

# Intersoft.Crosslight.Services.Barcode Assembly

Some remarks and description about this assembly.

## Classes

<a href="#">AbstractExpandedDecoder</a>	Used for internal mechanism.
<a href="#">AbstractRSSReader</a>	Used for internal mechanism.
<a href="#">AlignmentPattern</a>	Encapsulates an alignment pattern, which are the smaller square patterns found in all but the simplest QR Codes.
<a href="#">AztecCode</a>	Aztec 2D code representation
<a href="#">AztecDetectorResult</a>	The class contains all information about the Aztec code which was found
<a href="#">AztecReader</a>	This implementation can detect and decode Aztec codes in an image.
<a href="#">AztecResultMetadata</a>	Aztec result meta data.
<a href="#">AztecWriter</a>	Generates Aztec 2D barcodes.
<a href="#">BarcodeMetadata</a>	Metadata about a PDF417 Barcode
<a href="#">BarcodeReaderGenericBase&lt;T&gt;</a>	A smart class to decode the barcode inside a bitmap object
<a href="#">BarcodeScannerManager</a>	Represent barcode scanner manager.
<a href="#">BarcodeValue</a>	A Barcode Value for the PDF417 barcode. The scanner will iterate through the bitmatrix, and given the different methods or iterations will increment a given barcode value's confidence. When done, this will return the values of highest confidence.
<a href="#">BarcodeWriterGeneric&lt;TOutput&gt;</a>	A base class for specific barcode writers with specific formats of barcode images.
<a href="#">BaseLuminanceSource</a>	The base class for luminance sources which supports cropping and rotating based upon the luminance values.
<a href="#">BigIntegerException</a>	BigInteger-related exception class.
<a href="#">Binarizer</a>	This class hierarchy provides a set of methods to convert luminance data to 1 bit data. It allows the algorithm to vary polymorphically, for example allowing a very expensive thresholding technique for servers and a fast one for mobile. It also permits the implementation to vary, e.g. a JNI version for Android and a Java fallback version for other platforms.
<a href="#">BinaryBitmap</a>	This class is the core bitmap class used by barcode scanner to represent 1 bit data. Reader objects accept a BinaryBitmap and attempt to decode it.
<a href="#">BinaryShiftToken</a>	Represent binary shift token.
<a href="#">BitArray</a>	A simple, fast array of bits, represented compactly by an array of ints internally.
<a href="#">BitMatrix</a>	Represents a 2D matrix of bits. In function arguments below, and throughout the common module, x is the column position, and y is the row position. The ordering is always x, y. The origin is at the top-left. Internally the bits are represented in a 1-D array of 32-bit ints. However, each row begins with a new int. This is done intentionally so that we can copy out a row into a BitArray very efficiently. The ordering of bits is row-major. Within each int, the least significant bits are used first, meaning they represent lower x values. This is compatible with BitArray's implementation.
<a href="#">BitSource</a>	This provides an easy abstraction to read bits at a time from a sequence of bytes, where the number of bits read is not often a multiple of 8. This class is thread-safe but not reentrant. Unless the caller modifies the bytes array it passed in, in which case all bets are off.
<a href="#">BoundingBox</a>	A Bounding Box helper class
<a href="#">ByQuadrantReader</a>	This class attempts to decode a barcode from an image, not by scanning the whole image, but by scanning subsets of the image. This is important when there may be multiple barcodes in an image, and detecting a barcode may find parts of multiple barcode and fail to decode (e.g. QR Codes). Instead this scans the four quadrants of the image -- and also the center 'quadrant' to cover the case where a barcode is found in the center.
<a href="#">ByteMatrix</a>	JAVAPORT: The original code was a 2D array of ints, but since it only ever gets assigned 0, 1 and 2 I'm going to use less memory and go with bytes.
<a href="#">CameraResolution</a>	Represent camera resolution.
<a href="#">CharacterSetECI</a>	Encapsulates a Character Set ECI, according to "Extended Channel Interpretations" 5.3.1.1 of ISO 18004.
<a href="#">CodaBarReader</a>	Decodes Codabar barcodes.
<a href="#">CodaBarWriter</a>	This class renders CodaBar as System.Boolean[].
<a href="#">Code128Reader</a>	Decodes Code 128 barcodes.
<a href="#">Code128Writer</a>	This object renders a CODE128 code as a <a href="#">BitMatrix</a> .

Code39Reader	Decodes Code 39 barcodes. This does not support "Full ASCII Code 39" yet.
Code39Writer	This object renders a CODE39 code as a <a href="#">BitMatrix</a> .
Code93Reader	Decodes Code 93 barcodes.
Codeword	A Codeword in the PDF417 barcode
DataCharacter	Represent data character.
DatamatrixEncodingOptions	The class holds the available options for the DatamatrixWriter
DataMatrixReader	This implementation can detect and decode Data Matrix codes in an image.
DataMatrixWriter	This object renders a Data Matrix code as a BitMatrix 2D array of greyscale values.
Decoder	The main class which implements Aztec Code decoding -- as opposed to locating and extracting the Aztec Code from an image.
Decoder	The main class which implements Data Matrix Code decoding -- as opposed to locating and extracting the Data Matrix Code from an image.
Decoder	The main class which implements MaxiCode decoding -- as opposed to locating and extracting the MaxiCode from an image.
Decoder	The main class which implements QR Code decoding -- as opposed to locating and extracting the QR Code from an image.
DecoderResult	Encapsulates the result of decoding a matrix of bits. This typically applies to 2D barcode formats. For now it contains the raw bytes obtained, as well as a String interpretation of those bytes, if applicable.
DecodingOptions	Defines an container for encoder options
DefaultGridSampler	Represent default grid sampler.
DefaultPlacement	Symbol Character Placement Program. Adapted from Annex M.1 in ISO/IEC 16022:2000(E).
DetectionResult	Represent detection result.
DetectionResultColumn	Represents a Column in the Detection Result
DetectionResultRowIndicatorColumn	Represents a Column in the Detection Result
Detector	Encapsulates logic that can detect an Aztec Code in an image, even if the Aztec Code is rotated or skewed, or partially obscured.
Detector	Encapsulates logic that can detect a Data Matrix Code in an image, even if the Data Matrix Code is rotated or skewed, or partially obscured.
Detector	Encapsulates logic that can detect a QR Code in an image, even if the QR Code is rotated or skewed, or partially obscured.
Detector	Encapsulates logic that can detect a PDF417 Code in an image, even if the PDF417 Code is rotated or skewed, or partially obscured.
DetectorResult	Encapsulates the result of detecting a barcode in an image. This includes the raw matrix of black/white pixels corresponding to the barcode, and possibly points of interest in the image, like the location of finder patterns or corners of the barcode in the image.
Dimension	Simply encapsulates a width and height.
Dimensions	Represent dimemnsions.
EAN13Reader	Implements decoding of the EAN-13 format.
EAN13Writer	This object renders an EAN13 code as a <a href="#">BitMatrix</a> .
EAN8Reader	Implements decoding of the EAN-8 format.
EAN8Writer	This object renders an EAN8 code as a <a href="#">BitMatrix</a> .
ECI	Superclass of classes encapsulating types ECIs, according to "Extended Channel Interpretations" 5.3 of ISO 18004.
Encodation	Enumeration for encodation types
Encoder	Generates Aztec 2D barcodes.
Encoder	Represent QRCode Encoder.
EncodingOptions	Defines an container for encoder options
ErrorCorrection	Error Correction Code for ECC200.
ErrorCorrection	PDF417 error correction implementation. This example is quite useful in understanding the algorithm.

ErrorCorrectionLevel	See ISO 18004:2006, 6.5.1. This enum encapsulates the four error correction levels defined by the QR code standard.
FinderPattern	Represent RSS Finder Pattern.
FinderPattern	Encapsulates a finder pattern, which are the three square patterns found in the corners of QR Codes. It also encapsulates a count of similar finder patterns, as a convenience to the finder's bookkeeping.
FinderPatternFinder	This class attempts to find finder patterns in a QR Code. Finder patterns are the square markers at three corners of a QR Code. This class is thread-safe but not reentrant. Each thread must allocate its own object.
FinderPatternInfo	Encapsulates information about finder patterns in an image, including the location of the three finder patterns, and their estimated module size.
FormatException	Thrown when a barcode was successfully detected, but some aspect of the content did not conform to the barcode's format rules. This could have been due to a mis-detection.
GenericGF	This class contains utility methods for performing mathematical operations over the Galois Fields. Operations use a given primitive polynomial in calculations. Throughout this package, elements of the GF are represented as an int for convenience and speed (but at the cost of memory).
GenericMultipleBarcodeReader	Attempts to locate multiple barcodes in an image by repeatedly decoding portion of the image. After one barcode is found, the areas left, above, right and below the barcode's ResultPoint are scanned, recursively. A caller may want to also employ ByQuadrantReader when attempting to find multiple 2D barcodes, like QR Codes, in an image, where the presence of multiple barcodes might prevent detecting any one of them. That is, instead of passing a Reader a caller might pass new ByQuadrantReader(reader).
GlobalHistogramBinarizer	This Binarizer implementation uses the old global histogram approach. It is suitable for low-end mobile devices which don't have enough CPU or memory to use a local thresholding algorithm. However, because it picks a global black point, it cannot handle difficult shadows and gradients. Faster mobile devices and all desktop applications should probably use HybridBinarizer instead.
GridSampler	Represent Grid Sampler.
HighLevelEncoder	This produces nearly optimal encodings of text into the first-level of encoding used by Aztec code. It uses a dynamic algorithm. For each prefix of the string, it determines a set of encodings that could lead to this prefix. We repeatedly add a character and generate a new set of optimal encodings until we have read through the entire input.
HybridBinarizer	This class implements a local thresholding algorithm, which while slower than the GlobalHistogramBinarizer, is fairly efficient for what it does. It is designed for high frequency images of barcodes with black data on white backgrounds. For this application, it does a much better job than a global blackpoint with severe shadows and gradients. This class extends GlobalHistogramBinarizer, using the older histogram approach for 1D readers, and the newer local approach for 2D readers. 1D decoding using a per-row histogram is already inherently local, and only fails for horizontal gradients. We can revisit that problem later, but for now it was not a win to use local blocks for 1D. This Binarizer is the default for the unit tests and the recommended class for library users.
InvertedLuminanceSource	A wrapper implementation of LuminanceSource which inverts the luminances it returns -- black becomes white and vice versa, and each value becomes (255-value).
ITFReader	Implements decoding of the ITF format, or Interleaved Two of Five. This Reader will scan ITF barcodes of certain lengths only. At the moment it reads length 6, 8, 10, 12, 14, 16, 18, 20, 24, 44 and 48 as these have appeared "in the wild". Not all lengths are scanned, especially shorter ones, to avoid false positives. This in turn is due to a lack of required checksum function. The checksum is optional and is not applied by this Reader. The consumer of the decoded value will have to apply a checksum if required. <a href="http://en.wikipedia.org/wiki/Interleaved_2_of_5">http://en.wikipedia.org/wiki/Interleaved_2_of_5</a> is a great reference for Interleaved 2 of 5 information.
ITFWriter	This object renders a ITF code as a <a href="#">BitMatrix</a> .
LuminanceSource	The purpose of this class hierarchy is to abstract different bitmap implementations across platforms into a standard interface for requesting greyscale luminance values. The interface only provides immutable methods; therefore crop and rotation create copies. This is to ensure that one Reader does not modify the original luminance source and leave it in an unknown state for other Readers in the chain.
MaskUtil	Represent mask util.
MathUtils	Represent math utilities.
MatrixUtil	Represent matrix util.
MaxiCodeReader	This implementation can detect and decode a MaxiCode in an image.
MobileBarcodeScannerBase	Represent mobile barcode scanner base.
MobileBarcodeScanningOptions	Represent barcode scanning options.
Mode	See ISO 18004:2006, 6.4.1, Tables 2 and 3. This enum encapsulates the various modes in which data can be encoded to bits in the QR code standard.
MSIReader	Decodes MSI barcodes.
MSIWriter	This object renders a MSI code as a <a href="#">BitMatrix</a> .
MultiDetector	Encapsulates logic that can detect one or more QR Codes in an image, even if the QR Code is rotated or skewed, or partially obscured.

MultiFormatOneDReader	Represent multi format one dimension reader.
MultiFormatReader	MultiFormatReader is a convenience class and the main entry point into the library for most uses. By default it attempts to decode all barcode formats that the library supports. Optionally, you can provide a hints object to request different behavior, for example only decoding QR codes.
MultiFormatUPCEANReader	A reader that can read all available UPC/EAN formats. If a caller wants to try to read all such formats, it is most efficient to use this implementation rather than invoke individual readers.
MultiFormatWriter	This is a factory class which finds the appropriate Writer subclass for the BarcodeFormat requested and encodes the barcode with the supplied contents.
OneDimensionalCodeWriter	Encapsulates functionality and implementation that is common to one-dimensional barcodes.
OneDReader	Encapsulates functionality and implementation that is common to all families of one-dimensional barcodes.
PDF417CodewordDecoder	Represent pdf 417 codeword decoder.
PDF417DetectorResult	PDF 417 Detector Result class. Skipped private backing stores.
PDF417Reader	This implementation can detect and decode PDF417 codes in an image.
PDF417ResultMetadata	PDF 417 result meta data.
PDF417ScanningDecoder	Represent pdf 417 scanning decoder.
PDF417Writer	Represent pdf 417 writer.
PerspectiveTransform	Represent perspective transform.
PlanarYUVLuminanceSource	This object extends LuminanceSource around an array of YUV data returned from the camera driver, with the option to crop to a rectangle within the full data. This can be used to exclude superfluous pixels around the perimeter and speed up decoding. It works for any pixel format where the Y channel is planar and appears first, including YCbCr_420_SP and YCbCr_422_SP.
PlesseyWriter	This object renders a Plessey code as a <a href="#">BitMatrix</a> .
QRCode	Represent QRCode.
QRCodeDecoderMetaData	Meta-data container for QR Code decoding. Instances of this class may be used to convey information back to the decoding caller. Callers are expected to process this.
QRCodeMultiReader	This implementation can detect and decode multiple QR Codes in an image.
QRCodeReader	This implementation can detect and decode QR Codes in an image.
QRCodeWriter	Represent QRCode Writer.
ReaderException	Represent reader exception.
ReedSolomonDecoder	Implements Reed-Solomon decoding, as the name implies.
ReedSolomonEncoder	Implements Reed-Solomon encoding, as the name implies.
Result	Encapsulates the result of decoding a barcode within an image.
ResultPoint	Encapsulates a point of interest in an image containing a barcode. Typically, this would be the location of a finder pattern or the corner of the barcode, for example.
RSS14Reader	Decodes RSS-14, including truncated and stacked variants. See ISO/IEC 24724:2006.
RSSExpandedReader	Represent RSS expanded reader.
RSSUtils	Adapted from listings in ISO/IEC 24724 Appendix B and Appendix G.
SimpleToken	Represent simple token.
StringUtils	Represent string utils.
SupportClass	Contains conversion support elements such as classes, interfaces and static methods.
SymbolInfo	Symbol info table for DataMatrix.
Token	Represent token.
UPCARReader	Implements decoding of the UPC-A format.
UPCAWriter	This object renders a UPC-A code as a <a href="#">BitMatrix</a> .
UPCEANReader	Encapsulates functionality and implementation that is common to UPC and EAN families of one-dimensional barcodes.
UPCEANWriter	Represent UPCEAN writer.
UPCEReader	Implements decoding of the UPC-E format. This is a great reference for UPC-E information.
Version	The Version object encapsulates attributes about a particular size Data Matrix Code.

Version	See ISO 18004:2006 Annex D
WhiteRectangleDetector	Represent white rectangle detector.
WriterException	Represent writer exception.

## Interfaces

<a href="#">IBarcodeReaderService</a>	Defines member required for barcode reader service.
<a href="#">IBarcodeRenderer&lt;TOutput&gt;</a>	Interface for a class to convert a <a href="#">BitMatrix</a> to an output image format
<a href="#">IBarcodeWriterGeneric&lt;TOutput&gt;</a>	Interface for a smart class to encode some content into a barcode
<a href="#">IEncodingService</a>	Represent member required for encoding service.
<a href="#">IMobileBarcodeScanner</a>	Defines the member for mobile barcode scanner.
<a href="#">IMultipleBarcodeReader</a>	Implementation of this interface attempt to read several barcodes from one image.
<a href="#">IReader</a>	Implementations of this interface can decode an image of a barcode in some format into the <a href="#">String</a> it encodes. For example, <a href="#">QRCodeReader</a> can decode a QR code. The decoder may optionally receive hints from the caller which may help it decode more quickly or accurately. See <a href="#">MultiFormatReader</a> , which attempts to determine what barcode format is present within the image as well, and then decodes it accordingly.
<a href="#">IWriter</a>	The base class for all objects which encode/generate a barcode image.

## Delegates

<a href="#">ResultPointCallback</a>	Callback which is invoked when a possible result point (significant point in the barcode image such as a corner) is found.
-------------------------------------	--

## Enumerations

<a href="#">BarcodeFormat</a>	Represent list of barcode format.
<a href="#">Compaction</a>	PDF417 compaction mode
<a href="#">DecodeHintType</a>	Encapsulates a type of hint that a caller may pass to a barcode reader to help it more quickly or accurately decode it. It is up to implementations to decide what, if anything, to do with the information that is supplied.
<a href="#">EncodeHintType</a>	These are a set of hints that you may pass to Writers to specify their behavior.
<a href="#">PDF417ErrorCorrectionLevel</a>	defines the level of the error correction / count of error correction codewords
<a href="#">ResultMetadataType</a>	Represents some type of metadata about the result of the decoding that the decoder wishes to communicate back to the caller.
<a href="#">Sign</a>	The number's sign, where Positive also stands for the number zero.
<a href="#">SymbolShapeHint</a>	Enumeration for <a href="#">DataMatrix</a> symbol shape hint. It can be used to force square or rectangular symbols.