It differs from <b>DespeckHorizontal</b> and <b>DespeckVertical</b> in that it sets a single size limit for both dimension.

### ***DespeckVertical Property***

<b>Definition:</b>	Specifies the maximum height of image objects to remove.
--------------------	--

<b>Data Type:</b>	Integer
<b>Design Access:</b>	Read/Write
<b>Runtime Access:</b>	Read-only
<b>Default Value</b>	None
<b>See Also:</b>	DespeckHorizontal Property, DespeckIsolated Property
<b>Comments:</b>	This property limits only the vertical dimension, so that a horizontal line that is hundreds of pixels long will be removed if it is only one or two pixels tall. Values larger than 1 or 2 tend to degrade text and lines. Use negative values (-1 or -2) to despeck with character protection, an option that requires more time but can result in text that is more easily OCR'ed.

### ***DotShadingActivate Property***

<b>Definition:</b>	If True, Dot Shading Detection and Removal is enabled during image enhancement.
<b>Data Type:</b>	Enumerated
<b>Design Access:</b>	Read/Write
<b>Runtime Access:</b>	Read-only
<b>Possible Values:</b>	False (Default) True
<b>See Also:</b>	DotShadingHeight Property, DotShadingHorzAdjust Property, DotShadingMaxSpeckSize Property, DotShadingProtect Property, DotShadingReport Property, DotShadingWidth Property, DotShadingVertAdjust Property
<b>Comments:</b>	If this property is True before the image enhancement action is set, all areas on an image that are dot shaded and match the parameters listed under 'See Also' will be removed. Dot shading removal is one of the slowest enhancement options; therefore, this options should be used only when necessary if total throughput must be maximized.

### ***DotShadingHeight Property***

<b>Definition:</b>	Determines the smallest vertical height of an expected shaded zone to detect.
<b>Data Type:</b>	Integer
<b>Design Access:</b>	Read/Write
<b>Runtime Access:</b>	Read-only
<b>See Also:</b>	DotShadingActivate Property
<b>Comments:</b>	Set lower to detect thinner shaded zones. Set higher to reduce false detects.

### ***DotShadingHorzAdjust Property***

<b>Definition:</b>	Specifies the number of pixels to add to the width of the automatic dot size detection. Enables the removal of dots that would be left by the unaltered size detection algorithm.
<b>Data Type:</b>	Integer
<b>Design Access:</b>	Read/Write
<b>Runtime Access:</b>	Read-only
<b>Possible Values:</b>	0
<b>See Also:</b>	DotShadingActivate Property
<b>Comments:</b>	Set to 1 if some specks survive; this will cause the automated speck remover to remove dots 1 pixel wider than it measured automatically. Set to 2 to remove dots 2 pixels wider. Set to -1 or -2 if text is being degraded (leaving horizontal white streaks). This will decrease the width of the specks being removed by 1 or 2 pixels, and will protect small text. The default of 0 works best in most cases.

### ***DotShadingMaxSpeckSize Property***

<b>Definition:</b>	Specifies the maximum width of individual specks comprising shaded area to be removed, measured in pixels.
<b>Data Type:</b>	Integer
<b>Design Access:</b>	Read/Write
<b>Runtime Access:</b>	Read-only
<b>Default Value</b>	0
<b>See Also:</b>	DotShadingActivate Property
<b>Comments:</b>	If a dot shaded area is found that is comprised of dots larger than specified in this property, it will not be removed.

### ***DotShadingProtect Property***

<b>Definition:</b>	Protects the characters in a dot shaded region from degradation that might occur during enhancement.
<b>Data Type:</b>	Integer
<b>Design Access:</b>	Read/Write
<b>Runtime Access:</b>	Read-only
<b>Default Value:</b>	0
<b>See Also:</b>	DotShadingActivate Property
<b>Comments:</b>	<p>This enumerated property has these options:</p> <ul style="list-style-type: none"><li>1 protect in the horizontal plane</li><li>2 protect in the vertical plane</li><li>3 protect in both horizontal and vertical planes</li></ul> <p>Because dot shading removal may remove characters which touch the dots in a dot shaded region, it is frequently necessary to turn on the character feature. If this feature is used, negative numbers should be used in the <b>DotShadingHorzAdjust</b> and <b>DotShadingVertAdjust</b> properties.</p>

### ***DotShadingReport Property***

<b>Definition:</b>	Enables the reporting of the coordinates of dot shaded regions.
<b>Data Type:</b>	Enumerated
<b>Design Access:</b>	Read/Write
<b>Runtime Access:</b>	Read-only
<b>Possible Values:</b>	0--Omit (Default) 1--Report
<b>See Also:</b>	DotShadingActivate Property
<b>Comments:</b>	<p>If this property is set to <i>1--Report</i>, ScanFix reports the pixel locations of dot shaded regions in the image. The report is supplied through the <i>Message</i> parameter of the <b>Message</b> Event. The report has the following format:</p> <p>SHADE: LINES 22 to 99, COLUMNS 21 to 106, SIZE 4</p>

### ***DotShadingVertAdjust Property***

<b>Definition:</b>	Specifies the number of pixels to add to or remove from the calculated dot size.
<b>Data Type:</b>	Integer
<b>Design Access:</b>	Read/Write
<b>Runtime Access:</b>	Read-only
<b>Default Value:</b>	0
<b>See Also:</b>	DotShadingActivate Property
<b>Comments:</b>	Set to 1 or 2 if some specks survive the de-specking process. Set to -1 or -2 if text is being degraded (leaving vertical white streaks). The default of 0 works best.

### ***DotShadingWidth Property***

<b>Definition:</b>	Specifies the minimum width that a dot shaded region must reach to be removed during enhancement.
<b>Data Type:</b>	Integer
<b>Design Access:</b>	Read/Write
<b>Runtime Access:</b>	Read-only
<b>Default Value:</b>	100
<b>See Also:</b>	DotShadingActivate Property
<b>Comments:</b>	Any dot shaded region that is not at least as many pixels wide as specified in this property will not be removed during enhancement. If dot matrix text is being interpreted as a dot shaded region, set this property to a higher value. If small dot shaded regions are being ignored, set to a lower value.

## **Error Event**

<b>Definition:</b>	Occurs for each error internal to the control.	
<b>Parameters:</b>	Number	A long error code that identifies the error
	Description	Descriptive string of the error
	SCode	A composite long number indicating the severity of the error, the facility code, the origin of the error, and the native error code
	Source	Descriptive string of the source of the error
	HelpFile	Suggested help file name that should have a detailed explanation of the error
	HelpContext	Context ID in the help file
	CancelDisplay	If set to True during this event, the standard error dialog will not be displayed
<b>Description:</b>	Any time an error occurs inside the TMS File control, the <b>Error</b> event is triggered.	

## **HorzLineActivate Property**

<b>Definition:</b>	If True, Horizontal Line Detection and Removal is enabled during enhancement.
<b>Data Type:</b>	Boolean
<b>Design Access:</b>	Read/Write
<b>Runtime Access:</b>	Read-only
<b>Possible Values:</b>	True (Default) False
<b>See Also:</b>	HorzLineClean Property, HorzLineMaxGap Property, HorzLineMaxWidth Property, HorzLineMinLength Property, HorzLineReconstruct Property, HorzLineReport Property
<b>Comments:</b>	The properties shown under 'See Also' set parameters to the line removal process. Vertical line removal is enabled separately through the <b>VertLineActivate</b> property.

### ***HorzLineClean Property***

<b>Definition:</b>	Specifies the distance in pixels that the engine will search on both sides of removed lines for isolated specks that will also be removed.
<b>Data Type:</b>	Integer
<b>Design Access:</b>	Read/Write
<b>Runtime Access:</b>	Read-only
<b>Default Value:</b>	2
<b>See Also:</b>	HorzLineActivate Property
<b>Comments:</b>	This property is especially useful for removing poor-quality lines. The edge cleaning factor determines how far from the line ScanFix will clean up noise. Set it to 3 or 4 if line remnants remain, and to 1 or 0 if adjacent text is being degraded.

### ***HorzLineMaxGap Property***

<b>Definition:</b>	Specifies the maximum gap, measured in pixels, that can be present in line. Any larger gap is considered the separation between two individual lines.
<b>Data Type:</b>	Integer
<b>Design Access:</b>	Read/Write
<b>Runtime Access:</b>	Read-only
<b>Default Value:</b>	2
<b>See Also:</b>	HorzLineActivate Property
<b>Comments:</b>	If the lines that are being removed are broken or low resolution, it is likely that all of the segments will not reach the length specified in the HorzLineMinLength property. Those line segments will therefore not be removed. By setting the HorzLineMaxGap property to accommodate the gaps in the lines, the entire line length can be removed.

### ***HorzLineMaxWidth Property***

<b>Definition:</b>	Specifies the maximum thickness, in pixels, of lines to remove.
<b>Data Type:</b>	Integer
<b>Design Access:</b>	Read/Write
<b>Runtime Access:</b>	Read-only
<b>Default Value:</b>	22
<b>See Also:</b>	HorzLine_Activate Property
<b>Comments:</b>	Some capital letters such as the letter T in a large title or headline have a vertical component which could be mistaken for a line and removed. To avoid erroneous line removal, set this property to a value smaller than the letter's line width.



### ***HorzLineMinLength Property***

<b>Definition:</b>	Specifies the minimum length of line to locate, report and remove; measured in pixels.
<b>Data Type:</b>	Integer
<b>Design Access:</b>	Read/Write
<b>Runtime Access:</b>	Read-only
<b>Default Value:</b>	150
<b>See Also:</b>	HorzLineActivate Property
<b>Comments:</b>	In order for any line to be removed, it must reach at least the length specified in this property. Be certain to set this property to a value larger than the length of any letter segments, such as  may be found in a large letter <b>L</b> . If lines are not being found and removed, refer to the <b>HorzLineMaxGap</b> property for assistance.

### ***HorzLineReconstruct Property***

<b>Definition:</b>	Specifies the pixel width of the largest character that will be repaired after an intersecting line is removed.
<b>Data Type:</b>	Integer
<b>Design Access:</b>	Read/Write
<b>Runtime Access:</b>	Read-only
<b>Default Value:</b>	20
<b>See Also:</b>	HorzLineActivate Property
<b>Comments:</b>	This property sets the maximum pixel width of characters to be repaired after line removal. The default works well in most cases. If text is not being repaired, these values can be set higher, but they should be reduced if erroneous reconstruction is occurring. Set to 0 to disable reconstruction. In this situation, <b>HorzLineClean</b> property should be set lower than the default.

## ***HorzLineReport Property***

**Definition:** Enables and disables the reporting of horizontal line detection and removal. This property is ignored if **HorzLineActivate** is False.

**Data Type:** Enumerated

**Design Access:** Read/Write

**Runtime Access:** Read-only

**Possible Values:** 0--Omit Report (Default)  
1--Locate & Remove  
2--Locate Only

**Comments:** If set to *1--Locate & Remove*, the location of all horizontal lines will be reported in the **Message** event through the *Message* parameter. The report has the following format:

DOWN: 322, OVER: 110, HORZ: 2150, THICK: 8

In the above example, ScanFix reported a horizontal line 2150 pixels long and 8 pixels thick with one endpoint located at 322 pixels from the top and 110 pixels from the left edge of the image.

Valid options for this property are as follow:

*0--Omit*

This is the default value. When set to 0, a report will not be generated but horizontal lines will be removed.

*1--Locate & Remove*

When set to this value, the report will specify both the line and removal location.

*2--Locate Only*

This value will specify only the location of the line.

### ***HorzRegActivate Property***

<b>Definition:</b>	If True, enables registration to the left edge of an image during enhancement.
<b>Data Type:</b>	Boolean
<b>Design Access:</b>	Read/Write
<b>Runtime Access:</b>	Read-only
<b>Possible Values:</b>	False (Default) True
<b>See Also:</b>	Action Property, HorzRegAddOnly Property, HorzRegCentralFocus Property, HorzRegFindLine Property, HorzRegIgnoreHoles Property, HorzRegLeftMargin Property
<b>Comments:</b>	All of the properties listed above under 'See Also' set parameters to the horizontal registration of the image, and, therefore, must be set before starting enhancement.

### ***HorzRegAddOnly Property***

<b>Definition:</b>	If True, any existing margin will not be reduced during registration, regardless of the setting in <b>HorzRegLeftMargin</b> .
<b>Data Type:</b>	Enumerated
<b>Design Access:</b>	Read/Write
<b>Runtime Access:</b>	Read-only
<b>Possible Values:</b>	False (Default) True
<b>See Also:</b>	HorzRegActivate Property
<b>Comments:</b>	This property assures that no part of the image is inadvertently lost. However, if this property is True, the image will not be properly registered if it was offset to the right.

### ***HorzRegCentralFocus Property***

<b>Definition:</b>	If True, ScanFix determines the edge of text or a line to register to by beginning its search in the middle of the page.
<b>Data Type:</b>	Enumerated
<b>Design Access:</b>	Read/Write
<b>Runtime Access:</b>	Read-only
<b>Possible Values:</b>	False (Default) True
<b>See Also:</b>	HorzRegActivate Property
<b>Comments:</b>	With this property enabled, ScanFix generally ignores letterhead, page numbers, logos, footnotes, etc. when determining the edge of the image that will be used for registration.

### ***HorzRegFindLine Property***

<b>Definition:</b>	If set to a positive, non-zero integer, ScanFix will register to the left most line that is at least this many pixels long. If left at the default of zero (0), ScanFix will attempt to determine the edge of the image's text and register to that border.
<b>Data Type:</b>	Integer
<b>Design Access:</b>	Read/Write
<b>Runtime Access:</b>	Read-only
<b>Default Value</b>	0
<b>See Also:</b>	HorzRegActivate Property
<b>Comments:</b>	ScanFix can use either a vertical line or the edge of the text to register. In either case, the entire image is shifted to create a margin of the size specified in the <b>HorzRegLeftMargin</b> property.

### ***HorzRegIgnoreHoles Property***

<b>Definition:</b>	If True, ScanFix ignores binder holes and other left-edge noise in analyzing the left margin.
<b>Data Type:</b>	Enumerated
<b>Design Access:</b>	Read/Write
<b>Runtime Access:</b>	Read-only
<b>Possible Values:</b>	False (Default) True
<b>See Also:</b>	HorzRegActivate Property
<b>Comments:</b>	The ScanFix engine has been trained to recognize the typical appearance of binder holes and can be instructed to ignore them when it is attempting to identify the edge of the image data that will be used for registration.

### ***HorzRegLeftMargin Property***

<b>Definition:</b>	Specifies the size, in pixels, of the left margin that will remain in the document after registration.
<b>Data Type:</b>	Integer
<b>Design Access:</b>	Read/Write
<b>Runtime Access:</b>	Read-only
<b>Default Value</b>	0
<b>See Also:</b>	HorzRegActivate Property, HorzRegAddOnly Property
<b>Comments:</b>	This margin is the distance left between the left edge of the text or the line used for registration and the left edge of the image. If the <b>VertRegAddOnly</b> property is enabled, the resulting margin may be greater than is specified in <b>HorzRegLeftMargin</b> . A value below 30 are not recommended, as image text may be lost.

### ***ImageDataSource Property***

<b>Definition:</b>	Specifies the Link Id of the ImageBASIC control that will supply image data to this control.
<b>Data Type:</b>	String
<b>D/T Access:</b>	Read/Write
<b>R/T Access:</b>	Read/Write
<b>See Also:</b>	ImageDataChanged Event, InputFrom Property
<b>Description:</b>	This property may be set to the name of any ImageBASIC control that is capable of outputting image data from the ScanFix control. This property will only be used if the InputFrom property is set to 0. Example: <code>ScanFix1.ImageDataSource = TMSDisp1.Link</code>

### ***InputFileName Property***

<b>Definition:</b>	If <b>InputFrom</b> is set to <i>1--InputFileName</i> , this property specifies the fully qualified file name of the image to enhance.
<b>Data Type:</b>	String
<b>Design Access:</b>	Read/Write
<b>Runtime Access:</b>	Read-only
<b>Default Value</b>	None
<b>See Also:</b>	InputFrom Property, ImageDataSource Property
<b>Comments:</b>	This property is only used if the <b>InputFrom</b> is a file. If writing the output to file, note that the input file cannot be directly overwritten as it remains open during enhancement.

### ***InputFrom Property***

<b>Definition:</b>	Specifies the source of image data for enhancement.
<b>Data Type:</b>	Enumerated
<b>Design Access:</b>	Read/Write
<b>Runtime Access:</b>	Read-only
<b>Possible Values:</b>	0--Memory 1--File
<b>See Also:</b>	InputFileName Property, OutputTo Property
<b>Comments:</b>	<i>0--Memory</i> If the <b>InputFrom</b> property is set to <i>0--Memory</i> , the image supplied from the control named in <b>ImageDataSource</b> will be processed. The enhanced image will be sent to the destination specified in the <b>OutputTo</b> property.

### *1--File*

If set to *1--File*, the **InputFileName** property must be set to the full path and file name of the image file that you wish to process.

## ***InverseActivate Property***

<b>Definition:</b>	If True, Inverse Text Correction is enabled. If False, Inverse Text Correction is disabled.
<b>Data Type:</b>	Enumerated
<b>Design Access:</b>	Read/Write
<b>Runtime Access:</b>	Read-only
<b>Possible Values:</b>	0--Off (Default) 1--On
<b>See Also:</b>	InverseEdge Property, InverseHeight Property, InverseReport Property, InverseWidth Property
<b>Comments:</b>	<p>Inverse Text Correction automatically detects inverse text (white-on-black) and converts it into normal text (black-on-white). It will process multiple zones of inverse text on the page. The ScanFix engine will also recognize and correct inverse zones of different shapes, such as rectangles, circles, ovals, banners, etc.</p> <p>This feature of ScanFix is most useful because most OCR engines presently cannot read inverse text. By correcting the source of the difficulty, ScanFix now makes it possible.</p>

## ***InverseEdge Property***

<b>Definition:</b>	Specifies the minimum horizontal distance between the edges of the inverse zone and the inverse text.
<b>Data Type:</b>	Integer
<b>Design Access:</b>	Read/Write
<b>Runtime Access:</b>	Read-only
<b>Default Value</b>	10
<b>See Also:</b>	InverseActivate Property
<b>Comments:</b>	Indicates the minimum horizontal length of black pixels along the left and right edges of the inverse zone. Decrease this value if inverse bands are missed and increase if text is treated as inverse.

### ***InverseHeight Property***

<b>Definition:</b>	Specifies the pixel height of the smallest expected band of inverse text measured in pixels.
<b>Data Type:</b>	Integer
<b>Design Access:</b>	Read/Write
<b>Runtime Access:</b>	Read-only
<b>Default Value</b>	50
<b>See Also:</b>	InverseActivate Property
<b>Comments:</b>	Decrease this value if small inverse bands are missed; increase if text is treated as if inverted.

### ***InverseReport Property***

<b>Definition:</b>	If set to <i>1--Report</i> , the control reports the coordinates of all detected regions of inverted text.
<b>Data Type:</b>	Enumerated
<b>Design Access:</b>	Read/Write
<b>Runtime Access:</b>	Read-only
<b>Possible Values:</b>	0--Omit (Default) 1--Report
<b>See Also:</b>	InverseActivate Property
<b>Comments:</b>	The report is issued to the <i>Message</i> parameter of the <b>Message</b> event. The report has the following format: INVERT: LINES 22 to 118, COLUMNS 21 to 106

In this example, ScanFix reported a zone of inverse text starting 22 pixel lines down from the top of the image, extending to 118 pixel lines from the top, and 21 pixel columns from the left edge of the image, extending to 106 pixel columns from the left. Note that the inverse zone may be of any shape whatsoever, including rectangular, round, oval, banner, etc. ScanFix will report the location of the bounding rectangle of the zone.

## ***InverseWidth Property***

<b>Definition:</b>	Specifies the minimum pixel width of regions to consider as inverted.
<b>Data Type:</b>	Integer
<b>Design Access:</b>	Read/Write
<b>Runtime Access:</b>	Read-only
<b>Default Value</b>	500
<b>See Also:</b>	InverseActivate Property
<b>Comments:</b>	ScanFix will search for a horizontal black line or area of at least this width. Decrease this value if small inverse bands are being missed, and increase it if text is being treated as inverted.

## ***Message Event***

<b>Definition:</b>	This even is called frequently during the processing operations of ScanFix to relay information from the ScanFix report properties.	
<b>Parameters</b>	Message	Descriptive string of the process being performed
	MsgType	Integer process identification code.
<b>Comments:</b>	If any of the following properties are set to enable reporting, the information will be reported in the <i>Message</i> parameter to this event. DotShadingReport HorzLineReport InverseReport VertLineReport	

## ***OptionString Property***

<b>Definition:</b>	Permits the minor adjustment of many of the enhancement features.
<b>Data Type:</b>	String
<b>Design Access:</b>	Read/Write
<b>Runtime Access:</b>	Read-only
<b>Default Value</b>	None
<b>Comments:</b>	The <b>OptionString</b> property permits the user to modify the standard enhancement options. The OptionString variables that apply to each enhancement option are listed with the appropriate enabling property (e.g., HorzLineActivate or CropBlack).

## ***OutputFileName Property***

<b>Definition:</b>	Specifies the destination name for the enhanced image file. Used only when the <b>OutputTo</b> property is set to <i>1--OutputFileName</i> .
--------------------	--



<b>Data Type:</b>	String
<b>Design Access:</b>	Read/Write
<b>Runtime Access:</b>	Read-only
<b>Default Value</b>	None
<b>See Also:</b>	OutputTo Property
<b>Comments:</b>	The fully qualified path for the destination file must be specified. The file specified here is written only if OutputTo is set to <i>--OutputFileName</i> . Note that the input file and output file cannot be the same because of access conflicts.

### ***OutputTo Property***

<b>Definition:</b>	Specifies where the image data will be output after enhancement.
<b>Data Type:</b>	Enumerated
<b>Design Access:</b>	Read/Write
<b>Runtime Access:</b>	Read-only
<b>Possible Values:</b>	0--Memory 1--File
<b>See Also:</b>	OutputFileName Property
<b>Comments:</b>	<p><i>0--Memory</i></p> <p>When the <b>OutputTo</b> property is set to this value, the processed image will be supplied to any ImageBASIC control whose <b>ImageDataSource</b> property specifies this ScanFix control.</p> <p><i>1--File</i></p> <p>If the <b>OutputTo</b> property is set to <i>1--File</i>, the <b>OutputFileName</b> property must be set to the full path and file name where you wish to store the image.</p>

### ***PeriodRemovalExpectedFrequency Property***

- Definition:** If set to a positive integer, Dot Leader Detection and Removal is enabled during enhancement. Specifies the minimum expected gap, in pixels, between consecutive periods in a run.
- Data Type:** Integer
- Design Access:** Read/Write
- Runtime Access:** Read-only
- Default Value** 0
- Comments:** A ten (10) pixel spacing is typical. Use larger values if the periods are widely spaced and lower values if they are especially close together. Set to 0 to disable the removal of runs of periods.

### ***PeriodRemovalMaxHeight Property***

- Definition:** Specifies the maximum expected height, in pixels, of each dot in a run of periods.
- Data Type:** Integer
- Design Access:** Read/Write
- Runtime Access:** Read-only
- Default Value** 0
- Comments:** Start with 6 pixels, and use larger values for larger periods. Set to 0 to disable the removal of runs of periods.

### ***ScanFix Method***

- Definition:** Begins image enhancement processing.
- Parameters:** None
- Comments:** Before this method can be called, all of the enhancement parameter properties must be set. The InputFrom, OutputTo properties must also be set before calling the ScanFix method.

### ***SubImageActivate Property***

<b>Definition:</b>	If True, SubImage Processing is enabled during enhancement.
<b>Data Type:</b>	Enumerated
<b>Design Access:</b>	Read/Write
<b>Runtime Access:</b>	Read-only
<b>Possible Values:</b>	False (Default) True
<b>See Also:</b>	SubImageBottomEdge Property, SubImageLeftEdge Property, SubImagePad Property, SubImageRightEdge Property, SubImageTopEdge Property
<b>Comments:</b>	The Deskew and Register features should be set to True when splitting out a subimage to assure accuracy and consistent output. The extraction of the subimage is performed after registration and skew correction, but before other processing.

### ***SubImageBottomEdge Property***

<b>Definition:</b>	Specifies the bottom edge of the subimage to separate the top of the image.
<b>Data Type:</b>	Integer
<b>Design Access:</b>	Read/Write
<b>Runtime Access:</b>	Read-only
<b>Default Value</b>	0
<b>See Also:</b>	SubImageActivate Property
<b>Comments:</b>	Coordinates are all measured from the top, left corner of the image (0,0) to the bottom, right corner (ImageWidth, ImageHeight). Set to zero (0) to specify the full image height.

### ***SubImageLeftEdge Property***

<b>Definition:</b>	Specifies the distance, in pixels, from the left edge of the image to the left edge of the subimage.
<b>Data Type:</b>	Integer
<b>Design Access:</b>	Read/Write
<b>Runtime Access:</b>	Read-only
<b>Default Value</b>	0
<b>See Also:</b>	SubImageActivate Property
<b>Comments:</b>	A value of 0 in this property indicates that the subimage begins at the very edge of the image. A value of 1200 (4 in at 300 dpi) will start in the middle of a typical 8.5" X 11" page at 300 DPI.

### ***SubImagePad Property***

<b>Definition:</b>	If set to a positive, integer value, and <b>SubImage_Activate</b> is True, ScanFix will add a white border of the specified pixel width around all four sides of the extracted region.
<b>Data Type:</b>	Integer
<b>Design Access:</b>	Read/Write
<b>Runtime Access:</b>	Read-only
<b>Default Value</b>	0
<b>See Also:</b>	SubImageActivate Property
<b>Comments:</b>	This border will be drawn around all four sides of the subimage defined using the <b>SubImageActivate</b> parameters.

### ***SubImageRightEdge Property***

<b>Definition:</b>	Specifies the right edge the subimage to separate from the right edge of the image; measured in pixels.
<b>Data Type:</b>	Integer
<b>Design Access:</b>	Read/Write
<b>Runtime Access:</b>	Read-only
<b>Default Value</b>	0
<b>See Also:</b>	SubImageActivate Property
<b>Comments:</b>	Coordinates are all measured from the top, left corner of the image (0,0) to the bottom, right corner (ImageWidth, ImageHeight). Set to zero (0) to specify the full image width.

### ***SubImageTopEdge Property***

<b>Definition:</b>	Specifies the top edge of a subimage from the top of the image.
<b>Data Type:</b>	Integer
<b>Design Access:</b>	Read/Write
<b>Runtime Access:</b>	Read-only
<b>Default Value</b>	0
<b>See Also:</b>	SubImageActivate Property
<b>Comments:</b>	Coordinates are all measured from the top, left corner of the image (0,0) to the bottom, right corner (ImageWidth, ImageHeight).

### ***TurnAfter Property***

**Definition:** Enhanced images are rotated to the specified orientation after all other processing is complete.

**Data Type:** Enumerated

**Design Access:** Read/Write

**Runtime Access:** Read-only

**Possible Values:**

<b>Value</b>	<b>Description</b>
0--OFF (Default)	No action
1--90°	Rotates the image 90 clockwise
2--180°	Rotates the image 180 clockwise
3--270°	Rotates the image 270 counter-clockwise
4--Invert Image	Inverts the white and black text

### ***TurnBefore Property***

**Definition:** Rotates the image before performing any image enhancement options.

**Data Type:** Enumerated

**Design Access:** Read/Write

**Runtime Access:** Read-only

<b>Possible Values</b>	<b>Description</b>
0--OFF (Default)	No action
1--90°	Rotates the image 90 clockwise
2--180°	Rotates the image 180 clockwise
3--270°	Rotates the image 270 counter-clockwise
4--Invert Image	Inverts the white and black text

### ***VertLineActivate Property***

**Definition:** Enables a reports which describes the location and length of lines. This property also enables the vertical line management parameters.

**Data Type:** Enumerated

**Design Access:** Read/Write

**Runtime Access:** Read-only

**Possible Values:** False (Default)  
True

**See Also:** VertLineClean Property, VertLineMaxGap Property, VertLineMaxWidth Property, VertLineMinLength Property, VertLineReconstruct Property, VertLineReport Property

**Comments:** If this property is True before the image enhancement action is set, all vertical lines that match the parameters listed under 'See Also' will be removed or reported.

### ***VertLineClean Property***

**Definition:** Cleans-up noise near a line when removing it.

**Data Type:** Integer

**Design Access:** Read/Write

**Runtime Access:** Read-only

**Default Value** 2

**See Also:** VertLineActivate Property

**Comments:** This property is especially useful for removing poor-quality lines. The Edge Cleaning Factor determines how far from the line ScanFix will clean up noise. The default value is 2 pixels. Set it to 3 or 4 if line remnants remain, and to 1 or 0 if adjacent text is being degraded.

### ***VertLineMaxGap Property***

<b>Definition:</b>	Specifies the maximum gap allowed, measured in pixels.
<b>Data Type:</b>	Integer
<b>Design Access:</b>	Read/Write
<b>Runtime Access:</b>	Read-only
<b>Default Value</b>	1
<b>See Also:</b>	VertLineActivate Property
<b>Comments:</b>	Setting this value to 2 or 3 allows removal of lines in poorer-quality images. For extremely poor-quality images, such as dot-matrix and microfilm documents, you may want to set this value as high as 6. However, if the value is set too high, portions of text may be removed as well.

### ***VertLineMaxWidth Property***

<b>Definition:</b>	Specifies the maximum thickness of a line to be detected.
<b>Data Type:</b>	Integer
<b>Design Access:</b>	Read/Write
<b>Runtime Access:</b>	Read-only
<b>Default Value</b>	20
<b>See Also:</b>	VertLineActivate Property
<b>Comments:</b>	Some capital letters such as <b>T</b> in a large title or headline, have a vertical component which could be mistaken for a line and removed. To avoid erroneous line removal, set the Minimum Line To Detect parameter larger than the largest text.

### ***VertLineMinLength Property***

<b>Definition:</b>	Specifies the minimum length of a line, measured in pixels, to locate.
<b>Data Type:</b>	Integer
<b>Design Access:</b>	Read/Write
<b>Runtime Access:</b>	Read-only
<b>Default Value</b>	125
<b>See Also:</b>	VertLineActivate Property
<b>Comments:</b>	In order to be removed or reported, all vertical lines must reach at least this length. If the lines have gaps in them larger than specified in VertLineMaxGap, the line segments will be considered separate lines and may not reach the length specified in this property.

## ***VertLineReconstruct Property***

<b>Definition:</b>	Specifies the vertical dimensions of the largest character that is to be repaired after an intersecting line is removed.
<b>Data Type:</b>	Integer
<b>Design Access:</b>	Read/Write
<b>Runtime Access:</b>	Read-only
<b>Default Value</b>	20
<b>See Also:</b>	VertLineActivate
<b>Comments:</b>	Sets the maximum pixel width/height of characters to be repaired after line removal. The default value works well in most cases. If text is not being repaired, these values can be set higher, but they should be reduced if erroneous reconstruction is occurring. <b>VertLineReconstruct</b> can also be set to 0 to disable reconstruction; in that case, set the <b>VertLineClean</b> property lower than the default.

## ***VertLineReport Property***

<b>Definition:</b>	Reports the location of a line and its removal. Removal is optional.
<b>Data Type:</b>	Enumerated
<b>Design Access:</b>	Read/Write
<b>Runtime Access:</b>	Read-only
<b>Possible Values:</b>	
<b>Value</b>	<b>Description</b>
0--Omit	A report will not be generated
1--Locate & Remove	A report will specify both the line and removal location.
2--Locate Only	A reports specifies the location of the line.
<b>See Also:</b>	VertLineActivate
<b>Comments:</b>	In the following example, ScanFix reports a vertical line 2150 pixels high and 8 pixels thick with one endpoint located at 322 pixels from the top and 110 pixels from the left edge of the image. The report has the following format: DOWN: 322, OVER: 110, VERT: 2150, THICK: 8  The report is available through the <i>Message</i> parameter of the <b>Message</b> event.



### ***VertRegActivate Property***

<b>Definition:</b>	If True, enables the Vertical Line Management parameter properties, including a report that describes the location and length of vertical lines.
<b>Data Type:</b>	Boolean
<b>Design Access:</b>	Read/Write
<b>Runtime Access:</b>	Read-only
<b>Possible Values:</b>	True (Default) False
<b>See Also:</b>	VertRegAddOnly Property, VertRegCentralFocus Property, VertRegFindLine Property, VertRegTopMargin Property
<b>Comments:</b>	<b>VertLineActivate</b> must be set to True in order for the vertical line management parameters, listed under 'See Also,' to be activated.

### ***VertRegAddOnly Property***

<b>Definition:</b>	If True, restricts ScanFix to only adding to any existing top margin.
<b>Data Type:</b>	Enumerated
<b>Design Access:</b>	Read/Write
<b>Runtime Access:</b>	Read-only
<b>Possible Values:</b>	False (Default) True
<b>See Also:</b>	VertRegActivate Property, VertRegTopMargin Property
<b>Comments:</b>	By enabling this property it assures that no part of an image is inadvertently lost. If the image is too far from the edge of the page it will not be properly registered.

### ***VertRegCentralFocus Property***

<b>Definition:</b>	If True, ScanFix searches from the center of the page, rather than the top of the page, when determining a registration point.
<b>Data Type:</b>	Enumerated
<b>Design Access:</b>	Read/Write
<b>Runtime Access:</b>	Read-only
<b>Possible Values:</b>	False (Default) True
<b>See Also:</b>	VertRegActivate Property
<b>Comments:</b>	Under normal circumstances, ScanFix begins its search for a line or the top edge of the text at the top of the page. If this property is True, the search will begin in the center of the page, possibly ignoring letterhead, hole punches, etc.

### ***VertRegFindLine Property***

<b>Definition:</b>	Specifies the minimum length of a line, measured in pixels, that will be used for registration.
<b>Data Type:</b>	Integer
<b>Design Access:</b>	Read/Write
<b>Runtime Access:</b>	Read-only
<b>Default Value</b>	0
<b>See Also:</b>	VertRegActivate Property
<b>Comments:</b>	If this property is left at the default of zero (0), the control will attempt to determine the top edge of the text as the point to register to. If a positive integer value is specified for this property, the control will attempt to find a horizontal line at least this many pixels long and will register to that line.

### ***VertRegTopMargin Property***

<b>Definition:</b>	Specifies the size of the left or upper margin, measured in pixels, that will remain in the document after correction.
<b>Data Type:</b>	Integer
<b>Design Access:</b>	Read/Write
<b>Runtime Access:</b>	Read-only
<b>Default Value</b>	150
<b>See Also:</b>	VertRegActivate Property
<b>Comments:</b>	A value of 150 pixels corresponds to a 1/2 inch margin at 300 dpi; 300 to a 1 inch margin. Values below 30 are not recommended, as image text may be lost. If you lose a logo at the top of a page, deactivate <b>HorzRegCentralFocus</b> or increase the vertical margin.





# Index

## A

- AboutBox Method 55
- Active Property 55
- APPEND OptionString 51
- AutoRotate Specific Options 46
- AutoRotatePortrait 46
- AutoRotatePortrait Property 42, 56
- AutoRotateRevert Property 42, 57
- AutoRotateUpright 46
- AutoRotateUpright Property 42, 57

## B

- BATCHMODE OptionString 44
- Border Removal 6

## C

- Character Smoothing 6
- CompletionCustom Property 37, 58
- CompletionErode Property 37, 58
- CompletionGrow Property 37, 59
- CompletionSmooth Property 59
- Correcting Inverted Text 5
- CropBlack Property 60
- CropWhite Property 60

## D

- Debugging Options 47
- Deskew Control Options 48
- DeskewActivate Property 4, 30, 61
- DeskewCharacterProtect Property 30, 61
- DeskewMaxAcceptableSkew Property 4, 30, 61
- DeskewMinDetectLength Properties 4
- DeskewMinDetectLength Property 30, 62
- DespeckHorizontal Property 5, 11, 62
- DespeckIsolated Property 5, 11, 62
- DespeckVertical Property 11, 62
- Detecting Lines 24
- Document Registration 4
- Dot Shading Options 48
- DotShadingActivate Property 5, 14, 63
- DotShadingHeight Property 14, 64

- DotShadingHorzAdjust Property 14, 64
- DotShadingMaxSpeckSize Property 14, 64
- DotShadingProtect Property 14, 65
- DotShadingReport Property 14, 65
- DotShadingVertAdjust Property 14, 66
- DotShadingWidth Property 14, 66

## E

- Enhancing multi-page TIFFs 50, 51
- ENTRYDB OptionString 47
- Error Event 67
- Event
  - Message 76
- Events
  - Error 67

## F

- Fax Management Options 49
- File Reading Options 50
- File Writing Options 50

## H

- Horizontal and Vertical Registration 5, 19
- HorzLine\_Clean Property 69
- HorzLineActivate Property 5, 24, 67
- HorzLineClean Property 24, 68
- HorzLineMaxGap Property 24, 68
- HorzLineMaxWidth Property 24, 68
- HorzLineMinLength Property 25, 69
- HorzLineReconstruct Property 25, 69
- HorzLineReport Property 25, 70
- HorzRegActivate 19
- HorzRegActivate Property 4, 71
- HorzRegAddOnly Property 19, 71
- HorzRegCentralFocus Property 19, 22, 71
- HorzRegFindLine Property 19, 72
- HorzRegIgnoreHoles Property 20, 72
- HorzRegLeftMargin Property 20, 72

## I

- Image Cropping Options 44
- Image Orientation and Rotation 41
- ImageDataSource Property 73

- InputFileName Property 3, 8, 73
- InputFrom Property 3, 8, 73
- Intelligent Cropping 6
- Inverse Text Management Options 53
- Inverse Type Detection 5
- InverseActivate Property 5, 7, 8, 16, 74
- InverseEdge Property 7, 8, 17, 74
- InverseHeight Property 17, 75
- InverseReport Property 7, 8, 17, 75
- InverseWidth Property 17, 76
- INVERTCOLOR OptionString 51
- Inverted Text 5

## L

- Line Detection and Removal 5, 24
- Linking Controls
  - ImageDataSource Property 73

## M

- Message Event 76

## O

- OptionString 50
  - AutoRotate Specific Options 46
  - Debugging Options 47
  - Deskew Control Options 48
  - Dot Shading Options 48
  - Fax Management Options 49
  - File Reading Options 50
  - File Writing Options 50
  - Image Cropping Options 44
  - Inverse Text Management Options 53
  - Other String Options 53
  - Processing Mode Options 44
  - Registration Control Options 53
  - SFSHADE\_AFTER 53
  - SIZE\_SAME 49
  - SubImage\_Activate Control Options 54
- OptionString Property 43, 60, 76
- OptionStrings
  - APPEND 51
  - BATCHMODE 44
  - ENTRYDB 47
  - INVERTCOLOR 51
  - RE\_SIZE 54
  - SET\_HRES 50
  - SET\_RES 50

- SET\_VRES 50
- SF\_DENSITY 48
- SF\_DOTVERIFY 49
- SF\_DOUBLE 49
- SF\_FAXHDR 49
- SF\_IMGINFO 47
- SF\_NOSAVE 51
- SF\_NOTAG 52
- SF\_NOWORK 52
- SF\_ONCE 52
- SF\_RMBLOB 45
- SF\_SEEREG 54
- SF\_SEESKEW 48
- SF\_SKEWTO 48
- SFBLACK 44
- SFFACTOR 48
- SFHCROP 44
- SFINV\_DENSITY 53
- SFNOCHECK 47
- SFPAGE 50
- SFPORTRAIT 46
- SFREG\_GAP 53
- SFREG\_MIN 53
- SFREG\_WIDTH 53
- SFSAFE 44
- SFSHADE\_AFTER 48
- SFSPECK\_AFTER 53
- SFTIFF\_F 49
- SFTIME 47
- SFUPRIGHT 46
- SFVCROP 45
- SFWHITE 45
- SFYIELD 44
- SIZE\_MAX 52
- SIZE\_MAXH 52
- SIZE\_MAXW 52
- SIZE\_SAME 52
- SIZE\_SAMEH 52
- SIZE\_SAMEW 53

- Other String Options 53
- Output File Type 50
- OutputFileName Property 3, 8, 76, 77
- OutputTo Property 3, 8, 77

## P

- PeriodRemovalExpectedFrequency Property
  - 35, 78
- PeriodRemovalMaxHeight Property 35, 78
- Premium Processing Options 6
- Processing Mode Options 44

## Programming Considerations 2

### Properties

- Active 55
- AutoRotatePortrait 42, 46, 56
- AutoRotateRevert 42, 57
- AutoRotateUpright 42, 46, 57
- CompletionCustom 37, 58
- CompletionErode 37, 58
- CompletionGrow 37, 59
- CompletionSmooth 59
- CropBlack 60
- CropWhite 60
- DeskewActivate 4, 30, 61
- DeskewCharacterProtect 30, 61
- DeskewMaxAcceptableSkew 4, 30, 61
- DeskewMinDetectLength 4, 30, 62
- DespeckHorizontal 5, 11, 62
- DespeckIsolated 5, 11, 62
- DespeckVertical 11, 62
- DotShadingActivate 5, 14, 63
- DotShadingHeight 14, 64
- DotShadingHorzAdjust 14, 64
- DotShadingMaxSpeckSize 14, 64
- DotShadingProtect 14, 65
- DotShadingReport 14, 65
- DotShadingVertAdjust 14, 66
- DotShadingWidth 14, 66
- HorzLine\_Clean 69
- HorzLineActivate 5, 24, 67
- HorzLineClean 24, 68
- HorzLineMaxGap 24, 68
- HorzLineMaxWidth 24, 68
- HorzLineMinLength 25, 69
- HorzLineReconstruct 25, 69
- HorzLineReport 25, 70
- HorzRegActivate 4, 19, 71
- HorzRegAddOnly 19, 71
- HorzRegCentralFocus 19, 22, 71
- HorzRegFindLine 19, 72
- HorzRegIgnoreHoles 20, 72
- HorzRegLeftMargin 20, 72
- ImageDataSource 73
- InputFileName 3, 8, 73
- InputFrom 3, 8, 73
- InverseActivate 5, 7, 8, 16, 74
- InverseEdge 7, 8, 17, 74
- InverseHeight 17, 75
- InverseReport 7, 8, 17, 75
- InverseWidth 17, 76
- OptionString 43, 60, 76
- OutputFileName 3, 8, 76, 77

OutputTo 3, 8, 77

PeriodRemovalExpectedFrequency 35, 78

PeriodRemovalMaxHeight 35, 78

SubImageActivate 5, 32, 79

SubImageBottomEdge 32, 79

SubImageLeftEdge 32, 79

SubImagePad 33, 80

SubImageRightEdge 33, 80

SubImageTopEdge 33, 80

TurnAfter 81

TurnBefore 81

VertLine\_Clean 84

VertLineActivate 5, 27, 81

VertLineClean 27, 82

VertLineMaxGap 27, 83

VertLineMaxWidth 27, 83

VertLineMinLength 27, 83

VertLineReconstruct 27, 84

VertLineReport 28, 84

VertRegActivate 4, 22, 85

VertRegAddOnly 22, 85

VertRegCentralFocus 85

VertRegFindLine 22, 86

VertRegTopMargin 22, 86

## R

RE\_SIZE OptionString 54

Registering documents 4

Registration 5, 19

Registration Control Options 53

Removing Dot Leaders 35

Removing Lines 24

Removing Runs of Periods 35

Rotation and Orientation 6

## S

### Saving Files

ImageDataSource Property 73

Saving Files--Output File Type 50

SET\_HRES OptionString 50

SET\_RES OptionString 50

SET\_VRES OptionString 50

SF\_DENSITY OptionString 48

SF\_DOTVERIFY OptionString 49

SF\_DOUBLE OptionString 49

SF\_FAXHDR OptionString 49

SF\_IMGINFO OptionString 47

SF\_NOSAVE OptionString 51

SF\_NOTAG OptionString 52  
 SF\_NOWORK OptionString 52  
 SF\_ONCE OptionString 52  
 SF\_RMBLOB OptionString 45  
 SF\_SEEREG OptionString 54  
 SF\_SEESKEW OptionString 48  
 SF\_SKEWTO OptionString 48  
 SFBLACK OptionString 44  
 SFFACTOR OptionString 48  
 SFHCROP OptionString 44  
 SFINV\_DENSITY OptionString 53  
 SFNOCHECK OptionString 47  
 SFPAGE OptionString 50  
 SFPORTRAIT OptionString 46  
 SFREG\_GAP OptionString 53  
 SFREG\_MIN OptionString 53  
 SFREG\_WIDTH OptionString 53  
 SFSAFE OptionString 44  
 SFSHADE\_AFTER 53  
 SFSHADE\_AFTER OptionString 48  
 SFSPECK\_AFTER OptionString 53  
 SFTIFF\_F OptionString 49  
 SFTIME OptionString 47  
 SFUPRIGHT OptionString 46  
 SFVCROP OptionString 45  
 SFWHITE OptionString 45  
 SFYIELD OptionString 44  
 SIZE\_MAX OptionString 52  
 SIZE\_MAXH OptionString 52  
 SIZE\_MAXW OptionString 52  
 SIZE\_SAME 49  
 SIZE\_SAME OptionString 52  
 SIZE\_SAMEH OptionString 52  
 SIZE\_SAMEW OptionString 53  
 Skew Detection and Correction 4  
 Speck Removal 11  
 Standard Processing Options 4  
 Sub Image Control Options 54  
 SubImageActivate 52  
 SubImageActivate Property 5, 32, 79  
 SubImageBottomEdge Property 32, 79  
 SubImageLeftEdge Property 32, 79  
 SubImagePad Property 33, 80  
 SubImageRightEdge Property 33, 80  
 SubImageTopEdge Property 33, 80  
 System Requirements 2

## T

TurnAfter Property 81  
 TurnBefore Property 81

## V

VertLine\_Clean Property 84  
 VertLineActivate Property 5, 27, 81  
 VertLineClean Property 27, 82  
 VertLineMaxGap Property 27, 83  
 VertLineMaxWidth Property 27, 83  
 VertLineMinLength Property 27, 83  
 VertLineReconstruct Property 27, 84  
 VertLineReport Property 28, 84  
 VertRegActivate 22  
 VertRegActivate Property 4, 85  
 VertRegAddOnly Property 22, 85  
 VertRegCentralFocus Property 85  
 VertRegFindLine Property 22, 86  
 VertRegTopMargin Property 22, 86