Choosing the Right Barcode for your Application

LEADTOOLS®
THE WORLD LEADER IN IMAGING SDKs

LEAD TECHNOLOGIES INCORPORATED

www.leadtools.com

Introduction

For many of us, a day hardly goes by without seeing a barcode. For software developers, this creates immense opportunities to create software that handles both the encoding and decoding of barcodes. They are used in retail, government, commercial, smart phones, web applications, and more. There's hardly a market or industry that doesn't use barcodes on a daily basis.

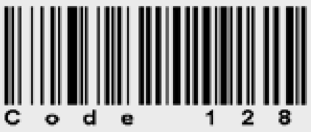
At the same time, all of this diversity and market saturation can present challenges as well. How does a developer or project manager get started on their path to success? When you boil it down, there are two primary questions you should ask: Which barcodes should I use? How will I use them? This white paper will help clear the muddied waters surrounding both of these questions by detailing several of the most popular barcodes and how programmers can use the award-winning imaging SDK technology in LEADTOOLS to quickly and easily develop their application.

Choosing the Right Barcode

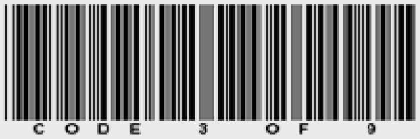
With so many types of barcodes available, choosing the right variety is essential to a successful software application. From the highest level view, there are two types of barcodes: One-Dimensional (a.k.a. Linear) and Two-Dimensional. 1D barcodes are the simplest and most widely used, but hold a small amount of information. 2D barcodes are gaining popularity due to the larger amount of information that can be packed into a smaller area. However, the decision is still more complicated than deciding between 1D and 2D. Here are some of the most popular barcodes and how they are typically used.

Code 128

This 1D barcode gets its name from the fact that it can encode the entire ASCII 128 character set. Surprisingly, it is still able to rival the compactness of other 1D barcodes that encode far fewer characters. This combination of density and diversity makes it one of the most widely used barcodes for industrial purposes such as shipping, packaging, transportation, and more.



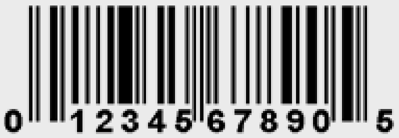
Code 39 (Code 3 of 9)



In a similar fashion, Code 39 derived its name from the number of characters it could encode, but has since been expanded to encode 43 rather than 39 characters (0-9, A-Z, plus a few special characters). Its density is lower than that of Code 128, and is most notable in its use as the standard barcode used by the United States Department of Defense, the Health Industry Bar Code Council, and other automotive purposes.

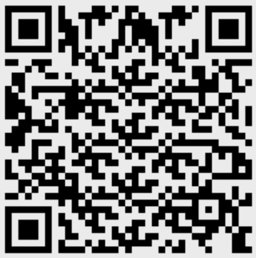
UPC/EAN

Arguably the most heavily used barcodes of all time, Universal Product Code (UPC) and European Article Number (EAN) are extensively used to identify consumer goods for point of sale retail applications. UPC and EAN primarily differ in that EAN uses two digits (00-99) and UPC uses one (0-9). To ensure the unique identification of retail goods, the values of UPC and EAN barcodes are set by international committees. This imposes a few limitations on the creativity afforded to software developers, but opportunities are still abundant due to the sheer market demand and widespread use.

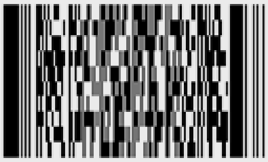


QR Code

Of the 2D barcodes, the most recognizable and consumer friendly is the QR Code. QR codes can pack an incredible amount of information into a small space: up to 2,953 bytes, 4,296 alphanumeric, or 7,089 numeric digits. In addition to the immense storage potential, it is also one of the most robust and error resistant codes ever invented. Therefore it is very popular in applications where poor image quality is almost a given, such as shaky mobile phones and hand-held scanners.

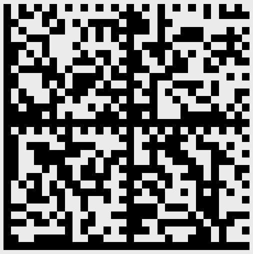


PDF417



PDF417 is the most common 2D barcode, and is heavily used on identification cards, shipping, and inventory management systems. Its popularity within these markets can be attributed to its large data capacity of 1,850 ASCII characters and resilient error correction. These attributes allow it to encode personal information and entire addresses while maintaining accuracy within harsh environments.

Data Matrix

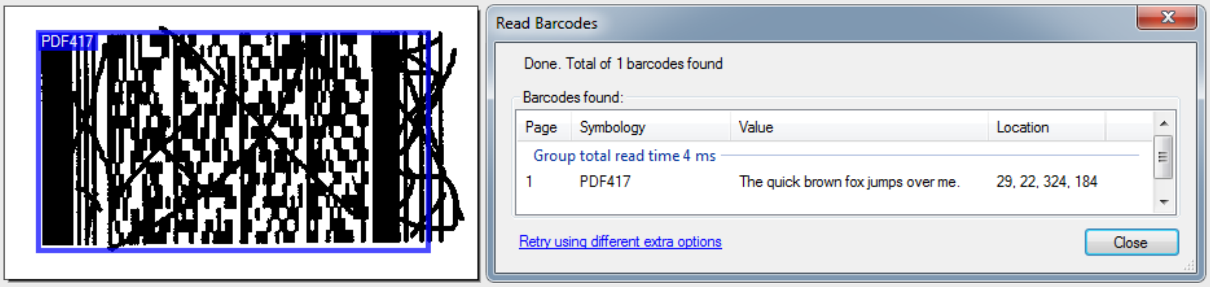


Data Matrix barcodes allow for the most compact storage of information and are proportionally sized based on the amount of text or binary data required (up to 2,218 ASCII characters). The smallest Data Matrix barcodes can encode fifty characters in just a few millimeters. It is less popular among consumers, but many corporations and enterprises take advantage of its compactness for their own internal logistics or to identify small items.

Using LEADTOOLS for Barcodes... and Everything Else!

Now that you've chosen the barcode(s) you want to use, how do you go about adding them to your application? Of course, you could write it yourself. This might be feasible for some varieties of 1D barcodes, but could add some major delays or staffing needs if you plan on supporting 2D barcodes or multiple 1D symbologies. Your other choice then, is to capitalize on the expertise of other programmers and use an SDK.

There is much to consider when looking for the best SDK for your needs. Most barcodes have straightforward specifications and standards, so as long as the SDK chosen for your application can read and write according to the rules you are in good shape. However, that assumes a perfect world where every image is clean, every photograph is taken in good lighting, and every document is scanned straight. That world simply doesn't exist so you should look for a toolkit that does much more than just barcodes and can provide you with everything required: scan, clean, preprocess, and deal with all the imperfections that are a part of document imaging.



Read Barcodes

Done. Total of 1 barcodes found

Barcodes found:

Page	Symbology	Value	Location
Group total read time 4 ms			
1	PDF417	The quick brown fox jumps over me.	29, 22, 324, 184

Retry using different extra options

Figure 1: A badly damaged PDF417 barcode that is still successfully read by LEADTOOLS

This is where LEADTOOLS excels! With over two decades of experience in creating digital imaging software development toolkits, we have seen it all. One of the biggest deciding factors for developers shopping for and testing different barcode toolkits is how well it works on their images. This can actually present a catch-22 because they are developing an application which has yet to be deployed into production and they don't have a large enough sample of images and situations.

LEADTOOLS is used in hundreds of thousands of desktop, server, and mobile applications across many different industries, so developers who use LEADTOOLS get an astounding peace of mind knowing that LEADTOOLS has been proven and market-tested against millions of images. Because of this experience, LEADTOOLS has adapted well to the quickly-changing marketplace by regularly adding new image processing routines that improve the reliability and accuracy of its barcode engines. For example, most barcodes found on images were captured with scanners. The majority of issues these scanners had were easily fixed by deskewing or resampling the image in some way to improve the definition of the lines. Mobile image capture has introduced a huge problem for barcode engines since the images regularly have bad lighting, three-dimensional skewing, and poor focusing. LEADTOOLS has image pre-processing routines that can run on the live camera feed to constantly and quickly fix the image without the user even knowing it, resulting in rapid barcode detection and decoding that many other libraries struggle with.

In addition to LEADTOOLS' ability to accurately read and write the most barcodes from the widest gamut of images, the library itself is very easy to work with. Reading and writing barcodes only takes a few lines of code.

```
// Create a Barcode engine
BarcodeEngine engine = new BarcodeEngine();

// Write a barcode to the image (UPC in this example)
BarcodeData data = new BarcodeData(BarcodeSymbology.UPCA, “01234567890”);
data.Bounds = new LogicalRectangle(0, 0, 400, 200, LogicalUnit.Pixel);
engine.Writer.WriteBarcode(myImage, data, null);

// Read all barcodes with default options.
BarcodeData[] barcodes = engine.Reader.ReadBarcodes(myImage,
   LogicalRectangle.Empty, 0, null);
```

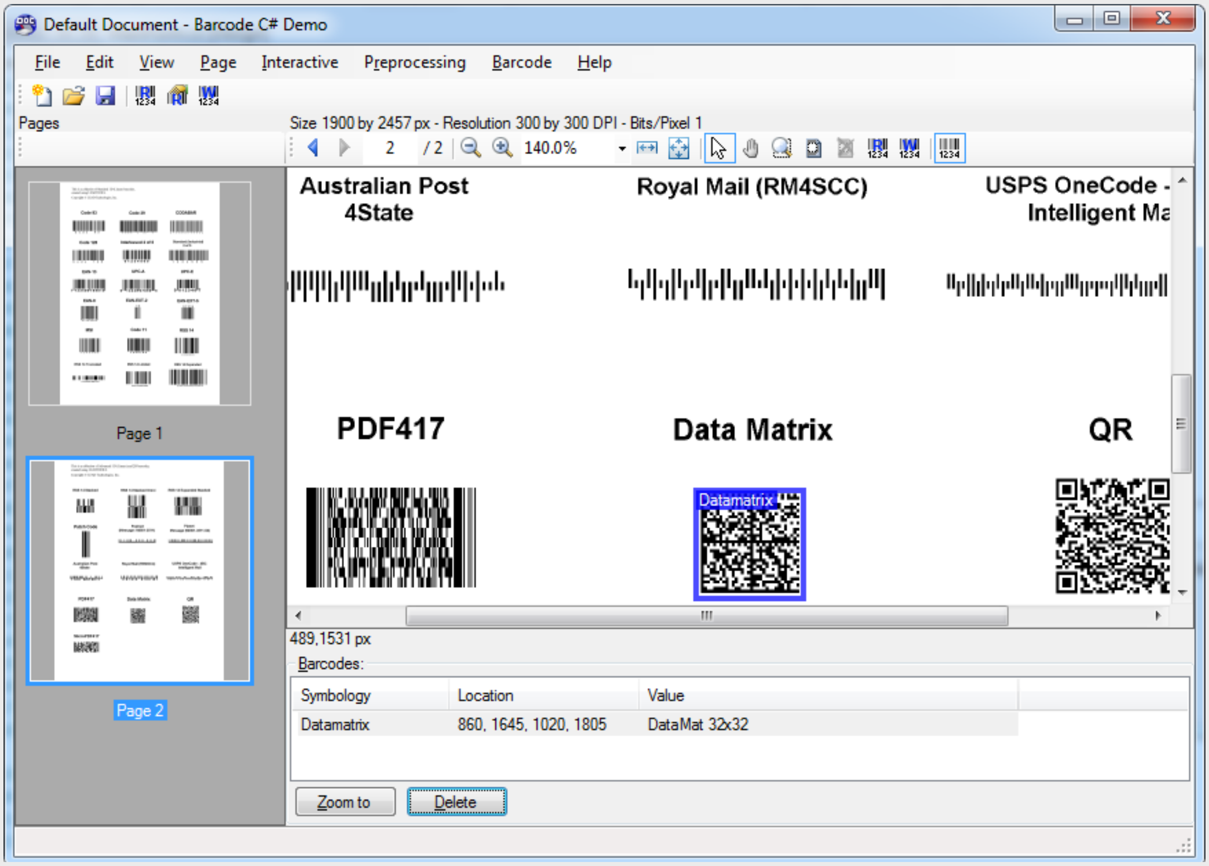
As shown above, LEADTOOLS provides high-level features to enable quick and easy reading of barcodes, but also exposes many options for fine-tuning the recognition process to your environment and images. When the type of barcode(s) is known, you can limit the search to only the symbologies your application will use, therefore increasing your performance to an even greater degree.

```
// Create a Barcode engine
BarcodeEngine engine = new BarcodeEngine();

// Read only PDF417 and Data Matrix barcodes
PDF417BarcodeReadOptions pdf417Options = new PDF417BarcodeReadOptions();
pdf417Options.BackColor = RasterColor.White;
pdf417Options.ForeColor = RasterColor.Black;
pdf417Options.SearchDirection = BarcodeSearchDirection.Vertical;

DatamatrixBarcodeReadOptions dataMatrixOptions =
   new DatamatrixBarcodeReadOptions();
dataMatrixOptions.BackColor = RasterColor.White;
dataMatrixOptions.ForeColor = RasterColor.Black;
dataMatrixOptions.EnableSmallSymbols = true;

BarcodeData[] barcodes = engine.Reader.ReadBarcodes(
   myImage, LogicalRectangle.Empty, 0,
   new BarcodeSymbology[] {
      BarcodeSymbology.PDF417, BarcodeSymbology.Datamatrix},
   new BarcodeReadOptions[] {
      pdf417Options, dataMatrixOptions });
```



Size 1900 by 2457 px - Resolution 300 by 300 DPI - Bits/Pixel 1

489,1531 px

Barcodes:

Symbology	Location	Value
Datamatrix	860, 1645, 1020, 1805	DataMat 32x32

Figure 2: Barcodes read from the narrow read options. Note that a PDF417 barcode exists, but is not returned because the engine was told to only search vertically.

Finally, since LEADTOOLS can provide a comprehensive solution for your barcode application, you don't have to deal with the inconvenience of mixing multiple SDKs together. This is a benefit that should not be overlooked, as trying to make sense of passing information and image data back and forth between multiple toolkits is more than just an inconvenience for the developer. It has real effects on the end-user as well because the application is not running at maximum efficiency. Since everything within LEADTOOLS works together seamlessly, no superfluous CPU cycles and RAM is needed and the overall end-product is a much greater quality.

LEADTOOLS can improve your entire software development life cycle: planning which barcodes you will support, writing the actual code with ease, handling real-world images, and providing a first-class application. Once you start using LEADTOOLS in your application, it won't take long to realize why it's The World Leader in Imaging SDKs.

6

Conclusion

This is just one of many real world solutions you can tackle with LEADTOOLS. Its state of the art Barcode SDK is the most flexible and powerful product in its class, and LEADTOOLS offers an incredible value with its comprehensive family of toolkits for raster, document, medical, and multimedia imaging. For more information on how LEAD Technologies can image-enable your application and boost your ROI, visit www.leadtools.com to download a free evaluation, or give us a call at +1-704-332-5532.

Sales: +1(704) 332-5532
sales@leadtools.com

Support: +1(704) 372-9681
support@leadtools.com

LEAD Technologies, Inc.
1927 South Tryon Street
Suite 200
Charlotte, NC 28203

About LEAD Technologies

With a rich history of over 26 years, LEAD has established itself as the world's leading provider of software development toolkits for document, medical, multimedia, raster, and vector imaging. LEAD's flagship product, LEADTOOLS, holds the top position in every major country throughout the world and boasts a healthy, diverse customer base and strong list of corporate partners including some of the largest and most influential organizations from around the globe.

LEADTOOLS®
THE WORLD LEADER IN IMAGING SDKs

LEAD TECHNOLOGIES
INCORPORATED