

Caere OCR ActiveX Control

User's Guide

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 are protected by one or more of the following copyright notices:

Portions of this product,© 1993 - 1997 Diamond Head Software, Inc. All rights reserved.

Portions of this product,© 1994 - 1996 Caere, Inc. All rights reserved.

Company and product names mentioned in this documentation are trademarks or registered trademarks of their respective companies. Lotus and Lotus Notes are registered trademarks of Lotus Development Corporation. 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 to Caere	1
Linking of ImageBASIC Controls.....	1
Licensing Configuration and Verification	3
Chapter 2 : Using the Caere Control	7
Overview of Using the Caere Control.....	7
Recognition of Image Text.....	7
Optimization of the Recognition Attempt.....	18
Preprocessing Image Enhancement.....	18
Character Type Selection.....	21
Using Language Lexicons.....	22
User-Defined Lexicons.....	24
Chapter 3 : Reference	27
Reference to Properties, Methods and Events.....	27
Index	47

Chapter 1 : Introduction to Caere

Linking of ImageBASIC Controls

Types of Inter-Control Communication

The ImageBASIC controls communicate amongst themselves in a unique manner to get information and data to the control that needs it. There are two times that this communication is performed:

- 1) Image data is passed, or
- 2) Services are performed

Each ImageBASIC control can be either a source or a recipient of one or both of these types of communication. The act of specifying the source and recipient of this communication is called *linking* the controls.

All ImageBASIC controls that can accept image data from another ImageBASIC control have a property called **ImageDataSource**. The **ImageDataSource** property specifies the ImageBASIC control that is the source for all incoming images.

- For example, the Caere OCR control cannot directly access image files on disk.
- The Pixel File control can read the image files into memory where they can be retrieved by the Display control.
- Therefore, the source of images for the Caere control is the Pixel File control. The communication between the controls is enabled by setting the Caere control's **ImageDataSource** to specify the Pixel File control.

Some ImageBASIC controls offer services other than image processing. The communication for these controls is enabled in a similar manner, but through properties other than the **ImageDataSource** property.

- For example, the Annotation control does not process image data and, therefore, does not have an **ImageDataSource** property.
- Instead, the Annotation control performs a service -- creating and maintaining annotations. The Annotation control must have a Display control on which to show its annotations.

- The Annotation control's **DisplaySource** property must specify the Display control that will be performing this function.

Linking at Design Time

As each ImageBASIC control is added to a Form at design time, its **ImageDataSource** property will be automatically set to an ImageBASIC control already on that Form if one exists.

The assignment may be changed at design time by selecting from the drop-down list of available ImageBASIC components. This list is shown with the **ImageDataSource** property in the Properties Window or Object Inspector. The same list of ImageBASIC controls is shown for the **DisplaySource** property, the **RegionSource** property, and all other...**Source** properties.

Linking at Runtime

As each instance of an ImageBASIC control is created, a unique identification string is calculated for that control. This string, or Link ID, is stored in the **Link** property of each control. This property is not visible at design time and cannot be changed by the application designer. The unique Link ID is calculated each time a control is created, whether it is created statically at design time or dynamically at runtime.

The assignment of a source control is made by setting the Source property of the receiving control -- perhaps the **ImageDataSource** of a Caere control -- to the **Link** property of the source control. For example, for a Caere control to accept image data from a TMS File control, set the **ImageDataSource** property as shown here for Visual Basic:

```
Caere1.ImageDataSource = TMSFile1.Link
```

Other forms of inter-control communication are established in the same manner. For example, allowing an Annotation control to display its objects on a Display control is performed by setting the Annotation control **DisplaySource** property as shown here:

```
Annot1.DisplaySource = TMSDispl1.Link
```

Licensing Configuration and Verification

In order to run, each ImageBASIC control must be able to verify the presence of a valid license token. These license tokens are stored on either a hardware key (shipped with each toolkit) or in a licensing database (the standard runtime distribution format).

Utilizing these tokens, licensing in ImageBASIC is based on enabling a set number of concurrent seats. Each license token allows a single PC to run any number of instances of a single control. For example, a single token for a Display will allow one workstation to concurrently run multiple applications, each of which employs any number of Display controls.

A unique token is required for each different type of ImageBASIC component that you have licensed:

- There is a special type of token for the Display control, another for the Caere control, another for the ScanFix control, and yet more token types for each additional control. The 16-bit and 32-bit versions of each control are also licensed separately.
- Each one of these licenses also comes in two varieties: *runtime* and *development*.

As suggested by the names, runtime licenses are necessary for an executable to function, and development licenses are necessary to develop an application using ImageBASIC.

A design time license will function as a runtime license, removing the need to add runtime tokens for application testing during development.

Where the Licenses Tokens are Kept

ImageBASIC will find licenses stored in either one of two locations -- in a licensing database or on a hardware key.

- The hardware key is plugged into a parallel port on the computer using ImageBASIC and will be automatically found each time an ImageBASIC component is used.
Note: When using Windows NT, the parallel port must be configured using the Sentinel drivers that are shipped with ImageBASIC.
- The licensing database must be created on the site where it will be used and may not be moved from the location in which it is installed.

The inability to move a licensing database is one of its protection features. This attachment to a single location is formed when the licensing database is first created. Although it may not be moved once

created, the licensing database can be written to a network drive to which all the machines running ImageBASIC have access.

A file called IMGBASIC.INI must be on each of these networked machines. This file contains an entry pointing to the location of the licensing database. ImageBASIC can now search the licensing database for an available copy of each license it needs to run.

The licensing database may also be created on a local drive, activating ImageBASIC on only that one machine. The same INI file is still necessary to pinpoint the location of the database.

How the License Tokens are Used

As each application that uses ImageBASIC is initiated, or the control is loaded into the development environment, the ImageBASIC licensing server attempts to find the proper token for each component.

For example, if an executable has been developed that uses ImageBASIC to display existing images, and then calls on Caere to perform OCR on that image, that application must be able to find one *Display runtime license*, one *File runtime license* and one *Caere runtime license*.

- If the application is successful in finding these licenses, it will load normally and function exactly as it was programmed.

When the application finds these licenses and is thereby informed that the correct licenses are available, it simultaneously locks the licenses so that no other application can use the same licenses to run at the same time.

If two copies of these same licenses are available, another computer will be able to run the same application and will then lock the second license.

- If the application cannot verify the presence of the required tokens, the ImageBASIC components will not operate.

The **Active** property of the controls that did not find license tokens will be set to False. The state of the **Active** property can be verified during Form Load.

A runtime error that can be trapped during Form Load will also occur.

Licensing Token Release

When an application that has locked one or more tokens terminates normally, the tokens are released and can immediately be taken by another user if a network licensing database is being used. This is the process by which concurrent licensing for any number of seats may be enabled.

- If the application ends abnormally -- the user might reboot or a concurrently running Windows application might lock up -- then the release of the licenses is conditional on the network or disk operating system.
- For Novell networks, the default time for releasing a file lock after a connection is lost is 5 (five) minutes.

For other networks, your system administrator should be able to tell you how long the NOS takes to release the lock.

The system administrator should be able to change the default release time to satisfy your particular needs.

- If the licensing database has been installed on a stand-alone machine, the locks are immediately released and will again be available when the application is started again.

Chapter 2 : Using the Caere Control

Overview of Using the Caere Control

This chapter will provide an in-depth discussion of the options available through ImageBASIC's Caere control and the interface to those options. The use of the control to optimize OCR and to perform the process are described here. Chapter 3 : 'Reference' on page 27 is a technical reference guide to the properties, method, and events of this control.

The main topics covered in this chapter are as follows:

- "Recognition of Image Text" on page 7 describes the selection of input image data and output parameters along with the capture of the output string.
- "Optimization of the Recognition Attempt" on page 18 covers all aspects of optimization available through this control including preprocessing enhancement, language lexicons and field type definition.

Recognition of Image Text

The steps required to perform OCR are listed below in "Performing OCR" with a brief description of each step. Details of the steps -- the selection of an image or image region to process, the setting of all preprocessing parameters including lexicon and field type specification, and the output of the recognition text -- are all described in the following sections.

The section titled 'Optimization of the Recognition Attempt' on page 18 provides a more thorough discussion of the enhancement options available through the control. The implementation of these options in the OCR process is introduced in the steps immediately below.

Performing OCR

The basic process of performing OCR using this control is as follows:

1) Specify a Source of Image Data

The Caere control must receive image data from another ImageBASIC control. The control that is supplying that image is specified in the **ImageDataSource** property, as illustrated below:

```
Caerel.ImageDataSource = TMSDispl.Link
```

2) Specify the Region(s) to Process

The Caere control can accept region coordinates from either one of two places:

- (a) Another control specified in the **RegionSource** property
- (b) Internal region definition

When the **OCRRegion** method is called, the Caere control will first query the **RegionSource** property:

If **RegionSource** specifies a valid source of region coordinates, then those coordinates will be applied to the image that is available from the **ImageDataSource**

If **RegionSource** is "<None>", the Caere control's own region definitions will be applied. Multiple regions can be defined using the **AddRegion** method as described

3) Set Processing Options

These options range from specifying the language of the document to the print style of the text in that document. Refer to **Optimization of the Recognition Attempt** on page 18 for details; some examples of setting processing options are as follows:

```
Caerel.Language = 2
Caerel.OptDotMatrix = True ' Expect dot matrix print
Caerel.OptRotation = 90 ' Degrees to rotate
```

4) Select the Output Format

Supported output formats range from plain ASCII text to many popular word processor formats and are specified through the **OutputFormat** property, as shown below:

```
Caerel.OutputFormat = 2
```

Note: When creating output in any format other than ASCII text, the output must be written directly to file.

The layout of the text for all text formats selected for the output (e.g., ASCII or word processor document) is specified through the **OutputStyle** property:

```
Caerel.OutputStyle = 1 ' Left Justified Only
```

5) Select the Output Destination

The **OutputTo** property specifies whether the recognition results are reported in the **Result** property or written directly to a disk file.

```
Caerel.OutputTo = 1 ' File
```

If the output is written to file, the **OutputFileName** property specifies the file name.

```
Caerel.OutputFileName = "c:\imagetext\file.wpd"
```

6) Begin Optical Character Recognition

OCR can be performed on either the entire document page or on a region of the page by executing either the **OCRPage** or **OCRRegion** method.

- If the **OutputTop** property is set to *0--String*, use either of the following techniques to capture the return string:

```
Dim sReturn as String
Dim nLoop as Integer
sReturn = Caerel.OCRPage
-- or --
Caerel.OCRRegion
For nLoop = 1 to Caerel.ResultCount
    Caerel.ResultIndex = 1
    sReturn(nLoop) = Caerel.Result
Next nLoop
```

- If the **OutputTop** property is set to *1--File*, call the method to start recognition, and the file specified in step 4), above, will be written when OCR is complete:

```
Caerel.OCRPage
```

- During the recognition process, two events may occur:

The **Error** event will occur if any terminal error in the recognition engine occurs.

The **Progress** event occurs frequently during recognition and reports the percentage of the region processed.

Specify a Source of Image Data

Using a Display Control as a Source of Image Data

The Caere control can accept image data only from another ImageBASIC control. This control cannot read directly from file. Therefore, the Caere control's **ImageDataSource** must specify another ImageBASIC control that can provide image data. If the image is to be displayed for operator manipulation or verification, the Display control may be selected, as shown below:

```
Dim sOCROutput As String
Caerel.ImageDataSource = TMSDispl.Link
```

```
TMSDispl.ImageDataSource = TMSFile1.Link
TMSFile1.InputFileName = "d:\images\1234.tif"
sOCROutput = Caere1.OCRRegion
```

Using a File Control as a Source of Image Data

In order to read image data from file and perform OCR without using a Display control, the Caere control must be linked to an ImageBASIC File control. In this case, all of the options and procedures for performing OCR with Caere are identical to performing OCR from a Display control, as illustrated below:

```
Caere1.ImageDataSource = TMSFile1.Link
TMSFile1.InputFileName = "d:\images\1234.tif"
sOCROutput = Caere1.OCRRegion
```

For an overview of configuring and starting a recognition attempt, see "Performing OCR" on page 7.

Define Region to Process

The Caere control can process one or more regions within the image in a single OCR attempt, or it can process the entire image as a single region. In either case, the image specified in the **ImageDataSource** property will be used.

If the **OCRPage** method is executed, the entire image will be processed without regard to any regions that are defined.

If the **OCRRegion** method is executed, the control will process only the region(s) found in the following locations:

- 1) First, the control will check the value of the **RegionSource** property. If **RegionSource** specifies a valid ImageBASIC control that can supply region coordinates, the region defined there will be used.
- 2) If the **RegionSource** property is empty, the control will use the regions defined internally.

Define Regions in the Caere Control

To define new regions within the Caere control, follow these steps:

- 1) Verify that no regions are currently defined:
 - a) The **RegCount** property should be 0 (zero)
 - b) If it is not, execute the **DeleteAllRegions** method to clear all currently defined regions:


```
Caere1.DeleteAllRegions
```

- 2) Execute the **AddRegion** method to allocate a new region, and then select the new region to update the region coordinate properties.

This method will allocate a new region and return the index to the new region:

```
intRegNumber = Caere1.AddRegion  
Caere1.RegIndex = intRegNumber
```

- 3) Set the region coordinates.

For example, your application may be allowing the user to define multiple regions for OCR by drawing the region with the mouse on an ImageBASIC Display control. In this case, the region coordinates in the Caere control might be set to the Working Region coordinates of the Display control:

```
Caere1.RegBottom = TMSDispl.RegBottom  
Caere1.RegLeft = TMSDispl.RegLeft  
Caere1.RegRight = TMSDispl.RegRight  
Caere1.RegTop = TMSDispl.RegTop
```

Set Processing Options

Processing options for the Caere control may be set before initialing recognition. The options range from the selection of a standard language lexicon to including a user-defined lexicon, and from selecting a character type (alphabetic or numeric) to a character size (typical fonts or small fonts). The following properties specify the application of the processing options:

ICRMode	Enumerated specification of the degree to which the engine will continue to attempt to read difficult characters. More diligence in attempting to recognize these characters can result in higher accuracy but at the cost of longer processing time.
Language	Enumerated specification of one of the predefined language lexicons. Refer to Using Language Lexicon§ on page22 for a complete list of the available languages.
LexicalClass	Specifies whether or not the text is a small font (4 to 6 points) and whether it is alphabetic or numeric. Refer to "Character Type Selection§ on page21 for details.
LexicalMode	Specifies how precisely recognized words must match the other field type and lexicon options.
LexiconFileName	Specifies the fully qualified path and file name to a user-defined lexicon.

OptAutoDeskew	<p>If True, the control will detect and attempt to correct image skew before further processing.</p> <p>If False, the image will be processed as presented.</p>
OptAutoPortrait	<p>If True, the control will rotate the incoming image to portrait orientation before attempting OCR. This option will allow the image to be rotated to upside-down.</p> <p>If False, the image will be processed as presented.</p>
OptAutoUpright	<p>If True, the control will attempt to determine the proper rotation for the image text to be upright and will rotate the image before attempting OCR.</p> <p>If False, the image will be processed as presented.</p>
OptDotMatrix	<p>If True, the control will attempt to compensate for the particular difficulties of processing dot matrix print. This processing is slower than normal OCR and should not be enabled unless dot matrix print may be presented.</p> <p>If False, the image will be processed using the normal character identification algorithms.</p>
OptFullOrientation	<p>If True, the control will attempt to determine both portrait orientation and upright orientation for the text of the image before beginning OCR.</p> <p>If False, the image will be processed as presented.</p>
OptRotation	<p>If True, the incoming image will be rotated to the specified orientation before beginning OCR. The valid options for this property are 0, 90, 180 and 270 degrees.</p> <p>If False, the image will be processed as presented.</p>
Pitch	<p>Specifies if the image text is composed of fixed width or not fixed width fonts. Fixed width fonts, such as Courier, use the same spacing for all characters, leaving extra white space on the sides of thin characters. These fonts are more accurately read because character separation is easier to determine.</p>
SpellCheck	<p>If True, the control will attempt to match all recognized words to its lexicons. Words that are found in the lexicon are assigned a higher confidence value. Recognized words with low confidence which almost match a word in the lexicon may be corrected to match the known word.</p>

Select the Output Format

Several different text formats can be output by the Caere control. These formats range from plain ANSI text to any of several popular word processor and spread sheet formats.

- When writing to any format other than plain text, the output results should be written directly to file because Visual Basic string variables and properties will not accept the binary data output that is produced as the other output formats.
- When writing recognition results to disk, the file will be created when recognition is complete.

Text Layout

The layout of the text, or the justification of the text, is specified through the **OutputStyle** property. This enumerated property specifies how the output paragraphs are justified. The options are as follows:

- 0 Best Fit Filled
- 1 Best Fit Not Filled
- 2 Stream

0--Best Fit Filled causes full justification of the output text, creating smooth edges on both the left and right sides of the text.

1--Best Fit Not Filled causes only left justification and the compression of all white space on the left of each line. The right edge of the paragraphs is left uneven.

2--Stream causes the text to be output with one space between each word and removes all blank lines. This option is primarily for use with full text indexing systems in which the layout of the image text is not important.

Output File Format

The Caere control can output many popular word processor formats in addition to standard ANSI text. The format of the output is specified through the enumerated property **OutputFormat** as illustrated here selecting RTF output:

```
Caere1.OutputFormat = 47
```

The complete list of options for this property is shown below.

Note: If any option other than ASCII is selected, the output must be written directly to file (see the **OutputTo** and **OutputFileName** properties).

- 7 Ami Pro 2.0

8 Ami Pro 3.0
9 ANSI
10 ASCII Database
11 ASCII DeColumnized
13 ASCII Standard (default value)
17 DCA/RTF
19 EBCDIC
21 Excel 3.0
22 Excel
23 Publisher
24 MS Word 4.0
25 MS Word 5.x
26 MS Word 6.0
27 MS Works
28 MultiMate 3.3
29 MultiMate Adv. 3.6
30 MultiMate Adv. 3.7
31 MultiMate 4
32 FrameMaker
34 Lotus 1-2-3
35 Lotus 1-2-3 (.WK3)
36 Lotus 1-2-3 (.WK4)
37 Lotus Manuscript
39 OfficeWriter 6.x
40 PageMaker
41 PFS: First Choice 2.0
42 PFS: First Choice 3.0
44 Professional Write 2.0
45 Professional Write 2.2
46 Quattro (.WK1)
47 RTF (Rich Text Format)
50 Samna Word IV

- 51 Windows Write
- 52 Word for Windows 1.x
- 53 Word for Windows 2.x
- 54 Word for Windows 6.0
- 57 WordPerfect 5.0 (DOS)
- 58 WordPerfect 5.1 (DOS)
- 59 WordPerfect 6.0 (Windows)
- 60 WordPerfect 6.1 (Windows)
- 61 WordStar 5.5
- 62 WordStar 6.0
- 63 WordStar 7.0
- 64 WordStar 1.x (Windows)
- 66 XyWrite III Plus
- 67 XyWrite IV

Select the Output Destination

After Caere completes a recognition attempt, the output string will be reported to one of two destinations, based on the value of the **OutputTo** property. This enumerated property has two valid options:

- 0 String
- 1 File

0--String populates the **Result** property with the output string. Under most circumstances, the **OutputFormat** property should be set to one of the ASCII text options when the OCR results are sent to the **Result** property.

1--File writes the OCR results string to a file. Several properties set parameters to the file writing process:

OutputFileName	A string property that specifies the fully qualified path and file name to the output file.
OutputFormat	An enumerated property that specifies the format of the output file. Options range from several configurations of ASCII text to popular word processing documents. Options for this property are detailed in the section "OutputFormat Property" on page 38.

OutputStyle	An enumerated property that specifies the layout of the output text. Options for this property are detailed in the section 'Select the Output Format' on page 13.
-------------	---

Begin Optical Character Recognition

Starting a recognition attempt is the final step in converting image text to machine readable text. Any configuration changes to the control or the output should be made prior to initializing recognition. Refer to 'Performing OCR' on page 7 for a step-by-step guide to setting up the recognition attempt.

After all configuration is complete, a new recognition attempt can begin. Begin recognition by executing one of the following methods:

OCRPage	Processes the entire input page without regard to its source or the definition of any regions within that source.
OCRRegion	<p>Begins OCR on one or more regions in the current image. If the RegionSource property is set to a valid value, the region coordinates specified from that source will be used.</p> <p>If the RegionSource property is "<None>", the Caere control's own region definitions will be applied. Refer to the section titled 'Define Region to Process' on page 10 for details on the control's region definitions.</p>

The following sample code will create this link to the Display control for both the **ImageDataSource** and the **RegionSource** of the Caere control.

```
' create data links between ImageBASIC controls
TMSDispl.ImageDataSource = TMSFile1.Link
Caere1.ImageDataSource = TMSDispl.Link

' set the source of region coordinates for OCR
Caere1.RegionSource = TMSDispl.Link

' load a file; it will be displayed in the TMS
' Display control
TMSFile1.LoadFile "c:\pending\10001.tif"

' specify a file as the text output and
' specify output file type and text layout
Caere1.OutputTo = 1 ' File
Caere1.OutputFormat = 31 ' Windows Write file
Caere1.OutputStyle = 1 ' Left Justified Only

' specify the file for output
```

```

Caere1.OutputFileName = "d:\text\10001.wri"

'   specify the Working Region of the Display
'   control; this is the portion of image data that
'   will be supplied to the Caere control for OCR
TMSDispl.RegTop = 150
TMSDispl.RegBottom = 1500
TMSDispl.RegLeft = 250
TMSDispl.RegRight = 2000

'   start OCR on the region; the output file will be
'   written by Caere when recognition is complete
Caere1.OCRRegion

```

Recognition Events

During the recognition attempt, the Caere engine can trigger certain events. These events can report errors or request information from the operator. Using these events, the developer can create a custom look and processing system based on individual requirements. The events that can occur as follows:

Progress	Occurs frequently throughout the processing performed by the control. Reports the current progress in the recognition process and allows canceling of the attempt.
Error	<p>Occurs each time an error internal to the Caere control is detected. This event reports the following parameters:</p> <p><i>Number</i> A long error code that identifies the error</p> <p><i>Description</i> Descriptive string of the error</p> <p><i>SCode</i> A composite long number indicating the severity of the error, the facility code, the origin of the error, and the native error code</p> <p><i>Source</i> Descriptive string of the source of the error</p> <p><i>HelpFile</i> Suggested help file name that should have a detailed explanation of the error</p> <p><i>HelpContext</i> Context ID of the appropriate topic in the help file named above</p> <p><i>CancelDisplay</i> If set to True during this event, the standard error dialog will not be displayed</p>

Optimization of the Recognition Attempt

Optical Character Recognition is a complex and difficult task for any computer or program. It is therefore best to provide as much information as possible about the expected results to the recognition system. This helps the system make the best determination in those cases where different interpretations of the image text are possible.

In addition, the image itself may be of less than ideal quality and may require some preprocessing enhancement. For example, many documents and images are dirty, distorted or low resolution. In addition, many types of forms include regions of shading or inverse (white-on-black) text. Other documents are formatted in columns or irregular sections. All of these document characteristics reduce OCR speed and accuracy.

Enhancement of the Input Image

The foremost aid in accurate recognition is provision of a clean, straight, high-resolution image. Of course, this is not always immediately possible, whether because of abuse of the paper document during its life cycle or because of capture or transmission distortions. ImageBASIC offers tools to perform a variety of image enhancement operations that can significantly improve OCR results.

Preprocessing Enhancement by the Control

The Caere control itself can adjust to and compensate for some image discrepancies; for example, skewed or rotated images may be corrected before processing. During processing, the use of field or character type definitions and both language and user-defined lexicons can significantly improve the accuracy of OCR output. Remember that all image enhancements performed by the Caere control in its preprocessing stage are discarded after recognition is complete, so if you need to save the corrected image, another tool must be used.

Correction of OCR Through Lexicons

Two types of lexicons are available for use by the Caere control: language lexicons and user-defined lexicons. A number of different language lexicons are included in the basic Caere installation. Any one of these languages may be selected for each recognition attempt. These standard lexicons contain common words in each language and are useful in the recognition of large bodies of text composed primarily of common or typical words.

Preprocessing Image Enhancement

The Caere control includes a number of options for the recognition of difficult or unusual images and document formats. Documents are frequently printed in

columns or in dot matrix text or in low resolution, all of which interfere with accurate character recognition. The Caere recognition engine has been trained to accommodate the distortions or difficulties inherent in recognizing these documents. A series of properties in the control, each beginning with the prefix *Opt*, may be independently set to enable each of the recognition features described in the paragraphs below.

Dot Matrix Printed Text

Some of the most difficult documents to accurately process by OCR are those which were made on a dot matrix printer. Dot matrix text is difficult for OCR engine to recognize because the characters are not continuous, and it is therefore difficult to distinguish between adjacent characters on the printed page. Because it can interfere with recognition of normal text, this option should be disabled except when it is needed.

The Caere recognition engine has been specifically trained to be able to more accurately recognize dot matrix print. To enable this feature, set the **OptDotMatrix** property to True before beginning recognition:

```
Caere1.OptDotMatrix = True
```

Low Resolution or Distorted Text

The **ICRMode** property specifies the type relative readability of an incoming image:

- 0 Standard -- Normal documents
- 1 Degraded -- Faxed or otherwise low resolution or distorted images

When recognizing faxed images, enable this feature of Caere by setting the **ICRMode** property:

```
Caere1.ICRMode = 1
```

This option increases the time needed for OCR to complete. It should be enabled only when necessary or when the quality of the incoming images is unknown. If high speed processing is required, this feature should be disabled.

Rotated Originals

Under many conditions, images are supplied for processing in a variety of orientations -- upside-down or landscape, for example. When processing these images using other ImageBASIC controls, both ScanFix and Display can rotate the image to the correct orientation. Using the display control, images may be manually corrected, while the ScanFix control can be instructed to automatically detect and correct for rotated images.

*If the orientation of incoming images is known*pre-recognition will proceed more efficiently if the correction that needs to be applied is specified by setting the **OptRotation** property to one of its enumerated values:

- 0 No rotation
- 1 90 degrees
- 2 180 degrees
- 3 270 degrees

*If the orientation of incoming image is not known*the control can automatically detect and correct orientation at the standard rotations of 90, 180 or 270 degrees. Fractional rotation, or skew, can be corrected separately. Three Boolean properties enable and disable automatic orientation:

OptAutoPortrait	<p>If True, the control will rotate the incoming image to portrait orientation before attempting OCR. This option will allow the image to be rotated to upside-down.</p> <p>Caere1.OptAutoPortrait = True</p> <p>If False, the image will be processed as presented.</p>
OptAutoUpright	<p>If True, the control will attempt to determine the proper rotation for the image text to be upright and will rotate the image before attempting OCR.</p> <p>Caere1.OptAutoUpright = True</p> <p>If False, the image will be processed as presented.</p>
OptFullOrientation	<p>If True, the control will attempt to determine both portrait orientation and upright orientation for the text of the image before beginning OCR.</p> <p>Caere1.OptFullOrientation = True</p> <p>If False, the image will be processed as presented.</p>

This rotation is performed as a pre-processing step and is valid only during the recognition attempt. In order to permanently rotate an image, another ImageBASIC component such as a Display control or ScanFix must be used.

Skewed Originals

Documents are commonly skewed -- i.e., rotated to some small degree up to about 10 degrees -- during scanning. Typically, the accuracy and efficiency of OCR rapidly degrades above skews of 5%. Therefore, the Caere control contains the necessary functionality to deskew an image before it is recognized.

To enable automatic deskewing, set the **OptAutoDeskew** property to True:


```
Caere1.OptAutoDeskew = True
```

The deskewing that is performed by the Caere control as a preprocessing step allows the control to properly recognize skewed images. However, the corrected image cannot be saved. To permanently correct the skew of an image, the ScanFix control should be used.

Character Type Selection

If the recognition engine knows what types of characters to expect in the image, OCR accuracy can be greatly improved. The following properties are used to optimize the separation and identification of individual characters:

LexicalClass Specifies the character type (alphabetic or numeric) and small fonts (font size of 4 to 6 points).

At design time, the LexicalClass property is set through a dialog that is displayed when the property name is double-clicked in the Properties Window. The available options may be independently enabled and disabled by checking or unchecking the box next to each.

At runtime, the property is set using the following constants:

Numeric IBBF_LEXCLS_NUMBER

Alphabetic IBBF_IEXCLS_ALPHA

Small Text IBBF_LEXCLS_SMALL_TEXT

To set multiple types for a single OCR attempt, the constants added together using the OR operand:

```
Dim nClass as Integer
```

```
nClass = IBBF_LEXCLS_NUMBER OR  
IBBF_LEXCLS_ALPHA
```

```
Caere1.LexicalClass = nClass
```

Pitch Specifies whether the type font is fixed width or variable. Character separation in fixed width (or nonproportional) fonts is more easily accomplished because all characters are a single width. The engine must be more careful separating adjacent characters in variable width (or proportional) fonts because the characters can overlap a slight amount.

The options for this enumerated property are as follows:

0 Not Fixed

1 Fixed

2 Auto (Default)

Using Language Lexicons

A number of different language files are available for use by the Caere control. These files enable the recognition engine to accurately process documents in many different languages. Using a language lexicon allows the recognition engine to assign a higher confidence level to any word that is found in the lexicon.

These language lexicons contain a standard dictionary for each language, but, as a rule, do not include technical or scientific terms. To optimize the recognition of documents containing relatively rare or specialized words, consider the use of a user-defined lexicon in addition to the standard language lexicon.

The following properties are used to specify the lexicon that will be applied to the recognition attempt:

Language	Enumerated specification of one of the predefined language lexicons
LexiconFileName	Specifies the full path and file name to a user-defined lexicon
LexicalMode	Enables the lexicons specified in the two properties above.
SpellCheck	Specifies whether the engine will use the lexicon to correct any words that it can or if the words that do not match a lexicon entry will be assigned a lower confidence.

The options for this enumerated property are as follows:

- 0 None
- 1 Check
- 2 Correct

0--None Allows the engine to assign confidence levels as it normally would.

1--Check If the recognized word is found in the lexicon, it will be assigned a higher confidence.

2--Correct If the recognized word almost matches a lexicon entry (i.e., if it would match using one of the alternate characters supplied by the recognition engine), the output string will be modified to the corrected spelling.

Enable Standard Language Lexicons

Selecting a Language Lexicon

The specification of a language lexicon to use during OCR is made with the **Language** property. This property is set according to an enumerated list of these languages:

- 0 English
- 1 German
- 2 French
- 3 Spanish
- 4 Italian
- 5 Dutch
- 6 Swedish
- 7 Norwegian
- 8 Finnish
- 9 Danish
- 10 Portuguese
- 12 Russian

Activating Lexicon Use

The **LexicalMode** property specifies whether or not a lexicon is to be used during the next OCR attempt. **LexicalMode** accepts values the following values:

- 0 No Lexical Processing
- 2 Absolute

0--No Lexical Processing

Indicates that the **Language** lexicon will not be accessed. Also, the setting of the **LexicalClass** property will be ignored and OCR will progress without any lexical help.

2--Absolute

Enables the specified lexicons. If the **SpellCheck** property is True, words in the output string may be modified to match words in these lexicons.

User-Defined Lexicons

The user may define a separate lexicon for use in processing documents composed of unique or unusual words not found in the standard language lexicons. A typical user-defined lexicon may include industry- or trade-specific terms not in general use in the language. Another common application is in the recognition of documents or fields in which all possible outputs are known such as employee lists or payrolls, or product lists or invoices.

Create a User-Defined Lexicon

User-defined lexicons are text (ASCII) files which may contain any number of words. The only limitation is that the entire lexicon is held in memory during processing, so a very large user-defined lexicon will reduce the available memory for other processes.

The formation of a new, user-defined lexicon is straightforward. Create an ASCII text file in any text editor or word processor that can save as text only.

- Each entry in a lexicon file must be on a separate line, separated from the next entry by a carriage return (ASCII code 0A hex) and line feed (0D hex) character.
- Each entry must consist of a single word without spaces.
- Entries are case-sensitive, so if an entry will appear with different capitalization, all of the possible forms should be entered in the lexicon.

A portion of a sample user-defined lexicon is shown below.

```
ImageBASIC
Caere
OCX
LexicalClass
lexical class
VBX
vbx
Delphi
VB
photometric
```

After its creation, the user-defined dictionary will be used during recognition if it is specified in the **LexiconFileName** property and if the **LexicalMode** property enables the use of lexicons

Enable User-Defined Lexicons

The **LexicalMode** property specifies whether or not a lexicon is to be used during the next OCR attempt, and how closely the words returned by the recognition engine must match entries in the available lexicons. **LexicalMode** accepts values the following values:

- 0 No Lexical Processing
- 2 Absolute

0--No Lexical Processing

Indicates that the **Language** lexicon will not be accessed. Also, the setting of the **LexicalClass** property will be ignored and OCR will progress without any lexical help.

2--Absolute

Enables the specified lexicons. If the **SpellCheck** property is True, words in the output string may be modified to match words in these lexicons.

Chapter 3 : Reference

Reference to Properties, Methods and Events

AboutBox Method

Definition:	Displays a message box showing version and copyright information when queried.
Syntax:	<code>Caere1.AboutBox</code>
Design Access:	Read-only
Runtime Access:	Read-only
Comments:	The message box is application modal and contains a single OK button. Clicking the button will close the message box.

Active Property

Definition:	<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>
Data Type:	Boolean
Design Access:	Read/Write
Runtime Access:	Read/Write (see limits below)
See Also:	"Licensing Configuration and Verification" on page 3
Comments:	<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">1) The related technology libraries for the control will not be loaded.2) The licensing server will not verify an available token for the control.

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.

AddRegion Method

Definition: Allocates a new region for multiple region processing.

Parameters: None

Syntax: `intRegIndex = Caere1.AddRegion`

Data Type: Long Integer

Return Values: Index value of the newly allocated region

See Also: DeleteRegion Method, RegCount Property

Comments: When this method is successfully executed, the **RegCount** property is incremented by one. A new element of the region definition properties, listed below, is allocated and all coordinates for the new region are set to 0 (zero). The new region may be modified by selecting the new region through the **RegIndex** property and changing the value of the region definition properties:

- RegLeft
- RegTop
- RegRight
- RegBottom

DebugPath Property

Definition: If set to a valid file name, the control will generate debug information and write it to the file.

Data Type: String

Design Access: Read/Write

Runtime Access: Read/Write

See Also: Progress Event

Comments: The debug information that is supplied here is related mainly to the OCR process. This information may be requested by a technical support representative. The information written to this file is detailed and reflects the low-level processes of the engine.

DeleteAllRegions Method

Definition:	Deletes all of the region elements defined in the Caere control.
Parameters:	None
Syntax:	<code>Caere1.DeleteAllRegions</code>
Data Type:	Long Integer
Return Values:	None
See Also:	AddRegion Method, RegIndex Property
Comments:	When this method is successfully executed, the RegCount and RegIndex properties are set to 0 (zero), and all region information is cleared from memory.

DeleteRegion Method

Definition:	Deletes the current region element as specified in the RegIndex property.
Parameters:	None
Syntax:	<code>Caere1.DeleteRegion</code>
Data Type:	Long Integer
Return Values:	None
See Also:	AddRegion Method, RegIndex Property
Comments:	When this method is successfully executed, the RegCount property is updated, and the RegIndex property is changed only if necessary.

Error Event

Definition:	Occurs for each error internal to the control.														
Parameters:	<table><tr><td>Number</td><td>A long error code that identifies the error</td></tr><tr><td>Description</td><td>Descriptive string of the error</td></tr><tr><td>SCode</td><td>A composite long number indicating the severity of the error, the facility code, the origin of the error, and the native error code</td></tr><tr><td>Source</td><td>Descriptive string of the source of the error</td></tr><tr><td>HelpFile</td><td>Suggested help file name that should have a detailed explanation of the error</td></tr><tr><td>HelpContext</td><td>Context ID of the appropriate topic in the help file named above</td></tr><tr><td>CancelDisplay</td><td>If set to True during this event, the standard error dialog will not be displayed</td></tr></table>	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 of the appropriate topic in the help file named above	CancelDisplay	If set to True during this event, the standard error dialog will not be displayed
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 of the appropriate topic in the help file named above														
CancelDisplay	If set to True during this event, the standard error dialog will not be displayed														

ICRMode Property

- Definition:** Specifies whether the image is standard or was generated by a fax server or machine. Faxed documents require additional processing to compensate for the typically lower resolution and image quality.
- Data Type:** Enumerated
- Design Access:** Read/Write
- Runtime Access:** Read/Write
- Possible Values:** 0 Standard
1 Degraded
- See Also:** OptAutoSetDegrade Property, OptAutoFax Property
- Description:** This property is superseded by the **OptAutoSetDegrade** and **OptAutoFax** properties but has been retained for backward compatibility.
0--Standard is the preferred setting for normal documents.
1--Degraded enhances recognition of faxed or other low resolution documents but can slow processing and reduce accuracy if used when unnecessary.

ImageDataSource Property

- Definition:** Specifies the ImageBASIC control that will supply image data for OCR by this control.
- Data Type:** String
- Syntax:** `Caerel.ImageDataSource = TMSDispl.Link`
- Design Access:** Read/Write
- Runtime Access:** Read/Write
- See Also:** RegionSource Property
- Comments:** **ImageDataSource** must specify an ImageBASIC control that can supply image data if OCR is to be attempted.
When an ImageBASIC control is added to a Form at design time, the **ImageDataSource** property is automatically populated with a source ImageBASIC control if one already exists on the Form.
At runtime, this property may be set to the **Link** value of any ImageBASIC control that is an image source; for example,
`Caerel.ImageDataSource = TMSDispl.Link`

ImageResolution Property

Definition:	Reports the DPI resolution of the image currently being processed by the control.
Data Type:	Long
Design Access:	Not Available
Runtime Access:	Read-only
See Also:	ImageDataSource Property
Comments:	This property will report a valid value only when an image is available to the control. This property is informational only and cannot be modified.

Language Property

Definition:	Specifies which of the installed languages lexicons will be used during recognition.
Data Type:	Enumerated
Design Access:	Read/Write
Runtime Access:	Read/Write
See Also:	LexicalClass Property, LexicalMode Property
Comments:	This property should be set to use different libraries to accord with the language in which the image text being recognized was written. Valid options for this property are as follows:

- 0 English
- 1 German
- 2 French
- 3 Spanish
- 4 Italian
- 5 Dutch
- 6 Swedish
- 7 Norwegian
- 8 Finnish
- 9 Danish
- 10 Portuguese
- 12 Russian

Although the Caere control is shipped with all of the currently available languages, one or more of these files may have been removed from your system. Therefore, be careful to set this property to an installed language file, or the recognition engine will not function properly.

LexicalClass Property

Definition: Specifies one of the built-in field types to optimize recognition.

Data Type: Enumerated

Design Access: Read/Write

Runtime Access: Read/Write

See Also: LexicalMode Property, Language Property, LexiconFileName Property

Comments: At design time, this property is set to any of its valid values through a dialog that is displayed when the property is clicked in the Properties Window. Each option may be marked with a check to select any number of field classes.

At runtime, the property is set through the constants listed below with the valid options. To select no lexical class, set this property to 0 (zero). More than one of these classes may be selected by adding together the constants, as in this example:

```
nLexSum = IBBF_LEXCLS_NUMBER + IBBF_LEXCLS_TIME
```

```
Caere1.LexicalClass = nLexSum
```

The options for this property and the constant for each option are as follows:

Number	IBBF_LEXCLS_NUMBER
Alphabetic	IBBF_LEXCLS_ALPHA
Small Text	IBBF_LEXCLS_SMALL_TEXT

Number recognizes the numerals 0-9, fractions and mathematical symbols.

Alphabetic recognizes all standard characters and punctuation.

Small Text optimizes for recognition of fonts from 4 to 6 points.

LexicalMode Property

Definition: Specifies the level to which the recognition engine will attempt to match the data in the lexicon(s) being used.

Data Type: Enumerated

Design Access: Read/Write

Runtime Access: Read/Write

Possible Values: 0 No Lexical Processing
2 Absolute

See Also: LexiconFileName Property, LexicalClass Property, Using Language Lexicon\$ on page 22

Comments: *0--No Lexical Processing* instructs the engine to ignore the information in both the standard language lexicon and the user-defined lexicon, if one has been specified.

2--Absolute instructs the recognition engine to return as unrecognizable any word that it does not find in either the standard lexicon, specified with the **Language** property, or the user-defined lexicon, if one has been specified in the **LexiconFileName** property.

LexiconFileName Property

Definition: Specifies fully qualified path and file name of the user-defined lexicon that will be used during OCR.

Data Type: String

Design Access: Read/Write

Runtime Access: Read/Write

See Also: LexicalMode Property, User-Defined Lexicon§ on page 24, Language Property

Comments: A user-defined lexicon is a plain text file with one entry per line. The exact use to which the lexicon will be put is specified by the **LexicalMode** property.

Link Property

Definition: Reports the Link ID calculated for this control at its creation.

Data Type: String

Syntax: `TMSDispl.ImageDataSource = Caere1.Link`

Design Access: Not Available

Runtime Access: Read-only

Comments: Each ImageBASIC control is assigned a unique Link ID at its creation. This Link ID can be specified in the **ImageDataSource**, **DisplaySource**, **RegionSource**, and **AnnoteSource** properties of various ImageBASIC controls. These source properties specify the ImageBASIC control that is supplying information or services to a control.

Note: The Caere control cannot be a source for any of these services. Therefore, the **Link** control for this control will not be used.

OCRPage Method

Definition: Initiates a recognition attempt on the current image page.

Parameters: None

Syntax: `Caere1.OCRPage`

Data Type: String

Return Values: OCR result string

See Also: OCRRegion Method, OutputTo Property

Comments: Recognition will be performed on the image currently available from the ImageBASIC control specified in the **ImageDataSource** property. The entire input page will be processed even if the **RegionSource** property specifies a valid source of region coordinates.

The recognition string will be sent to the destination specified in the OutputTo property. Refer to "Select the Output Destination" on page 15 for details on the output of recognition text.

OCRRegion Method

Definition: Initiates a recognition attempt on the current image region using the region coordinates supplied by the ImageBASIC control specified in the **RegionSource** property.

Parameters: None

Syntax: Caere1.OCRRegion

Data Type: String

Return Values: OCR result string

See Also: OCRPage Method, RegionSource Property, InputFrom Property, OutputTo Property

Comments: The image on which recognition will be performed is specified as follows:

If the **InputFrom** property is set to *0--ImageDataSource*, the image for OCR will be read from the ImageBASIC control specified in the **ImageDataSource** property.

If the **InputFrom** property is set to *1--File*, the image for OCR will be read from the file(s) specified in the **InputFileName** property.

The region of this image to be processed can be specified in either of two ways:

- 1) The **RegionSource** property may be set to specify any ImageBASIC control that can supply region coordinates. As of this writing, the Display controls are the only controls that can define a Working Region.

By default, if the image data is read from a Display control, the Working Region defined in that control will be the region that is read by OCR.

- 2) The Caere control can define the region using the following properties:

RegBottom

RegLeft

RegRight

RegTop

OptAutoDeskew Property

- Definition:** If True, the recognition engine will attempt to identify and correct skewed text before recognition begins. If False, the text will be read as presented.
- Data Type:** Boolean
- Design Access:** Read/Write
- Runtime Access:** Read/Write
- See Also:** OptAutoUpright Property, "Optimization of the Recognition Attempt" on page 18
- Description:** The recognition of text is degraded at skews of more than a few degrees. Therefore, deskewing should be enabled unless the incoming images are known to be straight.

OptAutoPortrait Property

- Definition:** If True, incoming images will be analyzed to determine if the page or region is portrait (i.e., the sides are longer than the width) and the images will be rotated to portrait if necessary.
- Data Type:** Boolean
- Design Access:** Read/Write
- Runtime Access:** Read/Write
- See Also:** OptAutoUpright Property, "Optimization of the Recognition Attempt" on page 18
- Description:** Most documents are printed in portrait mode, but many scanning processes feed the documents in landscape mode to speed the paper feed. The text in the image that is being read must be upright, and Caere will rotate the incoming image to portrait before further processing if the **OptAutoPortrait** property is True.
- Note:** The image may be upside-down even if it is portrait. Therefore, the **OptAutoUpright** property may be used instead of **OptAutoPortrait** if necessary.

OptAutoUpright Property

- Definition:** If True, Caere will attempt to identify the upright position of the text on the input image and rotate the image to correct for inappropriate rotation before recognition is started.
- Data Type:** Boolean
- Design Access:** Read/Write
- Runtime Access:** Read/Write
- See Also:** OptAutoPortrait Property, OptAutoDeskew Property, "Optimization of the Recognition Attempt" on page 18
- Description:** Determination of the actual upright position of text is based largely on the relative abundance of ascenders relative to descenders on the letters. For this reason, a relatively large segment of text -- at least two lines of text with white space on the left and right -- must be available in the region for the engine to correctly determine what is upright.
- Note:** The automatic determination of the upright position of the image text requires some additional processing time, and, therefore, this option should be disabled unless it is necessary.

OptDotMatrix Property

- Definition:** If True, Caere will expect incoming images to contain dot matrix text. If False, normal recognition will proceed, but dot matrix text will be poorly recognized.
- Data Type:** Boolean
- Design Access:** Read/Write
- Runtime Access:** Read/Write
- See Also:** OptDotDetect Property, OptAutoSetDegrade Property
- Description:** Because of its nature, dot matrix text requires special processing parameters to maximize recognition accuracy. Caere includes additional processing steps when recognizing dot matrix, but this property must be set to True to enable this feature.
- Note:** Because of the additional processing performed when this property is True, recognition will proceed slightly slower than usual. In addition, regular text may not be recognized as accurately. Therefore, disable dot matrix recognition unless it is required.

OptFullOrientation Property

- Definition:** If True, incoming images will be analyzed for to determine if they are both portrait and upright. The image will be rotated as necessary to make the text on the image upright.

Data Type: Boolean
Design Access: Read/Write
Runtime Access: Read/Write
See Also: OptAutoUpright Property, OptAutoPortraitProperty
Description: Setting this property to True is equivalent to setting the **OptAutoUpright** and **OptAutoPortrait** properties to True.

OptRotation Property

Definition: Specifies the degree to which the incoming image should be rotated before further processing.

Data Type: Enumerated

Design Access: Read/Write

Runtime Access: Read/Write

Possible Values:

0	None
1	90 Degrees
2	180 Degrees
3	270 Degrees

See Also: OptAutoDeskew Property, OptAutoUpright Property

Description: Options are available in the Caere control to rotate the incoming image to any of the primary positions or to allow the control to automatically determine the proper orientation. If the rotation of the incoming images is known, pre-recognition processing is faster if the exact degree of correctional rotation is performed. If the orientation of the incoming images is unknown, the control can perform automatic orientation, enabled through the **OptAutoUpright** property. Rotation of the image is rotation to the primary positions. Small degrees of rotation, or skew, can also be detected and corrected. Skew correction is enabled through the **OptAutoDeskew** property.

OutputFileName Property

Definition: Specifies the path and file name of the file to write with the recognition results.

Data Type: String

Design Access: Read/Write

Runtime Access: Read/Write

See Also: OutputToProperty, OutputFormat Property

Comments: This property is valid only if the **OutputToProperty** specifies *1--File*.

OutputFormat Property

Definition: Specifies the format of the recognition output string.

Data Type: Enumerated

Design Access: Read/Write

Runtime Access: Read/Write

See Also: OutputFileNameProperty, OutputToProperty

Comments: The valid enumerated values for this property are shown in the list below.

Note: All output formats numbered six and higher must be written to file instead of to a Visual Basic string variable or property. This is because all of these formats are binary output rather than text, and Visual Basic string properties and variables cannot accept this data.

- 7 Ami Pro 2.0
- 8 Ami Pro 3.0
- 9 ANSI
- 10 ASCII Database
- 11 ASCII Decolumnized
- 13 ASCII Standard (Default)
- 17 DCA/RTF
- 19 EBCDIC
- 21 Excel 3.0
- 22 Excel 4.0
- 24 MS Word 4.0
- 25 MS Word 5.x
- 26 MS Word 6.0
- 27 MS Works
- 28 MultiMate 3.3
- 29 MultiMate Adv. 3.6
- 30 MultiMate Adv. 3.7
- 31 MultiMate 4
- 32 FrameMaker
- 34 Lotus 1-2-3
- 35 Lotus 1-2-3 (.WK3)
- 36 Lotus 1-2-3 (.WK4)
- 37 Lotus Manuscript
- 39 OfficeWriter 6.x
- 40 PageMaker
- 41 PFS: FirstChoice 2.0
- 42 PFS: FirstChoice 3.0

- 44 Professional Write 2.0
- 45 Professional Write 2.2
- 46 Quattro (.WK1)
- 47 RTF (Rich Text Format)
- 50 Samna Word IV
- 51 Windows Write
- 52 Word for Windows 1.x
- 53 Word for Windows 2.x
- 54 Word for Windows 6.x
- 57 WordPerfect 5.0 (DOS)
- 58 WordPerfect 5.1 (DOS)
- 59 WordPerfect 6.0 (Windows)
- 60 WordPerfect 6.1 (Windows)
- 61 WordStar 5.5
- 62 WordStar 6.0
- 63 WordStar 7.0
- 64 WordStar 1.x (Windows)
- 65 Ventura
- 66 XyWrite III Plus
- 67 XyWrite IV

OutputStyle Property

Definition: Specifies how the left and right margins of the output text will be formatted.

Data Type: Integer (Enumerated)

Design Access: Read/Write

Runtime Access: Read/Write

See Also: OutputFileNameProperty, Result Property

Comments: Valid options for this property are as follows:

- 0 Left and Right Justify
- 1 Left Justify
- 2 Stream

0--Left and Right Justify causes the output text to be fully justified, and all white space on either end of all lines is disregarded.

1--Left Justify causes the output text to be justified to the left; i.e., all white space on the left of the text is compressed.

2--Stream causes the output text to be output as a single line of text; i.e., all white space, carriage returns and line feeds are stripped. This option is intended primarily for output to full text indexing systems.

OutputTo Property

- Definition:** Specifies the destination for OCR results.
- Data Type:** Integer (Enumerated)
- Design Access:** Read/Write
- Runtime Access:** Read/Write
- See Also:** OutputFileNameProperty, Result Property
- Comments:** Valid options for this property are as follows:
- 0 String
 - 1 File
- 0--String* causes the recognition results to be reported in the **Result** property.
- 1--File* causes the recognition results to be written to the file specified in the **OutputFileName** property.

Pitch Property

- Definition:** Specifies if the image text is a fixed width font, not a fixed width font, or unknown.
- Data Type:** Enumerated
- Design Access:** Read/Write
- Runtime Access:** Read/Write
- See Also:** LexicalClass Property
- Comments:** The valid options for this property are as follows:
- 0 Not Fixed
 - 1 Fixed
 - 2 Automatic Detection (Default)

Progress Event

- Definition:** Occurs frequently during recognition and provides percentage completed and the option to cancel recognition.
- Parameters:**
- | | |
|---------|--|
| Percent | Percentage completion of current OCR attempt |
| Cancel | If set to True, this OCR attempt is canceled |
- Comments:** The only way to cancel a recognition attempt is to set the *Cancel* parameter to this event to True during the event. The cancellation may be passed to the event as a global variable set in a button click or through some other method, as shown here:
- ```
Private Sub cmdCancel_Click()
 gnStop = True
End Sub
```

```

Private Sub Caere1_Progress(ByVal Percent As
 Integer, Cancel As Boolean)

 ' ensure processing of all GUI events

 DoEvents

 ' set local parameter to stop recognition

 Cancel = gnStop

End Sub

```

### ***RegBottom Property***

- Definition:** Reports or sets the image pixel height of the region for recognition.
- Data Type:** Long
- Design Access:** Read/Write
- Runtime Access:** Read/Write
- See Also:** OCRRegion Method, RegIndex Property
- Comments:** When the **OCRRegion** method is executed to begin recognition and no **RegionSource** is specified, the Caere control's own region definition properties are applied. The **RegCount** property reports the number of defined regions and the following additional properties are used to define those regions:
- RegIndex May be set to any integer from 1 to **RegCount**. When set, the following properties are updated.
  - RegLeft Image pixel coordinate of left edge of region
  - RegTop Image pixel coordinate of top edge of region
  - RegRight Image pixel coordinate of right edge of region
  - RegBottom Image pixel coordinate of bottom edge of region

### ***RegCount Property***

- Definition:** Reports the total number of regions defined for multi-region processing.
- Data Type:** Long
- Design Access:** Not Available
- Runtime Access:** Read-only
- See Also:** RegIndex Property, AddRegion Method
- Comments:** When the **OCRRegion** method is executed to begin recognition and no **RegionSource** is specified, the Caere control's own region definition properties are applied. The **RegCount**

property reports the number of defined regions and the following additional properties are used to define those regions:

|                  |                                                                                                       |
|------------------|-------------------------------------------------------------------------------------------------------|
| <b>RegIndex</b>  | May be set to any integer from 1 to <b>RegCount</b> . When set, the following properties are updated. |
| <b>RegLeft</b>   | Image pixel coordinate of left edge of region                                                         |
| <b>RegTop</b>    | Image pixel coordinate of top edge of region                                                          |
| <b>RegRight</b>  | Image pixel coordinate of right edge of region                                                        |
| <b>RegBottom</b> | Image pixel coordinate of bottom edge of region                                                       |

### ***RegIndex Property***

|                        |                                                                                                                                                                                                                                                                                                                           |
|------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Definition:</b>     | Specifies the active region for the reporting and setting of region coordinates.                                                                                                                                                                                                                                          |
| <b>Data Type:</b>      | Long                                                                                                                                                                                                                                                                                                                      |
| <b>Design Access:</b>  | Not Available                                                                                                                                                                                                                                                                                                             |
| <b>Runtime Access:</b> | Read-only                                                                                                                                                                                                                                                                                                                 |
| <b>See Also:</b>       | RegCount Property, RegBottom Property, RegLeft Property, RegRight Property, RegTop Property                                                                                                                                                                                                                               |
| <b>Comments:</b>       | When the <b>OCRRegion</b> method is executed to begin recognition and no <b>RegionSource</b> is specified, the Caere control's own region definition properties are applied. The <b>RegCount</b> property reports the number of defined regions and the following additional properties are used to define those regions: |
| <b>RegIndex</b>        | May be set to any integer from 1 to <b>RegCount</b> . When set, the following properties are updated.                                                                                                                                                                                                                     |
| <b>RegLeft</b>         | Image pixel coordinate of left edge of region                                                                                                                                                                                                                                                                             |
| <b>RegTop</b>          | Image pixel coordinate of top edge of region                                                                                                                                                                                                                                                                              |
| <b>RegRight</b>        | Image pixel coordinate of right edge of region                                                                                                                                                                                                                                                                            |
| <b>RegBottom</b>       | Image pixel coordinate of bottom edge of region                                                                                                                                                                                                                                                                           |

### ***RegionSource Property***

|                        |                                                                                                                                    |
|------------------------|------------------------------------------------------------------------------------------------------------------------------------|
| <b>Definition:</b>     | Specifies the ImageBASIC control that will supply region coordinates for recognition when the <b>OCRRegion</b> method is executed. |
| <b>Data Type:</b>      | String                                                                                                                             |
| <b>Design Access:</b>  | Read/Write                                                                                                                         |
| <b>Runtime Access:</b> | Read/Write                                                                                                                         |
| <b>See Also:</b>       | ImageDataSourceProperty                                                                                                            |
| <b>Comments:</b>       | As of this writing, the primary control used for region definition is the Display control. The Working Region that is defined in   |

that control is the image region that will be retrieved when **RegionSource** specifies that control.

### ***RegLeft Property***

**Definition:** Specifies the left edge of the region to OCR in image pixels.  
**Data Type:** Long  
**Design Access:** Read/Write  
**Runtime Access:** Read/Write  
**See Also:** RegTop Property, RegRight Property, RegionSource Property  
**Comments:** If accepting image data from an ImageBASIC control that can define regions, the **RegionSource** property may be set to the source control to define the region to OCR.  
If accepting image data from an ImageBASIC control that cannot define a region, the **RegBottom**, **RegRight**, **RegLeft** and **RegTop** properties may be set to define the region for processing through the **OCRRegion** method.

### ***RegRight Property***

**Definition:** Specifies the width of the region to OCR in image pixels.  
**Data Type:** Long  
**Design Access:** Read/Write  
**Runtime Access:** Read/Write  
**See Also:** RegLeft Property, RegBottom Property, RegionSource Property  
**Comments:** If accepting image data from an ImageBASIC control that can define regions, the **RegionSource** property may be set to the source control to define the region to OCR.  
If accepting image data from an ImageBASIC control that cannot define a region, the **RegBottom**, **RegRight**, **RegLeft** and **RegTop** properties may be set to define the region for processing through the **OCRRegion** method.

### ***RegTop Property***

**Definition:** Specifies the top edge of the region to OCR in image pixels.  
**Data Type:** Long  
**Design Access:** Read/Write  
**Runtime Access:** Read/Write  
**See Also:** RegLeft Property, RegBottom Property, RegionSource Property  
**Comments:** If accepting image data from an ImageBASIC control that can define regions, the **RegionSource** property may be set to the source control to define the region to OCR.

If accepting image data from an ImageBASIC control that cannot define a region, the **RegBottom**, **RegRight**, **RegLeft** and **RegTop** properties may be set to define the region for processing through the **OCRRegion** method.

### ***Result Property***

- Definition:** If **OutputTo** specifies *0--String*, reports the final recognition results after a successful OCR attempt.
- Data Type:** String
- Design Access:** Not Available
- Runtime Access:** Read-only
- See Also:** **OutputToProperty**
- Comments:** The string that is reported in this property will be the final recognition results. The string will be formatted as specified in the **OutputFormat** property.
- Valid only when **OutputTo** is set to *1--String*. After recognition is complete, this property will report the total number of regions that were processed. When the **ResultIndex** property is set to any integer value between 1 (one) and **ResultCount** inclusive, the **Result** property is populated with the output string for the specified region.

### ***ResultCount Property***

- Definition:** Reports the number of processed regions. In this revision of the Caere control, will always report 1 (one).
- Data Type:** Long
- Design Access:** Not Available
- Runtime Access:** Read-only
- See Also:** **ResultIndex Property**, **ResultProperty**
- Comments:** Valid only when **OutputTo** is set to *1--String*. After OCR is complete, this property will report the total number of regions that were processed. When the **ResultIndex** property is set to any integer value between 1 (one) and **ResultCount** inclusive, the **Result** property is populated with the output string for that region.

### ***ResultIndex Property***

- Definition:** Specifies the region whose OCR results will be reported in the **Result** property. In this revision of the Caere control, the only valid value is 1 (one).



**Data Type:** Long

**Design Access:** Not Available

**Runtime Access:** Read/Write

**See Also:** ResultCountProperty, ResultProperty

**Comments:** Valid only when **OutputToIs** set to *1--String*.  
 After recognition is complete, the **ResultCount** property will report the total number of regions that were processed. When the **ResultIndex** property is set to any integer value between 1 (one) and **ResultCount** inclusive, the **Result** property is populated with the output string for the specified region.

### ***SpellCheck Property***

**Definition:** If True, the recognition engine will attempt to identify and correct misspelled OCR results by comparing the OCR output to words in all active dictionaries. If False, misspelled words will not be corrected, but the most confident character string will be reported.

**Data Type:** Boolean

**Design Access:** Read/Write

**Runtime Access:** Read/Write

**See Also:** Language Property, LexiconFileName Property

**Comments:** This feature will not correct actual misspellings in the document. If enabled, it will only correct for OCR mistakes, not errors in the input text.



# Index

## A

- AboutBox Method 27
- Active Property 27
- Annotation Control 1

## C

- Cancel Parameter 40
- Character Type
  - Pitch Property 40
- Checking Spelling
  - SpellCheck Property 45
- Completion Event Parameters
  - Cancel 40
  - Percent 40
- Control Communication 1

## D

- DebugPath Property 28
- Deskewing
  - OptAutoDeskew Property 35
- Dictionaries (Lexicons) 33
- Dot Matrix Originals 19
- Dot Matrix Text
  - OptDotMatrix Property 36

## E

- Error Event 29
- Event Parameters
  - Cancel 40
  - Percent 40
- Events
  - Error 29
  - Progress 40

## F

- Faxed Text
  - ICRMode Property 30
- Field Type Definition 32
- File Names, Lexicons 33
- File Output
  - OutputFileName Property 37
  - OutputFormat Property 38
  - OutputStyle Property 39
  - OutputTo Property 40
- Foreign Language Recognition 31

## I

- ICRMode Property 30
- Image Transfer
  - Link Property 33
- ImageDataSource 1, 2
- ImageDataSource Property 30
- ImageResolution Property 31

## L

- Language Lexicons 22
- Language Property 31
- LexicalClass Property 32
- LexicalMode Property 32
- LexiconFileName Property 33
- Lexicons
  - LexicalClass 32
  - LexiconFileName Property 33
- Licensing
  - Active Property 27
- Limiting OCR Results 32
- Link ID 2
- Link property 2, 33
- Linking
  - ImageDataSource Property 30
- Linking Controls 1, 2
- Load Time Improvement
  - Active Property 27
- Locating Lexicons 33
- Low Resolution
  - ICRMode Property 30

## M

- Methods
  - AboutBox 27
  - OCRPage 33
  - OCRRegion 34
- Multiple Region Definition
  - RegCount Property 41
  - RegIndex Property 42

## O

- OCR Results
  - Result Property 44
- OCR Setup 7
- OCRPage Method 33
- OCRRegion Method 34
- OptAutoDeskew Property 35
- OptAutoPortrait Property 35
- OptAutoUpright Property 36
- OptDotMatrix Property 36

- OptFullOrientation Property 36
- OptRotation Property 37
- Orientation
  - OptAutoPortrait Property 35
  - OptAutoUpright Property 36
  - OptFullOrientation Property 36
  - OptRotation Property 37
- OutputFileName Property 37
- OutputFormat Property 38
- OutputStyle Property 39
- OutputTo Property 40

## P

- Percent Parameter 40
- Pitch Property 40
- Progress Event 40
- Properties
  - Active 27
  - DebugPath 28
  - ICRMode 30
  - ImageDataSource 30
  - ImageResolution 31
  - Language 31
  - LexicalMode 32
  - Link 33
  - OptAutoDeskew 35
  - OptAutoPortrait 35
  - OptAutoUpright 36
  - OptDotMatrix 36
  - OptFullOrientation 36
  - OptRotation 37
  - OutputFileName 37
  - OutputFormat 38
  - OutputStyle 39
  - OutputTo 40
  - Pitch 40
  - RegBottom 41
  - RegCount 41
  - RegIndex 42
  - RegionSource 42
  - RegLeft 43
  - RegRight 43
  - RegTop 43
  - Result 44
  - ResultCount 44
  - ResultIndex 44
  - SpellCheck 45

## R

- Recognition

- OCRPage Method 33
- OCRRegion Method 34
- RegBottom Property 41
- RegCount Property 41
- RegIndex Property 42
- RegionSource Property 42
- RegLeft Property 43
- RegRight Property 43
- RegTop Property 43
- Recognition Output 40
  - Result Property 44
- Recognition Output Text Format
  - OutputStyle Property 39
- Recognition, Starting 7
- RegBottom Property 41
- RegCount Property 41
- RegIndex Property 42
- RegionSource Property 42
- RegLeft Property 43
- RegRight Property 43
- RegTop Property 43
- Result Property 44
- ResultCount Property 44
- ResultIndex Property 44
- Rotated Images
  - OptAutoPortrait Property 35
  - OptAutoUpright Property 36
  - OptFullOrientation Property 36
  - OptRotation Property 37
- Rotated Originals 19
- Routing Data between Controls 1
- Runtime Linking 2

## S

- Skewed Originals 20
- Skewed Text
  - OptAutoDeskew Property 35
- Source of Image Data 1
- SpellCheck Property 45

## T

- Timing of Licensing Verification
  - Active Property 27
- TMS Display Control 1