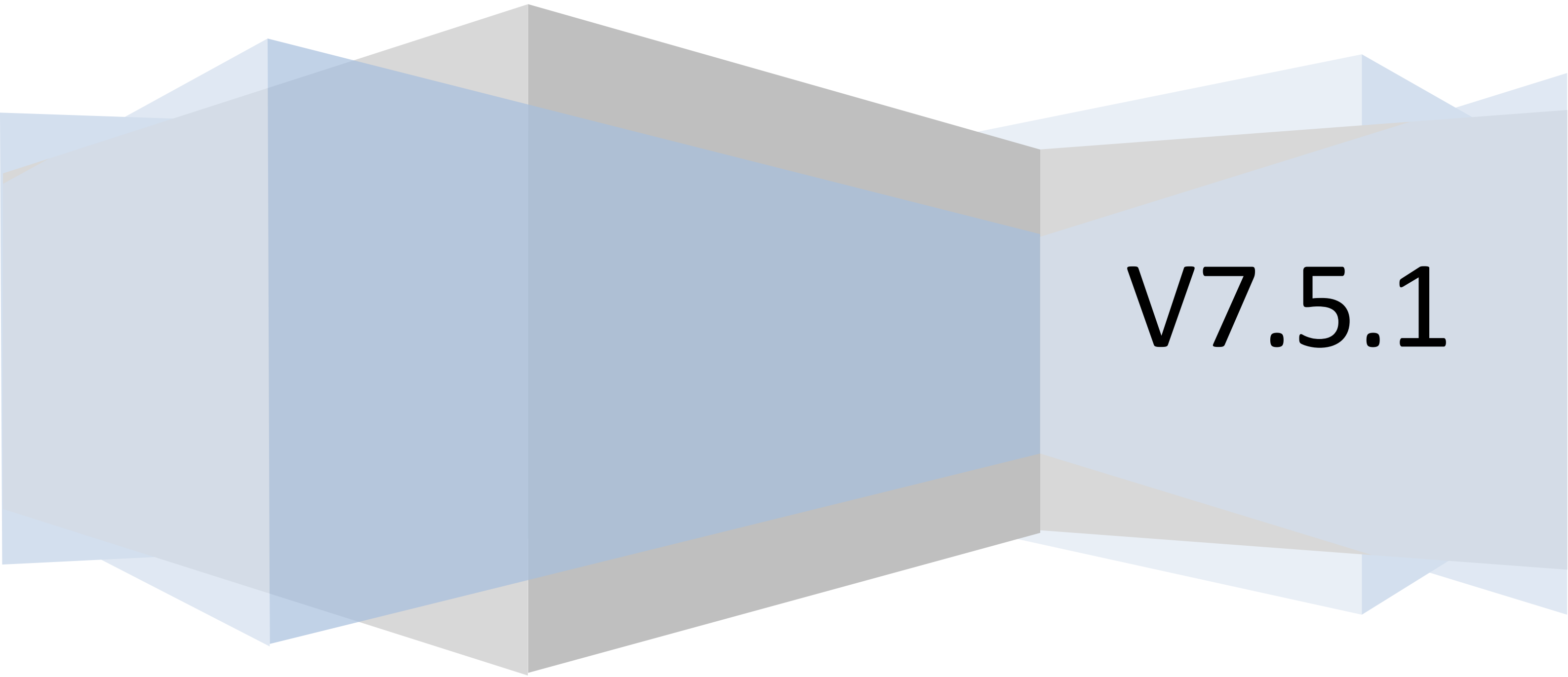
Softek Software Ltd

Softek Barcode Reader Toolkit for IOS 4 and IOS 5

Product Documentation



V7.5.1

1 Contents

2 Overview

The Softek Barcode Reader Toolkit for IOS allows you to integrate barcode reading into your app using an objective C class and a static library called bardecode.a.

The objective C class provides a simple way to set the properties of the SDK and also controls frame grabbing from the video camera on the device. Note that the source code to the objective C class is included in the download and developers may wish to adapt it to meet the requirements of their app.

The SDK is compatible with IOS 5. To use the sample project on IOS 5 it is necessary to select “Edit Project Settings” from the Project menu and change the Base SDK setting in the Build tab from IOS 4.2 to IOS 5.

3 SDK Contents

The SDK includes the following files and folders:

sdk/barcode.h	Header file used in the Bardecode class
sdk/Bardecode.h	Header file for the Bardecode class
sdk/Bardecode.m	Objective C file for the Bardecode class
sdk/libbardecode.a	Softek Barcode Reader SDK static library
xcode_project	Sample XCode project

4 Integrating the SDK into an existing XCode App (IOS 4 or later)

4.1 Add the SDK files to your project

Navigate to the sdk folder using Finder.

Drag and drop the Bardecode.m file to the Classes section of the Groups and Files pane of your XCode project.

Repeat for Bardecode.h

Drag and drop the libbardecode.a file to the Products section of the Groups and Files pane.

4.2 Add the hooks into an existing View Controller

Add a new class member:

```
Bardecode *barcode;
```

Modify viewDidLoad to create an instance of the Bardecode class when the view loads:

```
barcode = [Bardecode alloc] ;
[barcode init] ;
barcode.Delegate = self ;
barcode.ViewForPreview = self.view ;
```

Set some properties, such as the barcode types you wish to read…

```
barcode.ReadQRCode = true;
```

Refer to chapter 5 for more properties of the bardecode class.

Add a dealloc function to clean up when the view unloads:

```
- (void)dealloc {
  [barcode dealloc];
   [super dealloc];
}
```

Add a BardecodeDidFinish function to your class to act as a call-back:

```
- (void) BardecodeDidFinish:(NSNotification*)notification {
  if ([[notification name] isEqualToString:@"BarcodeDidRead"])
  {
            // Barcode read OK
            // Barcode value is [barcode Barcodevalue]
            // Barcode type is [barcode BarcodeType]
  }
  else if ([[notification name] isEqualToString:@"BarcodeDidNotRead"])
  {
    // Barcode not found
  }
  else if ([[notification name] isEqualToString:@"UserDidCancel"])
  {
    // User canceled operation
  }
}
```

And use the following code to launch the barcode reader from your class:

```
[barcode ScanBarcodeFromViewFinder];
```

5 Properties of the Bardecode class

5.1 Properties unique to the Bardecode class

Name	Type	Comment
BarcodeType	NSString *	Type of barcode e.g "CODE39"
BarcodeValue	NSString *	Value of barcode
Delegate	UIViewController *	Class that supports the BardecodeDidFinish function
ViewForPreview	UIView *	View to use when creating the preview view

5.2 General SDK properties

The Bardecode class also supports most of the properties common to all versions of the Softek Barcode Reader SDK. For example, the ReadDatamatrix property controls whether or not the SDK will scan for datamatrix barcodes. Full details of all the SDK properties can be found here.

6 Supported Barcode Formats

6.1 1-D Barcode Formats

The following 1-D barcode formats are support by the SDK (with corresponding properties given in brackets):

- Codabar also known as Code 2 of 7, Codeabar, Ames Code, NW-7 and Monarch (ReadCodabar)
- Code 128 Symbol Sets A, B and C (ReadCode128)
- Code 128 Short Format (ReadShortCode128)
- Code 2 of 5 Datalogic (ReadCode25ni)
- Code 2 of 5 Iata1 (ReadCode25ni)
- Code 2 of 5 Iata2 (ReadCode25ni)
- Code 2 of 5 Industrial (ReadCode25ni)
- Code 2 of 5 Interleaved (ReadCode25)
- Code 2 of 5 Matrix (ReadCode25ni)
- Code 3 of 9 (ReadCode39)
- Code 3 of 9 Extended (ReadCode39 and ExtendedCode39)
- Code 93 (ReadCode93)
- EAN-8, European Article Number/International Article Number (ReadEAN8)
- EAN-13 and UPC-A, European Article Number/International Article Number (ReadEAN13)
- GS1-128, UCC-128, EAN-128 (ReadCode128)
- GS1-Databar (please see 2-D section below)
- Patch Code Symbols (ReadPatchCodes)
- UPC-A, Universal Product Code (ReadEAN13 and ReadUPCA)
- UPC-E, Universal Product Code (ReadUPCE)

6.2 2-D and Stacked Barcode Formats

The following 2-D and stacked barcode formats are also supported:

- Data Matrix ECC200 sizes 8x8 to 144x144 (ReadDataMatrix)
- GS1-Databar or Reduced Space Symbology. Omnidirectional, Stacked Omnidirectional, Expanded, Expanded Stacked and Limited (ReadDatabar)
- Micro-PDF-417 (ReadMicroPDF417)
- PDF-417, Portable Data File (ReadPDF417)
- QR-Code (ReadQRCode)

7 Appendix B: Properties Reference

AllowDuplicateValues	allow duplicate barcode values on the same page
BitmapResolution	set the resolution of a bitmap
CodabarMaxVariance	max width variance for codabar characters
Code25Checksum	handle final character of code 25 barcode as checksum
Code39Checksum	handle final character of code 39 barcode as checksum
Code39NeedStartStop	expect start/stop characters with code 39 barcodes
ColorChunks	divide scan lines into sections for threshold levels
ColorProcessingLevel	control the time spent processing a color image
ColorThreshold	set the color threshold level for a color image
ConvertUPCEToEAN13	automatically convert UPC-E format to EAN-13
DatabarOptions	set options for GS1 Databar barcodes
Despeckle	remove speckled marks from an image before scanning
Encoding	set the encoding method for barcode values
ErrorCorrection	attempt to correct errors
ExtendedCode39	assume extended code 39 barcode format
GammaCorrection	set gamma correction level for color images
LicenseKey	set the license key
LineJump	control the frequency of line sampling
MaxLength	set the maximum length for a barcode
MedianFilter	perform a median filter on the image before scanning
MinLength	set the minimum length for a barcode
MinOccurrence	specify the lowest permitted score for a barcode
MinSeparation	minimum distance between barcodes of same value
MinSpaceBarWidth	minimum size of a space between bars
MultipleRead	scan for more than one barcode
NoiseReduction	perform noise reduction before scanning

PageNo	set the page number to scan in a muli-page image
PatchCodeMinOccurrence	minimum score for a Patch Code barcode.
Pattern	only report barcodes that fit the specified pattern
Pdf417Debug	enable debug mode for PDF-417 barcodes
Photometric	set photometric interpretation for bi-tonal bitmaps
PrefOccurrence	specify the preferred score for a barcode
QuietZoneSize	set the size of the quiet zone around a barcode
ReadCodabar	scan for codabar barcodes
ReadCode128	scan for code-128 barcodes
ReadCode25	scan for code-25 barcodes
ReadCode25ni	scan for non-interleaved code-25 barcodes
ReadCode39	scan for code-39 barcodes
ReadCode93	scan for code-93 barcodes
ReadDatabar	scan for databar barcodes
ReadDataMatrix	scan for datamatrix barcodes
ReadEAN13	scan for ean-13 barcodes
ReadEAN8	scan for ean-8 barcodes
ReadMicroPDF417	scan for micro pdf-417 barcodes
ReadNumeric	only read numeric barcodes
ReadPatchCodes	scan for patch codes
ReadPDF417	scan for pdf-417 barcodes
ReadQRCode	scan for QR-Codes
ReadShortCode128	scan for short code-128 barcodes
ReadUPCA	scan for upc-a barcodes
ReadUPCE	scan for upc-e barcodes
ScanDirection	specify the orientations in which to scan
ShortCode128MinLength	set the minimum length for a short code-128 barcode

ShowCodabarStartStop	include codabar start/stop characters
ShowCheckDigit	display check digits where possible
SkewLineJump	frequency of line sampling for skewed barcodes
SkewTolerance	maximum tolerance for skewed barcodes
UseOldCode128Algorithm	use the old code-128 detection method
UseOverSampling	process multiple scan lines at the same time
UseRunCache	use a memory cache for run-length information
WeightLongerBarcodes	accept lower scores for longer barcodes

7.1 AllowDuplicateValues

Overview

The AllowDuplicateValues can be used to stop the toolkit from reporting duplicate barcodes on the same page in an image. This can be useful for images where the middle section of a barcode is badly damaged or missing. With the property set to TRUE the toolkit may report that there are 2 barcodes of the same type and value. With the property set to FALSE it would assume that the 2 barcodes were part of a single barcode and set the bounding rectangle accordingly.

Type:	BOOL
Default value:	TRUE
See also:	MinSeparation

7.2 BitmapResolution

Overview

BitmapResolution is the resolution of the bitmap to be scanned in ScanBarCodeFromBitmap, in dots per inch. This value only effects the expected size of the quiet area around a barcode and for most images can be left to the default value.

Type:	SHORT
Default value:	200

7.3 CodabarMaxVariance

Overview

CodabarMaxVariance is the maximum percentage variance that a character in a codabar barcode can have from the average for that barcode.

Type: SHORT
Default value: 20

Note: This is an **Advanced property** and can only be set using LoadXMLSettings

See also: ReadCodabar

7.4 Code25Checksum

Overview

When True the toolkit will only report Code 25 barcodes where the last character is a valid checksum for the rest of the barcode. The toolkit expects a Code 25 checksum to be calculated using the following method:

Sum all of the even positioned characters (the right hand message character is always even), and multiply by 3.

Sum all the odd positioned characters.

Sum the totals from steps 1 and 2.

The checksum is the smallest number that when added to this sum results in a multiple of 10.

If the resulting number of characters is odd and you are using Interleaved Code 2 of 5 then add a leading 0 to the message data.

Type: BOOL
Default value: FALSE

See also: ReadCode25

7.5 Code39Checksum

Overview

When True the toolkit will only report Code 39 barcodes where the last character is a valid checksum for the rest of the barcode. The toolkit expects a Code 39 checksum to be calculated using modulus-43.

The following table shows the character and value used for the calculation...

Char	Value	Char	Value	Char	Value	Char	Value
0	0	B	11	M	22	X	33

1	1	C	12	N	23	Y	34
2	2	D	13	O	24	Z	35
3	3	E	14	P	25	-	36
4	4	F	15	Q	26	.	37
5	5	G	16	R	27	space	38
6	6	H	17	S	28	$	39
7	7	I	18	T	29	/	40
8	8	J	19	U	30	+	41
9	9	K	20	V	31	%	42
A	10	L	21	W	32		

e.g

Data = 12345ABCDE+

Sum of values: 1 + 2 + 3 + 4 + 5 + 10 + 11 + 12 + 13 + 14 + 41 = 116
115 / 43 = 2 rem 30, so U is the check digit.
Data and check digit = 12345ABCDE+U

Type: BOOL
Default value: FALSE

See also: ReadCode39

Code39NeedStartStop

ExtendedCode39

7.6 Code39NeedStartStop

Overview

When set to TRUE the toolkit will only report Code 39 barcodes that start and end with a * character.

Setting this property to FALSE is not recommended for the following reasons:

It is not a valid Code 39 barcode without the start and stop * character.

Without a start/stop * character, a Code 39 barcode reads with 2 different values, left to right, and right to left. The toolkit will report it as 2 different barcodes unless the scan direction is restricted to one direction only.

The probablility of a false positive reading is increased significantly by setting this property to FALSE.

Type: BOOL
Default value: TRUE

7.7 ColorChunks

Overview

ColorChunks specifies how many sections a scan line of an image should be broken into when calculating threshold levels for black and white pixels.

Type: SHORT
Default value: 1

Note: This is an **Advanced property** and can only be set using LoadXMLSettings

See also: ColorProcessingLevel

7.8 ColorProcessingLevel

Overview

The ColorProcessingLevel property controls the amount of processing time spent reading barcode values from color images. Values range from 0 to 5, with a default of 2. A low value will process color images faster but accuracy and read-rate levels will be lower than if a high value is used.

Please note that setting the ColorThreshold property to a non-zero value effectively sets ColorProcessingLevel to 0.

Type: SHORT
Default value: 2

See also: ColorChunks

ColorThreshold

7.9 ColorThreshold

Overview

ColorThreshold is the color value used by the control to decide whether a pixel should be considered to be black or white. The value should be in the range 0 to 255.

Please note that if this property is set to a non-zero value than ColorProcessingLevel is effectively set to a value of 0. It is recommended to set this property to 0 and control the accuracy of reading from color images through the ColorProcessingLevel property.

Type:	SHORT
Default value:	0
See also:	ColorProcessingLevel

7.10 ConvertUPCEToEAN13

Overview

A UPC-E barcode is actually an EAN-13/UPC-A barcode that has had certain digits removed to create an 8 digit number. Only certain EAN-13/UPC-A barcodes can go through this process. For example, the UPC-A barcode "023456000073 " can be suppressed to the UPC-E value "02345673" and restored to it's original value by the barcode reader. The Softek barcode Reader SDK can interpret a UPC-E barcode in either format via the ConvertUPCEToEAN13 property.

When set to TRUE the toolkit will convert type UPC-E barcodes into EAN-13 format.

Type:	BOOL
Default value:	TRUE
See also:	ReadUPCE
	ReadEAN13
	ReadUPCA

7.11 DatabarOptions

Overview

The DatabarOptions property can be used to set various options for GS1 Databar recognition. All options are turned on by default, but some applications may find it useful to disable certain features for performance reasons.

The property works as a mask and can be constructed from the following values:

1	Read the supplementary 2-D portion if indicated by the linkage flag.
2	Read RSS-14 barcodes

4	Read RSS-14 Stacked barcodes
8	Read RSS-Limited barcodes
16	Read RSS-Expanded barcodes
32	Read RSS-Expanded Stacked barcodes

Type: SHORT
Default value: 255

See also: ReadDatabar

7.12 Despeckle

Overview

If the **Despeckle** property is set to TRUE and the NoiseReduction property is none zero, then the toolkit removes white speckles inside the bars of a barcode before removing black marks from the spaces between bars.

Type: BOOL
Default value: FALSE

See also: NoiseReduction

7.13 Encoding

Overview

The Encoding property controls the format in which the toolkit returns strings for barcode types that use full symbol sets such as PDF-417.

The property can take any of the following values:

0	Raw , with null characters suppressed.
1	Quoted printable
2	Unicode
3	UTF-8

Type: SHORT
Default Value: 0

7.14 ErrorCorrection

Overview

Some barcodes cannot be read because the process of scanning or faxing has split or merged bars together. When ErrorCorrection is set to True to toolkit will, where possible, make a best guess at such barcodes.

Note that this property currently only applies to Code 39 and Code 39 Extended barcodes.

Type:	BOOL
Default value:	FALSE

See also: ReadCode39

7.15 ExtendedCode39

Overview

A Code 39 barcode can be used to represent the entire ASCII-128 symbol set by using 2 normal Code 39 characters to represent one character in the ASCII-128 symbol set. A barcode reader cannot distinguish between normal and extended Code 39 barcodes and so the ExtendedCode39 property must be set to TRUE when reading barcodes encoded using the extended symbol set. Note that the ReadCode39 property must also be set to TRUE.

If the toolkit is unable to decode the string in the extended symbol set then it is left as a normal Code 39 barcode.

Type:	BOOL
Default value:	FALSE

See also: ReadCode39

7.16 GammaCorrection

Overview

If GammaCorrection is set to a value other than 100 then the toolkit will apply gamma correction to a color image. The amount of gamma correction is equal to GammaCorrection / 100. For example, to achieve a gamma correction of 0.5 the property should be set to a value of 50.

Type:	SHORT
Default value:	100

7.17 LicenseKey

Overview

Use the LicenseKey property to set your license key prior to calling the ScanBarCode, ScanBarCodeFromBitmap or ScanBarCodeFromDIB functions. With no license key the .net interface will return all barcode values as “Please contact sales@bardecode.com for a trial license string" and other interfaces will display a pop up box that the user will need to click on to continue.

Type:	STRING
Default value:	“”

7.18 LineJump

Overview

The LineJump property controls the frequency with which the toolkit samples scan lines as it moves through an image. Increasing the value of the LineJump property will increase the speed at which an image is processed but may decrease the read rate. The SkewLineJump property is used in a similar way when searching for skewed barcodes.

Type:	SHORT
Default value:	1

See also: SkewLineJump

7.19 MaxLength

Overview

MaxLength defines the largest length for a barcode string, including checksum characters.

Type:	SHORT
Default value:	999

See also: MinLength

7.20 MedianFilter

Overview

When TRUE the toolkit will apply a median filter to the image before checking for barcodes. This is a useful option for high resolution images that contain speckles of black and white. It is not recommended for images where the black bars or white spaces are less than 2 pixels wide.

Type:	BOOL
Default Value:	FALSE

7.21 MinLength

Overview

MinLength defines the smallest length for a barcode string, including checksum characters.

Type: SHORT
Default value: 4

See also: MaxLength

7.22 MinOccurrence

Overview

Please refer to PrefOccurrence for more information.

Type: SHORT
Default value: 2

7.23 MinSeparation

Overview

MinSeparation defines the minimum distance between barcodes of identical value and vertical alignment in 1/300th of an inch. If the distance between two barcodes of same value and on the same alignment is less than MinSeparation then the toolkit assumes that it is a single barcode that has been split into 2 parts by a problem in the scanning process.

Type: SHORT
Default value: 180

7.24 MinSpaceBarWidth

Overview

MinSpaceBarWidth is the minimum acceptable size for a space between the bars in a barcode. When set to a value of 0 the toolkit will automatically select the best value. Spaces that are smaller than the value used are ignored.

Type: SHORT
Default value: 0

7.25 MultipleRead

Overview

Normally the toolkit stops at the first positive match for a barcode. When **MultipleRead** is TRUE the toolkit will check the entire image for barcode strings and record each positive match.

Type: BOOL
Default value: FALSE

7.26 NoiseReduction

Overview

If the **NoiseReduction** property is none zero then the toolkit will run an image through a noise reduction filter before scanning for barcodes. The filter removes marks from an image that are unlikely to be part of a barcode. A larger value for NoiseReduction will remove larger marks from the image, but may also destroy vital barcode information. A typical value for **NoiseReduction** is 10.

Type: SHORT
Default value: 0

See also: Despeckle

7.27 PageNo

Overview

PageNo is a 1 based index that specifies the page to be scanned in an image. A value of zero indicates that every page will be scanned

Type: SHORT
Default value: 0

7.28 PatchCodeMinOccurrence

Overview

Please refer to PrefOccurrence for more information.

Type: SHORT
Default value: 30

7.29 Pattern

Overview

The Pattern property is a regular expression that each barcode found in an image is compared against. The toolkit will only return barcodes that match the pattern.

The toolkit use POSIX extended regular expression syntax.

Examples:

"ABCDEF" will match all barcodes containing "ABCDEF" (e.g "XYZABCDEFXYZ") .

"ABC[0-9]+" will match all barcodes containing "ABC" followed by one or more digits (e.g XYZABC71827XYZ").

"^ABC[0-9]+$" will match barcodes that only consist of "ABC" followed by one or more digits (e.g "ABC12345").

Note that if a Code 39 barcode uses a checksum character and the Pattern property is used to specify the entire string (ie. the last character of the pattern is $) then the Code39Checksum property must also be set to True.

Type:	STRING
Default value:	NULL

See also:	ReadNumeric

7.30 Pdf417Debug

Overview

Output information about the structure and cluster values of the barcode. The debug information is returned in place of the normal barcode value (see GetBarString).

The fields of the string are as follows:

Error correction status – true or false
Number of data columns
Number of rows
Error correction level
Number of unknown cluster values
And for each cluster: cluster value (score)

Type: BOOL
Default value: FALSE

7.31 Photometric

Overview

The Photometric property determines how the toolkit interprets a pixel value in a bi-tonal bitmap passed to the ScanBarCodeFromBitmap method.

	Pixel Value = 0	Pixel Value = 1
Photometric = 0	Black	White
Photometric = 1	White	Black

This property is not used with the ScanBarCode or ScanBarCodeFromDIB methods.

Type: SHORT
Default value: 0

See also: ScanBarCodeFromBitmap

7.32 PrefOccurrence

Overview

As the SDK scans an image it assigns a score to each barcode candidate. At the end of a scan, any candidates with a score >= PrefOccurrence are reported by the SDK. If no candidate meets this criteria then the SDK selects the candidate with the highest score and reports this barcode if it has a score >= MinOccurrence. Note that Patch Codes are only ever reported if the score is >= PatchCodeMinoccurrence.

Type: SHORT
Default value: 5

See also: MinOccurrence

PatchCodeMinOccurrence

7.33 QuietZoneSize

Overview

When the toolkit checks for a barcode on a scan line in an image, it ignores those parts of the line that are not preceded by the number of white pixels specified by QuietZoneSize. When the property has a value of 0 then the quiet zone is calculated 1/10th of the value of the image resolution (e.g. 10 pixels in a 100 dpi image).

Type: SHORT
Default value: 0

7.34 ReadCodabar

Overview

When set to TRUE the toolkit will search for codabar barcodes and the string returned by GetBarStringType will be set to “CODABAR”.

Type: BOOL
Default value: TRUE

See also: CodabarMaxVariance

7.35 ReadCode128

Overview

When set to TRUE the toolkit will search for type 128 barcodes and the string returned by GetBarStringType will be set to CODE128.

Type: BOOL
Default value: TRUE

See also: ReadShortCode128

UseOldCode128Algorithm

7.36 ReadCode25

Overview

When set to TRUE the toolkit will search for type 2 of 5 interleaved barcodes and the string returned by GetBarStringType will be set to “CODE25”.

Type: BOOL
Default value: TRUE

See also: ReadCode25ni

Code25Checksum

7.37 ReadCode25ni

Overview

When set to TRUE the toolkit will search for type 2 of 5 non-interleaved barcodes in the following formats:

- Code 2 of 5 Datalogic
- Code 2 of 5 Iata1
- Code 2 of 5 Iata2
- Code 2 of 5 Industrial
- Code 2 of 5 Interleaved
- Code 2 of 5 Matrix

The string returned by GetBarStringType will be set to “CODE25”.

Type: BOOL
Default value: FALSE

7.38 ReadCode39

Overview

When set to TRUE the toolkit will search for type 39 barcodes and the string returned by GetBarStringType will be set to “CODE39”.

Type:	BOOL
Default value:	TRUE

See also: Code39Checksum

Code39NeedStartStop

ExtendedCode39

7.39 ReadCode93

Overview

When set to TRUE the toolkit will search for type 93 barcodes and the string returned by GetBarStringType will be set to “CODE93”.

Type: BOOL
Default value: FALSE

7.40 ReadDatabar

Overview

When set to TRUE the toolkit will search for GS1 Databar barcodes and the string returned by GetBarStringType will be set to “DATABAR”. The following types of GS1 Databar are supported:

RSS-14

RSS-14 Truncated

RSS-14 Stacked

RSS-14 Stacked Omnidirectional

RSS Limited

RSS Expanded

RSS Expanded Stacked

Please note the the bounding rectangle for stacked versions of the barcode currently only includes either the top-most or bottom-most element of the stack.

Reading supplementary data

Some GS1 Databar barcodes encode supplementary data in the form of a micro-PDF-417 barcode above the linear portion of the barcode. To read the supplementary portion set ReadMicroPDF417 to True and ensure that DatabarOptions includes the option to read supplementary barcodes.

Type:	BOOL
Default value:	FALSE

See also: ReadMicroPDF417

DatabarOptions

7.41 ReadDataMatrix

Overview

When set to TRUE the toolkit will search for DataMatrix (ECC 200) barcodes and the string returned by GetBarStringType will be set to “DATAMATRIX”.

Type:	BOOL
Default value:	FALSE

7.42 ReadEAN13

Overview

When set to TRUE the toolkit will search for EAN-13 type barcodes and the string returned by GetBarStringType will be set to “EAN13”.

Type:	BOOL
Default value:	TRUE

7.43 ReadEAN8

Overview

When set to TRUE the toolkit will search for EAN-8 type barcodes and the string returned by GetBarStringType will be set to “EAN8”.

Type:	BOOL
Default value:	TRUE

7.44 ReadMicroPDF417

Overview

When set to TRUE the toolkit will search for micro-PDF-417 barcodes and the string returned by GetBarStringType will be set to “PDF417”.

Type:	BOOL
Default value:	FALSE

7.45 ReadNumeric

Overview

When True the toolkit will only report numeric barcodes. Note that this the same as setting the Pattern property to the value "^[0-9]+$".

Type:	BOOL
Default value:	FALSE

7.46 ReadPatchCodes

Overview

When set to TRUE the toolkit will search for patch code barcodes and the string returned by GetBarString will be set to PATCH.

Type:	BOOL
Default value:	FALSE

7.47 ReadPDF417

Overview

When set to TRUE the toolkit will search for PDF-417 barcodes and the string returned by GetBarString will be set to “PDF417”.

Type:	BOOL
Default value:	FALSE

7.48 ReadQRCode

Overview

When set to TRUE the toolkit will search for QR-Codes and the string returned by GetBarString will be set to “QRCODE”.

- All version sizes are supported, however best results will always be obtained when using smaller version sizes with maximum error correction.
- The Kanji symbol set is not currently supported.
- Skewed QR-Codes can be read using the default value for SkewTolerance. Changing the value of SkewTolerance will have no effect on the scanning of QR-Codes.
- For best results set ColorProcessingLevel to a value of 1.
- GetBarStringDirection will always return a value of 1 for QR-Codes.
- If only QR-Codes need to be recognized then set ScanDirection to a value of 1.

Type: BOOL
Default value: FALSE

7.49 ReadShortCode128

Overview

When set to TRUE the toolkit will search for Code 128 barcodes of symbol set C, without the normal start and stop characters. The barcode type for these barcodes is set to "SHORTCODE128".

Type:	BOOL
Default value:	FALSE

7.50 ReadUPCA

Overview

When set to TRUE the toolkit will search for UPC-A type barcodes.

UPC-A barcodes are a subset of EAN-13. For example, the UPC-A barcode "016000336100" is the same as the EAN-13 barcode "0016000336100". In fact, UPC-A barcodes are the sub-set of EAN-13 barcodes that start with a 0. The ReadEAN13 property controls whether any barcodes of type EAN-13 are recognized - and this includes UPC-A, whether or not ReadUPCA is set to true. The effect of the ReadUPCA flag is to control whether an EAN-13 barcode that starts with a 0 is returned as a 12 digit UPC-A or as a 13 digit EAN-13 barcode. The string returned by GetBarString will be set to either “UPCA” or “EAN13”.

Type:	BOOL
Default value:	FALSE

7.51 ReadUPCE

Overview

When set to TRUE the toolkit will search for UPC-E type barcodes.

A UPC-E barcode is actually an EAN-13/UPC-A barcode that has had certain digits removed to create an 8 digit number. Only certain EAN-13/UPC-A barcodes can go through this process.

For example, the UPC-A barcode "023456000073" can be suppressed to the UPC-E value "02345673" and restored to its original value by the barcode reader. The Softek barcode Reader SDK can interpret a UPC-E barcode in either format via the ConvertUPCEToEAN13 property. The string returned by GetBarStringType will be set to “UPCE”.

Type: BOOL
Default value: FALSE

7.52 ScanDirection

Overview

ScanDirection is a mask that controls the directions in which the barcode reader will look for barcodes in an image, and is built from the following values:

1 = Left to Right

2 = Top to Bottom

4 = Right To Left

8 = Bottom to Top

For example, a value of 5 (1 + 4) means that the reader will look for barcode from left to right and right to left.

Note: This property replaces the Rotation property used in previous versions.

Type: SHORT
Default value: 15

7.53 ShortCode128MinLength

Overview

ShortCode128MinLength defines the smallest length for a barcode string, including checksum characters.

Type: SHORT
Default value: 2

7.54 ShowCodabarStartStop

Overview

Include the start and stop characters when returning the value of a codabar barcode.

Type: BOOL
Default value: TRUE

7.55 ShowCheckDigit

Overview

When set to TRUE the OCX will include the barcode check digit in the returned string.

Note: This property only applies to barcode types with built in check digits (e.g Code 128).

Type: BOOL
Default value: FALSE

7.56 SkewLineJump

Overview

SkewLineJump works in a similar way to the LineJump property, but only effects the phase of the scanning process concerned with searching for skewed barcodes. It can be useful to set the 2 properties to different values for reasons of performance.

Type: SHORT
Default value: 9

7.57 SkewTolerance

Overview

SkewTolerance controls the maximum angle from the horizontal or vertical at which a barcode will be recognised by the toolkit. The table below shows the possible values for this property along with the approximate maximum angles:

0 = up to 5 degrees

1 = 13 degrees

2 = 21 degrees

3 = 29 degrees

4 = 37 degrees

5 = 45 degrees

Type: SHORT
Default value: 0

7.58 UseOldCode128Algorithm

Overview

Use the Code 128 detection algorithm as used in earlier versions of the toolkit (pre version 7.3.1).

Type: BOOL
Default value: FALSE

7.59 UseOverSampling

Overview

When UseOverSampling is TRUE the barcode reader samples 3 lines at a time (skipping 2 lines between each sample) and takes the average pixel value. This is useful for images containing both black and white speckles.

Type: BOOL
Default value: FALSE

7.60 UseRunCache

Overview

Use a memory cache for run-length information derived from an image.

Type: BOOL
Default value: TRUE

7.61 WeightLongerBarcodes

Overview

When WeightLongerBarcodes is TRUE the barcode reader will weight the counts used with the PrefOccurrence and MinOccurrence properties according to the length and type of the barcode in question. Barcode types using built in checksums are favoured above barcode types with no checksum.

Type: BOOL
Default value: TRUE

8 Appendix D: Release Notes

Version 7.5.1

Added QR-Code recognition. Note that this does not yet support the Kanji symbol set.

Version 7.4.2

1. Improvements to barcode recognition

Improvements have been made to the Code 39, Code 128, GS1-Databar and Datamatrix modules.

2. Code 93 Support

Support for Code 93 barcodes has also been added. Please refer to the manual page for further details.

3. PDF-417 decoding

An error has been fixed that could lead to corruption in the output values for some PDF-417 barcodes.

4. Partial reads on PDF-417 barcodes

It was possible for certain PDF-417 barcodes to give partial and incorrect readings. This has been corrected. The maximum length for a PDF-417 barcode has now been increased from 2K to 8K (when represented in quoted printable format).

5. Added most standard SDK properties to the Bardecode class.

Version 7.3.1j

1. Added support for rotating views to the Bardecode class. Please note the following if you wish to support rotating views in your app:

Override the didRotateFromInterfaceOrientation method in your view controller and call the ResizePreview method of the Bardecode class. e.g:

```
- (void)didRotateFromInterfaceOrientation:(UIInterfaceOrientation)fromInterfaceOrientation {

        [barcode ResizePreview];

}
```

2. If an image called video_preview_bg.png exists in your projects resources then it will be used as a background image for the video preview. The image used in the iBardecode sample project is 320X480 pixels solid black.

3. If an image called video_preview_ov.png exists in your project resources then it will be used as an overlay for the video preview.

Version 7.3.1i

Worked around a bug in the AVCaptureVideoPreviewLayer class in IOS 4.1. The bug made the preview layer freeze at the point of initialisation if the output device had already been added to the capture session.

Added a property called WaitForFocus which defaults to false.

Version 7.3.1h

Bardecode class for IOS 4:

Fixed short delay in sample project when barcode is detected (thanks to Oli Kessler of NCode for the idea).

Prevented class from sampling frames while auto-focus is taking place. The text on the cancel button changes to blue when the class is sampling frames and reverts to black when auto-focus is in operation.

Ensured that BarcodeDidRead can only be fired once for any single video capture session.

Reduced the size of the quiet zone to cope with tight borders around barcodes.

Increased the required hit-counts for barcodes to reduce chances of false positive reads.

Added a new property called viewForPreview which can be used to specify the UIView that the Bardecode class should use for overlaying the video preview and the cancel buttons. For example, the sample project has the following code:

```
barcode = [Bardecode alloc] ;
[barcode init] ;
barcode.Delegate = self ;
[barcode setViewForPreview:self.view] ;
[super viewDidLoad];
```

This may be useful if the delegate class is not a view controller.