

Softek Software Ltd

Softek Barcode Reader Toolkit for Android

Product Documentation



V7.5.1

1 Contents

2	Overview	1
3	Installation	1
4	Calling Bardecoder from another App	1
5	Settings for the Bardecoder App.....	3
5.1	Property Settings	3
6	Using the Java Interface.....	5
7	Supported Barcode Formats	6
7.1	1-D Barcode Formats	6
7.2	2-D and Stacked Barcode Formats.....	6
8	Appendix B: Properties Reference	7
8.1	AllowDuplicateValues	9
8.2	BitmapResolution.....	9
8.3	CodabarMaxVariance.....	9
8.4	Code25Checksum.....	10
8.5	Code39Checksum.....	10
8.6	Code39NeedStartStop	11
8.7	ColorChunks	12
8.8	ColorProcessingLevel	12
8.9	ColorThreshold.....	12
8.10	ConvertUPCEToEAN13	13
8.11	DatabarOptions.....	13
8.12	Despeckle	14
8.13	Encoding.....	14
8.14	ErrorCorrection	15
8.15	ExtendedCode39.....	15
8.16	GammaCorrection.....	15
8.17	LicenseKey.....	15
8.18	LineJump	16
8.19	MaxLength	16
8.20	MedianFilter.....	16
8.21	MinLength	17
8.22	MinOccurrence	17
8.23	MinSeparation	17

8.24	MinSpaceBarWidth	17
8.25	MultipleRead.....	17
8.26	NoiseReduction.....	18
8.27	PageNo.....	18
8.28	PatchCodeMinOccurrence	18
8.29	Pattern	18
8.30	Pdf417Debug.....	19
8.31	Photometric	19
8.32	PrefOccurrence	20
8.33	QuietZoneSize	20
8.34	ReadCodabar.....	20
8.35	ReadCode128.....	21
8.36	ReadCode25	21
8.37	ReadCode25ni.....	21
8.38	ReadCode39	22
8.39	ReadCode93.....	22
8.40	ReadDatabar	22
8.41	ReadDataMatrix.....	23
8.42	ReadEAN13	23
8.43	ReadEAN8	23
8.44	ReadMicroPDF417	24
8.45	ReadNumeric.....	24
8.46	ReadPatchCodes	24
8.47	ReadPDF417.....	24
8.48	ReadQRCode	24
8.49	ReadShortCode128	25
8.50	ReadUPCA	25
8.51	ReadUPCE.....	25
8.52	ScanDirection.....	26
8.53	ShortCode128MinLength.....	26
8.54	ShowCodabarStartStop.....	26
8.55	ShowCheckDigit	27
8.56	SkewLineJump.....	27
8.57	SkewTolerance.....	27

8.58	UseOldCode128Algorithm	28
8.59	UseOverSampling.....	28
8.60	UseRunCache	28
8.61	WeightLongerBarcodes.....	28
9	Appendix D: Release Notes.....	30
9.1	Version 7.4.1	30

2 Overview

The Softek Barcode Reader Toolkit for Android allows you to integrate barcode reading into your app using either a ready made helper app called Bardecoder or via a Java class and JNI library.

The Bardecoder app provides a simple interface that allows the calling app to configure the SDK and collect the results from a scan. The SDK includes a sample eclipse project called BardecodeExampleApp that shows how to do this.

The Java class and JNI library provide functions to scan a bitmap image for a barcode. The source code for the Bardecoder app is included with the SDK and this demonstrates how to sample frames from the video camera and pass them to the Java class for scanning.

3 Installation

The Bardecoder app need only be installed on a device if the simpler interface will be used (calling the Bardecoder app from your application).

The Bardecoder app can be installed on a target device with the adb command as follows:

```
adb install Bardecoder.apk
```

Note that the adb command is part of the [Android SDK](#).

4 Calling Bardecoder from another App

All the methods discussed in this section are demonstrated in the sample Eclipse project called BardecodeExampleApp.

To launch the barcode scanner activity from an app an Intent object needs to be created as follows:

```
Intent myIntent;  
myIntent = new Intent();  
myIntent.setComponent(new ComponentName("com.softeksoftware.bardecoder",  
    "com.softeksoftware.bardecoder.scanner"));
```

The putExtra method can then be used to assign various settings for the activity:

```
myIntent.putExtra("com.softeksoftware.bardecoder.scanner.name_of_setting", value);
```

...and the activity launched using the startActivityForResult method:

```
startActivityForResult(myIntent, 0);
```

The result of the scanning process is handled by over-riding the onActivityResult method:

```
// Process the result of the bar-code reading  
public void onActivityResult(int requestCode, int resultCode, Intent data)  
{  
    super.onActivityResult(requestCode, resultCode, data);  
    if (data != null)  
    {  
        Bundle bundle = data.getExtras();  
        // Get the value of the bar-code  
        value = bundle.getString("com.softeksoftware.bardecoder.scanner.value");  
    }  
}
```

} }

5 Settings for the Barcode App

The following settings are currently supported by the toolkit:

Name	Description	Default
beepOnSuccess	Play a beep sound when barcode is scanned	false
finishOnSuccess	Finish scanning activity when barcode is scanned	false
licenseKey	License key for the toolkit – if no license key is supplied then the final 3 character of a barcode will be returned as the string ???	""
minimumWidth	Minimum width of barcode measured in pixels	100
settings	XML property settings for the toolkit (see Chapter ? for more details)	""
useTorch	Turn on the torch while scanning	false
vibrateOnSuccess	Vibrate the phone when a barcode is scanned	false

Please refer to section 4 for details of how to apply settings.

5.1 Property Settings

Property settings control the underlying barcode reading engine and are common with settings found in version of the SDK for other platforms such as Windows and Linux. Appendix ? contains a list and description for each property.

Property values are passed in XML format via a string assigned to the key called:

```
com.softeksoftware.barcode.scanner.settings
```

In the BarcodeExampleApp, the settings are stored in the file res/raw/barcode.xml, which is loaded into a string using the loadSettings() method.

An example XML file is shown below:

```
<xml version='1.0' encoding='iso-8859-1'>
  <SoftekBarcode>
    <Properties>
      Settings for product label barcodes
      <ReadEAN8>1</ReadEAN8>
      <ReadEAN13>1</ReadEAN13>
      <ReadUPCA>1</ReadUPCA>
      <ReadUPCE>1</ReadUPCE>
      <ConvertUPCEToEAN13>0</ConvertUPCEToEAN13>
      <ReadDatabar>1</ReadDatabar>

      Settings for other 1-D barcodes types
      <ReadCodabar>0</ReadCodabar>
      <ReadCode128>1</ReadCode128>
      <ReadCode25ni>0</ReadCode25ni>
      <ReadCode25>1</ReadCode25>
      <ReadCode39>1</ReadCode39>
      <ReadCode93>0</ReadCode93>
      <ReadPatchCodes>0</ReadPatchCodes>

      Settings for stacked and 2-D barcode types
      <ReadDataMatrix>0</ReadDataMatrix>
      <ReadMicroPDF417>0</ReadMicroPDF417>
      <ReadPDF417>0</ReadPDF417>
      <ReadQRCode>0</ReadQRCode>

      Other settings
      <ReadNumeric>0</ReadNumeric>
      <CodabarMaxVariance>20</CodabarMaxVariance>
      <Code25Checksum>0</Code25Checksum>
    </Properties>
  </SoftekBarcode>
</xml>
```

```
<Code25MinOccurrenceLength>5</Code25MinOccurrenceLength>
<Code39Checksum>0</Code39Checksum>
<Code39NeedStartStop>1</Code39NeedStartStop>
<ColorChunks>4</ColorChunks>
<ColorProcessingLevel>2</ColorProcessingLevel>
<ColorThreshold>0</ColorThreshold>
<Despeckle>0</Despeckle>
<Encoding>0</Encoding>
<ErrorCorrection>0</ErrorCorrection>
<ExtendedCode39>0</ExtendedCode39>
<GammaCorrection>100</GammaCorrection>
<LineJump>1</LineJump>
<MaxLength>999</MaxLength>
<MedianFilter>0</MedianFilter>
<MinSeparation>180</MinSeparation>
<MinLength>4</MinLength>
<MinOccurrence>2</MinOccurrence>
<MinSpaceBarWidth>0</MinSpaceBarWidth>
<MultipleRead>0</MultipleRead>
<NoiseReduction>0</NoiseReduction>
<PatchCodeMinOccurrence>30</PatchCodeMinOccurrence>
<Pattern></Pattern>
<Pdf417Debug>0</Pdf417Debug>
<Photometric>0</Photometric>
<PrefOccurrence>5</PrefOccurrence>
<ShortCode128MinLength>2</ShortCode128MinLength>
<SkewLineJump>9</SkewLineJump>
<SkewTolerance>0</SkewTolerance>
<ShowCheckDigit>1</ShowCheckDigit>
<ShowCodabarStartStop>1</ShowCodabarStartStop>
<SkewSpeed>3</SkewSpeed>
<UseOldCode128Algorithm>0</UseOldCode128Algorithm>
<UseRunCache>1</UseRunCache>
<UseOverSampling>0</UseOverSampling>
<WeightLongerBarcodes>1</WeightLongerBarcodes>
  </Properties>
</SoftekBarcode>
</xml>
```


6 Using the Java Interface

The Java interface should be used when apps require a lower level of integration to the SDK than the Barcode app is able to provide.

The Java interface consists of 2 elements:

- Java class called Barcode
- JNI Library called libbarcode_jni.so

IMPORTANT: Leave the package name as “Softek” in the Barcode class or else it will not be able to load the JNI library.

The java class and JNI library do not control the video camera on the Android device – this must be done within your app and the SDK includes the source code to the Barcode app to show how this can be done.

The properties of the Java class are detailed in [Appendix B](#). Note that the Java class does not support barcode reading from file formats such as jpg or tif. It is designed to read barcodes using the ScanBarcodeFromBitmap function, with the bitmap presented in YUV420SP format – which also happens to be the format delivered by the camera preview callback.

The key file that demonstrates how to integrate the camera and the java class can be found in the download-archive at the following location:

Barcodeapp_X_X_X/src/com/softekssoftware/bardecoder/scanner.java

7 Supported Barcode Formats

7.1 1-D Barcode Formats

The following 1-D barcode formats are supported 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](#))

7.2 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](#))

8 Appendix B: Properties Reference

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

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

8.1 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](#)

8.2 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

8.3 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](#)

8.4 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](#)

8.5 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: $1 + 2 + 3 + 4 + 5 + 10 + 11 + 12 + 13 + 14 + 41 = 116$

$115 / 43 = 2 \text{ rem } 30$, so U is the check digit.

Data and check digit = 12345ABCDE+U

Type: BOOL

Default value: FALSE

See also: [ReadCode39](#)

[Code39NeedStartStop](#)

[ExtendedCode39](#)

8.6 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 probability of a false positive reading is increased significantly by setting this property to FALSE.

Type: BOOL

Default value: TRUE

8.7 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](#)

8.8 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](#)

8.9 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](#)

8.10 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](#)

8.11 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](#)

8.12 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](#)

8.13 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

8.14 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](#)

8.15 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](#)

8.16 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 $\text{GammaCorrection} / 100$. 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

8.17 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: ""

8.18 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](#)

8.19 MaxLength

Overview

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

Type: SHORT

Default value: 999

See also: [MinLength](#)

8.20 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

8.21 MinLength

Overview

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

Type: SHORT

Default value: 4

See also: [MaxLength](#)

8.22 MinOccurrence

Overview

Please refer to [PrefOccurrence](#) for more information.

Type: SHORT

Default value: 2

8.23 MinSeparation

Overview

MinSeparation defines the minimum distance between barcodes of identical value and vertical alignment in $1/300^{\text{th}}$ 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

8.24 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

8.25 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

8.26 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](#)

8.27 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

8.28 PatchCodeMinOccurrence

Overview

Please refer to [PrefOccurrence](#) for more information.

Type: SHORT
Default value: 30

8.29 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](#)

8.30 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

8.31 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](#)

8.32 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 \geq `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 \geq [MinOccurrence](#). Note that Patch Codes are only ever reported if the score is \geq [PatchCodeMinoccurrence](#).

Type: SHORT

Default value: 5

See also: [MinOccurrence](#)

[PatchCodeMinOccurrence](#)

8.33 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 $1/10^{\text{th}}$ of the value of the image resolution (e.g. 10 pixels in a 100 dpi image).

Type: SHORT

Default value: 0

8.34 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](#)

8.35 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](#)

8.36 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](#)

8.37 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

8.38 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](#)

8.39 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

8.40 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](#)

8.41 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

8.42 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

8.43 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

8.44 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

8.45 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

8.46 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

8.47 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

8.48 ReadQRCode

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

8.49 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

8.50 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

8.51 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

8.52 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

8.53 ShortCode128MinLength

Overview

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

Type: SHORT
Default value: 2

8.54 ShowCodabarStartStop

Overview

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

Type: BOOL
Default value: TRUE

8.55 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

8.56 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

8.57 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

8.58 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

8.59 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

8.60 UseRunCache

Overview

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

Type: BOOL
Default value: TRUE

8.61 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

9 Appendix D: Release Notes

9.1 Version 7.4.1

First version for Android.