

# Bar Codes in Document Imaging

By Larry Krummel

# Bar Codes in Document Imaging

by

**Larry Krummel**

edited by Jeff Frankel

Fourth Edition – February 2009

Copyright © 1989-2009 ALLMYPAPERS, INC.

All rights reserved. No part of this publication may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopy, recording, or any information retrieval system, without permission in writing from the publisher.

# Forward

The general processing of bar codes is a substantial industry with standards, suppliers and well-developed applications. The how and why of using bar codes in document imaging is reasonably new. The key difference between the traditional bar code industry and the use of bar codes in documents is the read process. Traditional bar code reading is generally a manual process except under rigidly controlled conditions for automatic reading. Document imaging generally requires automatic reading from a largely uncontrolled input i.e. the image of a page. This paper will discuss:

- How bar codes are used in document imaging
- How bar codes can be used to improve data capture
- How to prepare documents for the imaging process (AllMyPapers would like to thank Steve MacWilliams of Document Control Solutions for his contributions to this section of the paper)
- What is the technology of bar codes
- What are the effects of various usage protocols on bar code reading

# Contents

Forward.....	iii
Contents.....	iv
Table of Figures .....	v
Introduction .....	1
Strategies for Cost Reduction .....	2
Overview .....	2
Job Control and Batch Separation.....	2
Process Error Detection .....	3
Scanned Image Quality Assurance .....	4
Automatic Indexing .....	5
Simplex/Duplex Control.....	6
Scan Resolution Control.....	6
Control of Contrast & Threshold Values .....	7
Indexing for Imaging.....	8
Introduction.....	8
Capture Vs. Re-capture.....	8
Indexing Methods .....	8
Specific Applications Using 2-D Bar Codes.....	11
Summary .....	15
Document Preparation for Imaging .....	17
Introduction.....	17
Planning for Life after Imaging.....	17
Inventory Control .....	18
Document Identification and Indexing.....	20
Post Scanning Steps .....	20
Summary of Prep and De-Prep .....	21
The Case for Pocket-Style File Folders .....	22
Bar Code Technology .....	23
Overview .....	23
Symbols and Symbologies .....	23
Anatomy of a Symbol .....	24
2-D Symbologies .....	30
Bar Code Creation .....	33
Overview .....	33
Accuracy and Reliability .....	34
Printing Bar Codes on the Desktop.....	43
FAX Input .....	44
Special Symbology Considerations .....	44
Automatic Bar Code Reading.....	46
Overview .....	46
Why Grocery Store Bar Codes Scan So Well.....	46
Optimizing Bar Codes For Automatic Reading .....	50
Choosing a Development Tool Kit.....	53
Bar Code Imaging Applications.....	54

Several Bar Codes of Different Lengths .....	54
Automated High-Speed Forms Acquisition .....	55
Multiple Symbols Stacked Vertically & Horizontally .....	56
Dynamic Changes in a Bar Code Symbol .....	57
Scanner Quality Assurance Test Page.....	58
Typical Separator Page .....	59
Appendix A - Bar Code Samples .....	60
Appendix B -About All My Papers .....	62
Company .....	62
Contact All My Papers .....	62
Appendix C - Further Reading .....	63
Reference Materials .....	63
Standards Organizations .....	63
Appendix D - Bar Code Printing & Tagging Suppliers .....	64
Avery LabelPro For Windows .....	64
B-Coder Pro for Windows .....	64
American Reprographic Systems, Incorporated .....	64
Appendix E - Imaging Suppliers and Integrators.....	65
Document Control Solutions.....	65
ImageTrax .....	65
ColorBar .....	65
Quick-Filer .....	65

## Table of Figures

Figure 1- Example back-file indexing separator sheet with 2-D bar code .....	13
Figure 2 - A Typical Bar Code Symbol in the Code 39 Symbology .....	24
Figure 3 - Bar Code Symbol Features.....	25
Figure 4 - A bar code with quiet zones on four sides .....	26
Figure 5- A Bar Code With Checksum Character .....	27
Figure 6 - Horizontal (left) and Vertical (right) Orientation.....	29
Figure 7 - Reading Skewed Bar Codes.....	30
Figure 8 - A typical PDF-417 symbol.....	31
Figure 9 - A Bar Code Exhibiting Dynamic Changes Within a Symbol.....	37
Figure 10 - Bar Code With Quiet Zone Violation .....	38
Figure 11 - Setting Bar Code Density in a Typical Printing Application .....	40
Figure 12 - A Bar Code Before and After Faxing .....	44
Figure 13 - Page Schematic with Bar Code .....	52
Figure 14 - DOD waybill .....	54
Figure 15 - A typical proof of delivery document .....	55
Figure 16 - Pharmacy reorder form (partial).....	56
Figure 17 - Typical airline ticket with two-part bar code .....	57
Figure 18 - scanner quality assurance test page .....	58
Figure 19 – Typical Separator page.....	59
Figure 20 - Bar Code Symbol Samples.....	60

## Introduction

High volume document imaging systems are routinely called upon to efficiently capture, store, manipulate, and retrieve hundreds of millions of document images. While the initial investment in such a system is high, ongoing labor costs can quickly make these outlays seem minuscule. Although these systems employ high- speed scanning hardware and capable operators, net throughput is low. Often, errors are frequent and the cost of quality assurance is high. In such a system, capital equipment is not being used at its rated capacity and human labor is being used to perform tasks that a computer could do better and faster.

The reason for this is simple: Capture and indexing of document images is usually performed in an interactive fashion in which the scanner operator manually keys-in document index information, performs document counts, watches for paper handling problems, and explicitly changes scanner control parameters when nonstandard documents are encountered.

Bar codes may be added to the document imaging environment to implement various strategies to reduce errors, increase system throughput and save money by more fully utilizing capital equipment and by reducing the cost of labor required for a given document imaging application.

# Strategies for Cost Reduction

## **Overview**

The 1-D bar code can be used in the document imaging environment to implement:

- Job control and batch separation
- Process error detection
- Automatic indexing of documents
- On-the-fly scanner control
- Scanned image quality assurance

The implementation of these strategies will reduce labor costs by eliminating operator interaction during the document capture for all but the most flawed documents.

2-D bar codes represent a new paradigm for data capture which can be used to implement:

- Machine readable turn-around documents
- Document-level indexing in back file conversions
- Forms ID
- Form layout definition
- Encoding of form data content

## ***Job Control and Batch Separation***

In a typical interactive system, documents are received at the document capture station, in their folders. The operator must usually remove documents from the folder, load them into the scanner's input hopper, input the folder identification information into the system, and only then, can scanning begin. At this point, the operator may also be required to explicitly inform the system when the first document is ready for scanning and when the end of the folder's contents have been reached. For each folder, this process is repeated.

In a batch mode system, the operator never takes an action in regards to the disposition of an individual document unless a problem is encountered. In batch mode processing, documents are grouped into batches and are interleaved with special control documents called job separator pages. Interleaving into the document stream is done during document preparation before documents are brought to the scan station. Job separator pages include bar codes whose contents have a specified meaning to the scanner control software. The specific meanings will vary by application but, usually a certain number of bar codes will be reserved to mark the beginning of, and to optionally call out the number of a new document batch. Other codes could be used to denote a new folder, drawer, client, end of scanning, or any other arbitrary event occurring within the application.

Depending upon the needs of the application, job separation strategies can be as simple or complex as necessary. A simple implementation might automatically imbed the current batch and page numbers into the document image file name and store all files in a single directory. A more sophisticated scheme could employ multiple classes of job separator pages to control the placement of document image files into specific directory locations and control the naming of these files according to specified criteria such as folder, client, etc.

## ***Process Error Detection***

In an interactive system, documents which have been inserted into the batch upside-down, facing backwards, or which are not scanned because 2 or more pages are stuck together in the scanner, often go undetected unless the scanner operator is constantly watching both the capture station's display and the operation of the scanner's feed mechanism. When this kind of error is detected, the operator must re-feed the offending document(s), or worse yet, collate the entire group and rescan it.

When this kind of error goes undetected, documents are lost or become unreadable in the electronic record. Unless their absence from the system is detected at scan time or in subsequent quality control stages, the unavailability of these lost documents may not make itself felt until years after the fact. By then it may be too late to retrieve the originals. Those doing litigation support know that a case in law can be won or lost based on the presence or absence of a single piece of documentary evidence.

However, in a batch mode system, if every document can be encoded with a uniquely numbered bar code, then upside-down, and even backwards facing pages can be detected and corrected without operator intervention. Affixing uniquely numbered bar code labels to each document can insure that no document is ever lost due to a scanner mis-feed. At the end of each batch of scanning, the capture software can detect missing bar code numbers and alert the operator to the fact of a mis-feed, pin-pointing exactly which documents need to be scanned again.

## ***Scanned Image Quality Assurance***

Document image scanners are complex electromechanical devices which are prone to various kinds of wear and damage which manifest as subtle changes in the quality of scanned document images. Some of these problems include:

Scanner lamp decay - Scanner lamps don't burn out all at once. Rather, they grow dimmer and change in color temperature, gradually reducing image quality over time.

- **CCD Element Failure** - CCDs are the image sensing elements in a scanner. When one fails, small dropouts occur in document images. While often not visible to human eyesight, these dropouts can greatly reduce OCR and bar code read accuracy.
- **Defocused Optics** - The repeated vibration of scanning can cause scanner optics to go out of alignment over time causing a loss of effective resolution. Dirt, grease, paper dust, toner, carbon, and ink will all build up on the surface of optical elements causing fuzzy images.
- **Scanning Speed Variations** - Mechanical wear on the transport and electronics which have drifted out of adjustment can cause variations in scanner resolution. These changes can be local (in one portion of the page image) or global (across the entire page image) depending on the cause.

This is not a well recognized problem. Most 10,000 page per month office copiers are sold with maintenance contracts and rigorous preventive maintenance schedules. However, production scanners in mission critical applications, which often read 10,000 page images

per day, are often sold with no maintenance contracts and are provided without a recommended maintenance schedule.

**A sample scanner quality assurance test page appears on page 58**

The scanning of precisely printed test pages and the analysis of the resulting files by diagnostic software should be used to objectively monitor document image quality on a regular basis. The data obtained from such a program should be saved and used to develop a long-term historical profile of scanner performance. This profile should in turn, be used as the basis for a rigorous preventative maintenance program.

## ***Automatic Indexing***

Every document that is entered into a document management system must be uniquely and accurately identified in the system's database through the use of a document index number. In an interactive system, it is the job of the operator to input these numbers. This is not only time-consuming, but as discussed earlier, prone to costly errors.

In a batch mode system, the application of a uniquely numbered and properly oriented bar code to each document or the separator sheet preceding it, not only confers the quality assurance benefits discussed earlier but can also can automate the capture of document index numbers. The benefits are these:

- Document index numbers become part of the scanned record for human visual or machine retrieval
- Document index numbers automatically (and without the errors induced by human index keying) become part of the document image's database record
- The scanner operator is relieved of the obligation to manually enter document index numbers during the scanning process -- thus saving a tremendous amount of time and greatly reducing errors.
- 2-D bar codes printed on separator sheets can be used to implement automatic indexing of complex back files such that each document in each file folder is identified and indexed according to a user defined hierarchy. The details of this will be discussed in a later section Scanner Control

In addition to automating the capture of document source information, separator pages can also be used to signal that scanner parameters

need to be changed in response to a change in the character of the document stream. An example bar coded separator page appears on page Typical Separator Page 59.

## ***Simplex/Duplex Control***

Some scanners are capable of duplex scanning - that is, scanning both sides of a document. While this capability can increase throughput, its use can be problematic in an interactive system: If a document stream contains both single and double-sided pages, an interactive system requires that one of the following three courses of action be followed:

- Sort documents into single and double-sided groups and scan these groups separately
- Scan all documents in duplex mode creating a number of blank page images
- Constantly interrupt the scanning process to switch scanner modes.

An example bar-coded separator page appears on page 59

With a batch mode system, separator pages are defined to switch the scan application from one mode to the other and back again. This moves the simplex/duplex scan decision from the scan station operator to the document preparation phase where it is an easy matter to slip control pages into the document stream. This allows documents to be scanned in their file order, on a non-stop basis and without generating blank page images.

## ***Scan Resolution Control***

Because scanning resolution has a large impact on document image file size, on-the-fly control of scanner resolution is very helpful for document imaging systems where only certain documents need to be scanned at high resolution. In a manner similar to that described in the simplex/duplex example, a batch mode document capture system could employ separator pages to control scanner resolution on the fly. In such a system, document images which will be submitted to OCR or other resolution critical operations can be scanned at 300dpi, while all other documents can be scanned at 200dpi.

## ***Control of Contrast & Threshold Values***

Again, in a manner similar to that described in the simplex/duplex example, a batch mode document capture system could employ bar-coded separator pages to control scanner threshold and contrast values to accommodate documents printed on colored paper, carbons, and NCR paper forms.

# Indexing for Imaging

## *Introduction*

Although document image capture systems have become quite sophisticated, the ways in which index information is captured has not always kept up with our ability to capture, store, forward, and display masses of document images. As a result, the indexing portions for many large imaging projects rely heavily on human keying of index data or human mediation of machine captured index data. This section will discuss a number of ways to automate indexing process using 1-D and 2-D bar codes, with the goal of reducing the labor content of your capture application.

## *Capture Vs. Re-capture*

Very often, the indexing information entered into an imaging system is information that already resides in a computer. When this data is captured properly either through direct data processing means or by machine reading of the document, a great deal of human labor at the indexing step of capture can be eliminated. Where indexing data already exists, we will refer to its capture as re-capture.

## *Indexing Methods*

Because of the proliferation of computer systems, most large bodies of information which might be converted to imaging such as medical and business records have some correspondence to a computer database. When such a correspondence exists, this database can be used as the source from which indexing information is drawn. Other large bodies of information may consist of or include numbered or bar coded forms whose data may be machine read during the imaging process. This data can be used as the index or to aid in the indexing process.

While each imaging application is unique, indexing for imaging is generally accomplished using one of four models which we list in ascending order of automation. These models can be applied depending on the availability and quality of indexing information for

the documents to be imaged. We will call these four models; Manual Keying, Hybrid Keying, Label and Separator Automation, and Full Automation.

### **Manual Keying**

In a keying based indexing system, documents are scanned, then displayed on- screen. A human operator then inspects the document's content, and either keys-in some portion of that content, or makes a decision about indexing based on that content, keying-in the index according to external rules.

The proliferation of computer systems would seem to make the manual keying approach a rarity these days. However, this is not the case. It is heavily used and may be the most common method of indexing. Especially when some portion of the data has been hand-written.

### **Hybrid Keying**

Hybrid Keying adds some element of machine readability to the equation. Using OCR, ICR, bar code reading, or other technology these system either re-capture some of the data required by the imaging application, or re-capture provisional indexing information for checking by a human operator.

### **Partial Re-capture**

In a partial re-capture scenario, document index information such as bar-coded form numbers are captured by machine reading the forms during imaging. Once this data is captured, the image of the document is displayed to a human key operator who enters the remaining information required by the application. An example of such an application is the processing of overnight courier airbills.

When an airbill is received, its image is captured and its bar code read by the imaging system. If a record for that airbill exists in the system (which it might if it were a pre-printed airbill issued to a regular shipper), then its image is displayed to a key operator who enters consignee information. If the airbill has no record (one that is not pre-issued to a particular customer), then the key operator must enter shipper, billing, and consignee information.

In this scenario, the bar-coded form number was re-captured to establish the airbill's index in, or to retrieve the airbill's index from the

database, and in the latter case, retrieve the shipper's account information.

### **Provisional Indexing**

Provisional indexing applies OCR or similar technology to re-capture information from the document image to be used as the document's index. The data so retrieved is then provided a provisional index along with the document's image to a human operator who checks the accuracy of the provisional index and either confirms or corrects it. This operator or a separate key operator might also be involved in entering other document information.

An example of provisional indexing might involve the entry of medial claim forms which contain machine printed claim numbers which might or might not be accurately read by the imaging system due to variation in the condition of the forms received.

### **Label and Separator Automation**

Label and separator automation refers to the use of self-adhesive bar coded labels, or bar-coded separator pages which are applied to documents and interleaved into the document stream before scanning. These labels and separators identify the beginning and end of records in the document stream, and may identify each document by record number or by type within the record. The bar codes on these labels and separator pages are read by the imaging software and the extracted data is used as part of the index for the documents. A complete description of such an application is described in the section titled Document Preparation for Imaging on page 17.

### **Full Automation**

Full automation refers to the cases where the re-capture of an existing index is fully automatic. This can be accomplished using 1-D or 2-D bar codes. In the case of 1D bar codes, the 1-D bar code data is used to "look-up" entries in an external database. In the case of 2-D bar codes, the full index can reside in the bar code itself.

### **Bar Codes as an Alternative to OCR**

We have discussed the use of bar codes and OCR for the generation of index information. When given a choice between bar codes and OCR, bar codes are the preferred technology because most bar

codes implement error detection, and some have error correction. When an OCR engine makes an error, there is no way to know that an error has been made much less how to correct the error. As a result, when OCR is used to develop document indexes, human checking is required to verify the results as described in Provisional Indexing above. When bar codes are used for index generation, checksums in 1-D bar codes and error-correction in 2-D bar codes allow the process to be automated. Human intervention is only required when a particular bar code cannot be read, but in no case (thanks to checksums, etc.) is a substitution error possible as is the case with OCR. With bar codes, human intervention is the exception, not the rule.

The 2-D symbologies allow bar codes to become something more than mere serial part numbers. Because of their ability to encode large amounts of data, 2-D bar codes can become portable data files - hence the "PDF" in the symbology name PDF-417. A PDF-417 symbol can encode between 1,800 and 2,800 characters of information in a few square inches. With this kind of density, data about a shipment can travel with a shipment in a form that is instantly machine readable. No longer is it necessary to read a bar code, then use the bar code's message to reference a remote database to access the data associated with the coded object or form.

## ***Specific Applications Using 2-D Bar Codes***

### **Document Indexing for Back-File Conversions**

In a back-file conversion scenario, there is usually an existing database which is essentially an extraction of the data contained in the back-file (paper files). For this discussion, let's use the example of a mortgage lender's loan inventory. For each loan there exists a database record which includes summary data about the borrower, the property, and the inventory of paperwork associated with the loan. However, when reference to the actual paperwork is needed someone must physically access the paper files.

So the object of the back-file conversion is to capture images of all documents from the back-file and index them so that each one is properly linked to a database record. The desired result is the ability to quickly find, view, print, or transmit the documents without ever referring back to a physical piece of paper.

Scanning the back-file is a straightforward process. Indexing the back-file is more difficult. This kind of indexing can be done in one of two ways:

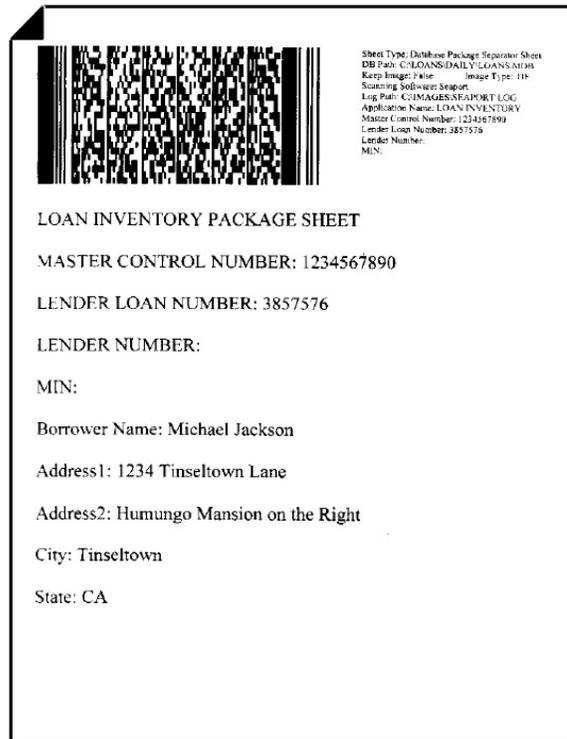
**Manually**, where a scan station operator views each scanned document on-screen and keys-in an appropriate classification data for the document image. At a minimum, each document must be indexed by loan number. It would be much more valuable however, to further index each document within each loan folder by type so that for example, when the certificate of title for a particular loan was required, it could be called up explicitly. Such classifications might include; legal documents, title insurance related documents, closing documents, credit related documents, and so on.

However, manually segregating document images in this manner can be inefficient and error-prone because:

- It requires operator knowledge of how to classify the documents. Whenever a gap in that operator knowledge occurs during the indexing process, the operator must create an exception for that document which in turn must be resolved by another human operator with the requisite knowledge.
- It requires human key entry which is prone to error.

**Automatically** - A much better way to index document images is to perform the indexing function automatically using coded separator sheets employing 2-D bar codes. The process would work something like this:

1. A program is run that queries the existing database for the complete inventory of documents in the back-file.
2. For each document, a separator sheet is generated which displays a 2-D bar code whose message contains all of the indexing data required to completely identify that document. An example of such a separator sheet is shown in the figure below.



**Figure 1- Example back-file indexing separator sheet with 2-D bar code**

In addition to the machine readable 2-D bar code, each separator sheet includes human readable data which includes the loan number, the borrower's name, the property address, and the name or classification of the document associated with the separator sheet. Separator sheets are printed in back-file order, that is in the order which matches the physical organization of the back-file.

3. Now the separator sheets are physically placed into the document stream so that each separator sheet precedes the document with which it is associated.
4. The document stream is scanned. The document image capture system is configured to properly name, route, tag, and otherwise index the resulting document images so that they are properly linked to the master database.

While much of what has been described could be done using 1-D bar codes, because 1-D bar codes can only encode relatively short messages, they could not carry all of the indexing information required to completely identify each document in a machine readable form. Therefore, the indexing process would require constant access to the master database in a client-server arrangement. This would in

turn, make it difficult to carry out the back-file conversion using outsourcing.

In contrast, the use of separator sheets employing 2-D bar codes which encode complete indexing information for each document means that the entire back file conversion can be accomplished using outsourcing, and without explicit access to the master data. In addition, the resulting fully indexed and imaged back-file could be published on CD-ROMs.

### **Turn-Around Documents - Encoding of form data content**

Credit card companies buy huge mailing lists of names and credit histories which are used to solicit new credit card customers through direct mail offerings. These mailings can run to ten million pieces or more, but response rate are often less than 1%. Keeping 10 million names and credit histories on-line while waiting for a few thousand consumers to respond to the offer unnecessarily monopolizes huge amounts of computer resources. It also poses the problem of when to purge the database. If names are purged after 90 days but completed applications continue to trickle in for six months, the credit card company must balance the cost of re-keying against the cost of keeping the entire database on line indefinitely.

If all customer information is securely encoded within a 2-D bar code on the face of the application, then the customer name and credit history would travel with the credit card application and could automatically be re-entered into the credit card company's computer system by machine scanning the completed applications as they are returned. Thus, in such a system, computing resources are not unnecessarily tied up and no re-keying of previously available information is required - except in the case of a severely damaged application.

### **Forms ID and Form Layout**

When a machine readable form is to be read by a computer system, the first task is to identify the form number and revision so that data locations and field definitions established for the form can be retrieved. While OCR may be employed for this purpose, a 2-D bar code can often be used to do the job more reliably. Because 2-D bar codes encode a relatively large message in a small area, they can be used to encode additional information such as the form's print run and vendor for quality control purposes.

Machine readable forms are used in all kinds of survey, application (employment, credit card, etc.), and testing applications. These forms employ combinations mark- sense fields, hand printing fields for text input and other kinds of user-supplied data destined for machine reading. In order to accurately retrieve the data returned on these forms, layout, content and meaning of each field must be carefully specified to the software that will read them. When the layout of such a form changes so must the programming of the read software. Further, when multiple forms are in use, returned forms must be sorted as to type and revision so that form data is correctly read into the proper database.

The form ID is used as an index to an external form description. The use of a 2-D bar code to encode the form layout allows the complete form description to travel with the form. This makes the form layout information (where to look for data, the meaning of each field, the destination database, and field mapping within that destination database) self-evident to the imaging system without access to an external database of form information.

There are two major advantages to such a system:

- Re-programming of the forms read engine is eliminated. No matter how many form revisions are in the field, the read engine can automatically adjust to any compatible form.
- Mixed populations or forms can re-read asynchronously and without operator intervention because the read engine can be self-reconfigured on a document-by- document basis.

### **Parcel Shipping**

In parcel shipping, large volume shippers might use software provided by the carrier to produce air bills which include human and machine readable information for all airbill fields, thereby eliminating all manual keying of shipment data and completely automating the sorting of the package while retaining human readability of the airbill.

## **Summary**

Using bar codes to automate data capture and re-capture can dramatically improve the reliability of existing applications which currently employ OCR, and more importantly, open up new

applications for automatic data capture. Employing 2-D bar codes can significantly reduce the labor cost of document image capture. At the same time, system throughput can increase and error rates can be greatly reduced.

The 2-D bar code changes the indexing process from provisional human-assisted indexing to fully-automated indexing - data processing, not people processing.

# Document Preparation for Imaging

## *Introduction*

In planning an imaging strategy, often the focus is on the cost per image scanned with an emphasis on the imaging system hardware, software, and the labor to scan pages. Too often overlooked in this planning process is document preparation for scanning, filing of the documents after scanning, and how the process of physical document retrieval, preparation, scanning, and filing can be leveraged to yield more compact, better organized and more complete physical and imaged files at the end of the process. These organizational processes which are performed prior to and after scanning we call document prep and de-prep. Throughout this discussion, we will use the example of a hypothetical mortgage lender to illustrate some of the issues of prep and de-prep.

Our lender will use an imaging system to achieve timely access to its documents. Its physical files are arranged by loan, and for each loan, the lender keeps several documents including a loan application, a credit report, title to the property, etc. A useful imaging application for this organization would therefore consist of a database of loans which is used to access all of the document images associated with each loan

The major issues which must be confronted for successful prep/de-prep include:

- Planning for life after imaging
- Inventory control
- Document identification and indexing
- Restoring the physical file after scanning

## *Planning for Life after Imaging*

Prior to imaging, each of our lender's loan records consist of a number of documents -some of which contain signatures or are documents which must be retained in physical form for legal reasons.

Other documents are purely informational or are duplicates of other documents and might only need to be kept as images - or not kept at all. In planning for imaging you should understand the nature of all the documents in the application so that the appropriate choices can be made respect to the disposal of physical documents. If it is found that some the documents can be destroyed after imaging, the reduced cost of storage after imaging will offset to some degree, the cost of conversion.

Prior to imaging, our lender's records are kept in relatively heavy folders with all documents in a loan file bound together for protection. For a wholly physical file whose contents are repeatedly accessed, physical protection and organization of this nature is appropriate. After imaging however, our lender may need to access the physical file only rarely. From analyzing our lender's business model, physical files, and the way in which imaging will be used by the lender, we see that:

- a) physical access to the documents will be greatly reduced after conversion;
- b) the traditional bound folders currently in use by our lender consume significant amounts of linear file space;
- c) binding clips will have to be removed prior to conversion, and that;
- d) the replacement of binding after conversion would be labor intensive.

Given these observations, it might be appropriate to consider an entirely different organizational approach for these files, that of using a light-weight, pocket-style folder to store the contents of each loan file as it is prepared for conversion. After conversion these same folders will provide safe storage for the documents using significantly less linear file space than traditional bound files, and without the rebinding labor that the older method would require.

## ***Inventory Control***

Inventory control in the context of a imaging conversion project such as is envisioned by our lender encompasses three areas:

- a) missing document identification;
- b) management of individual files and documents during conversion, and;

- c) ongoing inventory control at the document and file levels for the file room.

### **Missing Document Identification**

As long as every file is going to be physically handled prior to imaging, it is a good time for our lender to perform a document inventory on each loan file before imaging takes place. Since bar codes will be used to identify each document during conversion, these same bar codes can be used in conjunction with a records management software package to take an initial inventory at the document level, using bar coded labels and hand scanners, and following these steps:

- Create a hierarchy of bar coded labels using a unique account identifier or loan number as the prefix, and the document type as the suffix of the bar code message.
- Affix a bar code label bearing the loan or account number to each loan's pocket folder.
- Affix a bar code label bearing the loan number prefix and the document type suffix to the first page of each document in each loan file.
- Wand scan the bar codes of each folder and each document face page.

Since each loan record should have the same documents (a loan application, a title insurance policy, etc.) missing documents can now be identified by their absence in the records management software database.

Once this inventory has been taken, even records which have missing documents can be sent on through the conversion workflow to be scanned. As the missing documents are obtained, they can be bar coded and scanned. As long as the proper bar code labels are affixed to face of the recovered documents prior to scanning, they will be reunited with their counterparts in the image record. After scanning, the recovered documents can easily be returned to the proper folder by matching the human readable portion of the document's bar code label with the proper folder.

### **Management of Individual Files and Documents During Conversion**

During the process of imaging, our lender's business must continue as usual. When a loan file needs to be physically accessed, it must be readily available to staff members - regardless of where it is in the conversion workflow. Therefore, the location of every loan file and document must be tracked during the conversion. Using records management software, a check-out and check-in system can be established to allow safe access the physical files either by folder or by document, at any stage in the conversion process. The same tracking system could also provide progress of conversion metrics to facilitate management of the project.

### **Ongoing Inventory Control**

The same bar code-based records management system which facilitated missing document identification and inventory tracking during conversion could be used by our lender as a file room inventory system to keep track of folders and documents on an ongoing basis. By thinking of the folders and documents as inventory, and by carefully tracking their whereabouts using such a system, our lender can be sure that the physical records underlying their image data stays intact over the years.

## ***Document Identification and Indexing***

In order for our lender to achieve the timely access it needs, each document must reside in the imaging system and be properly linked to the appropriate loan record. In some manner therefore, the imaging system must be able to automatically identify each document at scan time. Given the thrust of this paper, we suggest this be accomplished by reading a bar code. Effective ways to bar code documents for automatic identification at scan time are by affixing self-adhesive bar code labels to the first page of each document as described, or by using bar coded separator sheets. Some applications, call for a combination of both approaches using a bar code separator sheet to identify the start of a new file folder and document bar code label to identify various document types.

## ***Post Scanning Steps***

Once the files have been scanned, they need to be placed back into their file folders and returned to the file room. Assuming that our lender has elected to use pocket- style folders, no labor beyond collating, insertion into the pockets and transport to, and filing in, the file room are necessary. Some scanners are robust enough to allow the entire pocket to be scanned as a separator sheet thereby keeping everything in sequence in the scanner output tray and allowing documents and pocket to be easily reassembled.

## ***Summary of Prep and De-Prep***

Given the scenario described, the steps for document prep and de-prep for our lender would be to:

- Plan for life after imaging by determining which documents can be stored or discarded after the conversion process; this determines how you may wish to store them now.
- Perform inventory control on, and edit the physical files as needed prior to image capture.
- Identify all documents by loan and document type so that each may be scanned as a discrete batch and removed from or inserted back into the workflow as a unit.
- Use bar code labels that will interface with records management software and a document imaging system. This will allow the use of portable bar code readers for records management and a scanner and document imaging application for automatic indexing of the document images.
- Physically prepare the documents so that they may be fed to a scanner by removing paper clips, staples, and binding.
- Label each document by loan and by document type early on in the document's life, using bar codes so that the imaging system can make this identification automatically at scan time. Don't wait until the conversion date to think through the process of labeling.
- After scanning, documents are returned to their folders and folders are returned to the file room.

## ***The Case for Pocket-Style File Folders***

Pocket-style file folders directly address several important issues with respect to the physical storage of files after scanning. These are the labor required to re-bind documents together, and the linear file space that these fasteners and traditional file folders consume. Pocket-style folders sharply reduce the need for both of these commodities.

The nature and frequency of access to the physical files changes after imaging. Before imaging, nearly every reference to a loan would have required physical file access. Here, book-style binding is appropriate.

After imaging, physical access is greatly reduced as the imaging system will suffice for day-to-day access. This reduces the role of the physical file to a legal archive where most accesses will be for the purpose of photocopying. In this scenario, bound documents are a handicap.

Pocket-style folders can be made from strong, light-weight materials which consume much less space than traditional folders and bindings, reducing the space consumed in the linear files by non-data items. This kind of thinking is applied as a competitive advantage in the overnight shipping industry where Federal Express and some of its peers use Tyvec™ envelopes to save both weight and space in the cargo holds of their aircraft.

Additionally, pocket-style folder systems can incorporate symbol banding on the pocket edges. Such symbol banding systems using patterns of black and white blocks allow folders to be organized in the file room by loan number (in the case of our lender), or any other criteria the user selects. With such a banding system, physical file organization is easily enforced as misfiled pockets are easily spotted by their out-of-sequence band patterns.

For those users who elect to continue with traditional filing methods, color-bar style banding production systems are available to match their existing file room banding formats, or to add color-coded banding to their unbanded physical file. These systems produce color-coded, self-adhesive strip labels which can be produced on demand and affixed to the file folder before, during, or after the conversion process.

Please refer to Appendix E of this document for a list of suppliers of products to facilitate document preparation, control, and filing.

# Bar Code Technology

## Overview

See “Why Grocery Store Bar Codes Scan So Well” on page (ref)

Bar codes are a technology for automatic identification which has found application in many business sectors. While bar codes have many applications, they have been most visibly embraced by segments of the retail grocery and consumer goods industries.

Bar codes are designed to be read by computers. A bar code is a self-contained message whose information is encoded in the geometry of its printed bars and spaces. When a bar code is read, the patterns of light and dark contained in the bars and spaces are translated into patterns of ones and zeros which the reading computer interprets as numeric or alphanumeric data. In a simple bar code system, bar code reading is accomplished by passing a wand containing a light source and a photo-receptor across the code to be read. As the device passes over the pattern of bars and spaces, the photoreceptor receives a pattern of reflected light corresponding to the bars and spaces of the bar code. These light to dark and dark to light transitions are translated into an on-off signal that is interpreted by the computer.

More sophisticated bar code reading technologies such as those that might be implemented in a batch mode document image capture system, can discern the presence of one or more bar codes within the image of a scanned document and can make the data contained within the bar code (the bar code’s message) available to the document image management system in a variety of ways.

## Symbols and Symbologies

To the uninitiated, most bar codes tend to look pretty much alike. However, there are a number of different bar code alphabets (called symbologies) in use today. Symbologies can roughly be divided in two major groups called linear (1-D) and two dimensional (2-D) symbologies. 1-D symbologies look like the traditional bar codes that you see on product packaging and are used to encode short messages like part, stock, serial, or customer numbers. 2-D bar codes are beginning to appear in places like shipping documents and medical insurance forms where they are used to encode larger

masses of data like the contents of an entire shipment or a portion of a patient's medical record. Because many characteristic of 2-D symbologies are derived from 1-D bar code technology, 1-D bar codes will be discussed here first.

## 1-D Symbologies

Examples of several bar code symbologies may be found in Appendix A.

A 1-D symbology can vary in the number of characters in its alphabet. For example, CODE 39, the symbology of the sample shown in the figure below, supports 43 different characters including letters, digits and some punctuation.

Another 1-D symbology called CODABAR includes only the ten digits plus the symbols \$-:/,+ . Some symbologies can represent the entire 128 character ASCII set.



*Figure 2 - A Typical Bar Code Symbol in the Code 39 Symbology*

While each symbology was developed with a slightly different purpose or market in mind, all symbologies share the common goal of being readable by computers. The various symbologies also differ from one another in other ways which will be discussed later.

One complete bar code is called a symbol. The figure above is a typical bar coded symbol. The digits running along the bottom of the symbol are provided so that the symbol can be read by humans as well as computers.

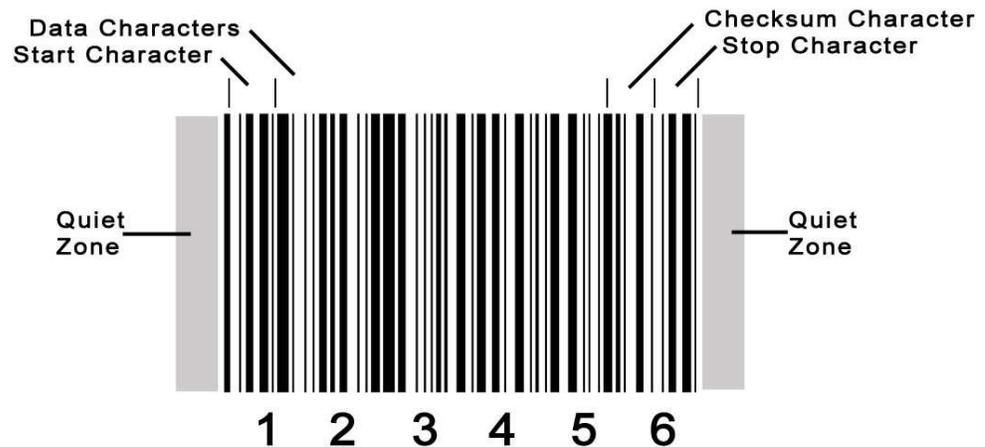
## ***Anatomy of a Symbol***

Because the details of 1-D bar code symbols are more readily apparent to the eye, the illustrations in this section show 1-D bar

Appendix C provides a reading list and the names of organizations that can provide more information about the various bar code symbologies.

codes. However, the concepts presented are generally applicable to both 1-D and 2-D symbologies. Further, while it is not the purpose of this document to fully describe the encoding details of each symbology, general information will be presented here to aid in the choice of an appropriate symbology along with guidelines for printing and reproduction of individual symbols.

A bar coded symbol consists of a number of printed bars, and intervening spaces. The characters within a symbol are encoded in the pattern of various sizes of narrow and wide, bars and spaces. The widths of the bars and spaces, as well as the number of each varies depending on symbology. The next figure shows a typical Code 39 bar code symbol along with some of the features common to many symbologies.



*Figure 3 - Bar Code Symbol Features*

### Quiet Zones

Quiet zones immediately precede the start character and follow the stop character. During the scanning process, quiet zones provide long periods of time (by a computer's reckoning) in which no light to dark or dark to light transitions occur. Quiet zones physically separate the bar code from other information on the page making it easier for scanners or software to differentiate bar code data from other information in the document image. For good bar code reading performance, quiet zones must be free of all extraneous marks. The width of the quiet zone required varies by symbology. For Code 39, the quiet zone must be a minimum of 10 times the width of the narrowest bar or space, or .25", whichever is greater.

In the document imaging environment, quiet zones should extend on all four sides of the symbol. If human readable characters are part of the symbol, they may appear close to the bar code as long as a quiet zone is present outside of this text. The figure below illustrates quiet zones appropriate for use in document imaging.



*Figure 4 - A bar code with quiet zones on four sides*

### **Start & Stop Characters**

Start and stop characters are pre-defined patterns of bars and spaces specific to each symbology, which mark the beginning and end of a symbol. In the CODE 39 symbol shown above, the start and stop characters are both the same. In some symbologies, the start and stop characters are different from one another.

### **Data Characters**

Data Characters are the characters that comprise the actual numbers or letters contained in the symbol. The CODE 39 symbology can accommodate a variable number of characters in a symbol. A few symbologies strongly recommend a fixed number of characters.

### **The Checksum Character**

A checksum character is a character included within a symbol whose value is calculated mathematically based on other characters in the symbol. Checksums are used to perform a mathematical check on the validity of a bar code's data. However, checksums are not always

used in all symbologies. Because a misinterpreted symbol will result in a lost document, when designing bar codes for a document imaging application, it is vitally important that the checksum feature be properly implemented.

When the symbol is read, the checksum character is generally assumed to be the last character in the symbol. A checksum calculation is made on all but the last character, and the result is compared to the value of that character. The result of the calculation is expected to equal the value of the checksum character. If a match is not made, no bar code read occurs.

The bar code symbol in the figure below contains the data “123456”. While only 6 characters of actual data are contained in this sample, note that 7 characters of information are presented. The last character is the checksum character. In this case, the checksum character is the letter “L”. While it may seem odd that a letter bears a mathematical relationship to the numbers “123456”, remember that all information is processed as numbers within a computer.



*Figure 5- A Bar Code With Checksum Character*

Be aware that while checksums can be implemented in any symbology, they are not required in all symbologies. Any bar code symbol which was not created expressly for your document imaging application, or whose original application is not known, should be treated with suspicion. It may or may not contain a valid checksum. Without a valid checksum character reliability of data is impaired.

## **Continuous & Discrete Symbologies**

Symbologies are said to be either continuous or discrete. Discrete symbologies code each character starting with a bar, ending with a bar, and use a space to separate characters. Continuous symbologies code each character starting with a bar, ending with a space, and provide no inter-character gap. While this characteristic does not have a direct bearing on the suitability of a particular symbology to the document imaging environment, it does have an influence a symbology's maximum character density and hence the minimum size for a particular bar coded message or symbol.

## **Bar and Space Widths**

Each symbology encodes data using a basic unit of measure called the x-width. The width of each bar or space within a symbol will always be equal to a multiple of the x-width. Bar and space widths for a particular symbology will range from one x-width to the maximum number of x-widths per element supported in that symbology. The simplest symbology (Postnet) permits only one unit of width for every bar or space, thus all elements occupy the same amount of horizontal space. More complex symbologies allow for up to four modules per element. These more complex coding schemes provide better encoding density but require much better control of both printing and scanning. As a rule of thumb, when printing or scanning bar code symbols, the printer or scanner used must be capable of printing/resolving 2 pixels for each x-width of a 1-D symbol and 3 pixels of a 2-D symbol.

## **Number of Characters In a Symbol**

Most symbologies allow nearly any number of characters to be encoded in a symbol. The three exceptions to this are UPC-A, UPC-E and EAN-13 which require 12, 6, and 13 digits, respectively. In a document imaging application, always use the same number of characters in every symbol regardless of whether the symbology allows such variations. When creating symbols add leading zeros as necessary to maintain the selected number of data characters per symbol.

## **Bar Code Character Sets**

As stated earlier, some symbology character sets support only the 10 numeric digits, others support digits and punctuation marks, some digits and uppercase letters, and some support the complete ASCII

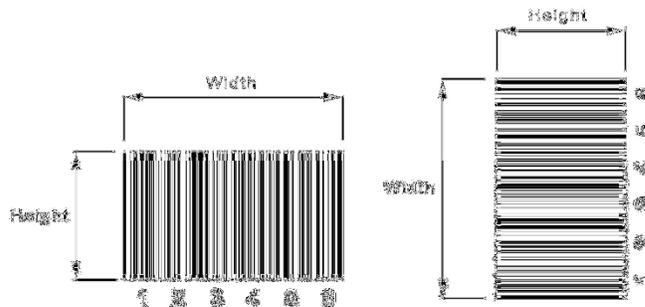
character set. If an indexing scheme requires alphanumeric content in the image id field, then a symbology supporting an appropriate character set must be employed. If document image identification numbers will be strictly numeric, then a wider choice of symbologies will be available.

### Overhead Characters

Overhead Characters are characters which are not part of the bar code's message, but are required parts of each bar code symbol. Start, stop, and checksum characters are examples of overhead characters.

### Height and Width of a Bar Code Symbol

In a document imaging environment, bar code symbols can't always be affixed to a document so that they read in the normal left to right orientation. In addition documents are not always loaded into the scanner in their normal reading orientation. Regardless of orientation, the height of a bar code symbol is always measured by the length of its bars, and its width is always measured perpendicular to its bars. The orientations shown on the left and right in the figure below are referred to as horizontal and vertical respectively.



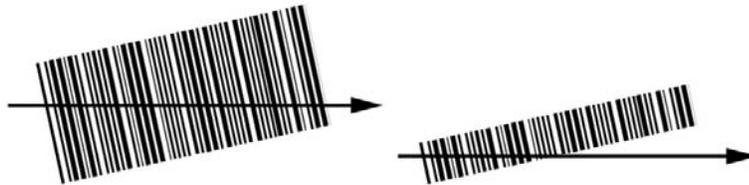
*Figure 6 - Horizontal (left) and Vertical (right) Orientation*

### Bar Code Skew

Unwanted rotation of the bar code image is called skewing. Skewing can be introduced by imperfect paper handling in a scanner and/or when a bar code label is affixed to a document. Badly skewed bar codes take longer to recognize and read, and so can reduce system throughput.

The height of a bar code determines how much skewing can be tolerated without impacting read performance. Simple bar code reading can be performed by scanning a path across the horizontal plane of the image (the vertical plane for vertically oriented bar codes). Reading a bar code with significant skew requires the system to search at all possible angles. The more angles searched, the slower the performance. Most bar code reading systems will provide control of search strategies which in turn will control performance. If only a small amount of skew is expected, it is not necessary to search for large amounts thereby improving performance.

In the figure below, two bar codes, identical except for their height, have been skewed to 12°. The arrows represent the default scanning path. Notice that the scanning path crosses every bar in the code on the left, but misses some of the bars in the bar code on the right. The bar code on the right will require more processing time than will the one on the left.



*Figure 7 - Reading Skewed Bar Codes*

Some bar code reading technologies are limited in the amount of skew they can tolerate (often 10° or less) regardless of bar code height. Such a technology would be unable to successfully read either bar code in the previous figure.

## **2-D Symbolologies**

As discussed earlier, 1-D symbolologies provide data integrity by implementing unique start and stop codes and checksum characters. 1-D symbols may be printed in different heights to provide for redundant scan paths and to allow for diagonal scanning. Symbols damaged in a small area may retain their integrity because of this redundancy.

The data in a 2-D Symbol is contained in small square or hexagonal elements which greatly reduce redundant or diagonal scan paths. This fundamental difference allows 2-D symbols to achieve enormous gains in data density over their 1-D cousins. However, the reduction

in of data redundancy dictates that 2-D symbologies implement error detection and correction systems in addition to the start, stop and checksum characters found in 1-D symbols.

Several, 2-D symbologies have come into use over the last several years. A 2-D symbol often appears as a checkerboard of black and white elements and are usually square in shape. A 2-D bar code symbol will include a unique finder pattern which will allow it to be distinguished from surrounding image data. This characteristic of 2-D symbols is especially important in document imaging applications since finding the symbol is as important as reading its data.

Some 2-D symbologies are called stacked symbologies. Their symbols are literally stacks of 1-D bar codes that are printed in a rectangular shape to achieve the most efficient use of document space. In addition to the data message, additional overhead characters are included in each stacked bar code to provide information describing which bar code in the “stack” any particular piece of data belongs to. This allows skewed 2-D symbols to be read without loss of data.

PDF-417 is a typical 2-D symbology which allows the compact printing of large data messages. The figure below illustrates a typical PDF-417 symbol.



*Figure 8 - A typical PDF-417 symbol*

While the symbol in the previous figure is no larger than a typical 1-D bar code symbol, it encodes the message; “The PDF-417 symbology can encode up to 1,850 ASCII characters or up to 2,725 digits in a single symbol”. In addition to the message itself, this symbol contains a significant amount of error correction code which will allow the message to be read even if part of the symbol is damaged. Currently, PDF-417 is the most popular 2-D symbology.

## Characteristics of 2-D Symbologies

The following table lists some of the attributes of several, 2-D symbologies.

Symbology	PDF-417	Maxi Code	Data Matrix	Code One
Application	Portable data files	parcel sorting	semiconductor part, lot, serial #s	Small item ID & sorting
Level of Usage & Support	High	Low	Low	Low
AIM Standard	Yes	In Process	In Process	Yes
Error Detecting	Yes	Yes	Yes	Yes
Error Correcting	Yes	Yes	Yes	Yes
Error correction levels	9	Fixed	12	Fixed
Max alphanumeric characters/symbols	1850	93	5000	2218
Min element size	.0075"	.035"	.006"	.0075"

## Reading 2-D Bar Codes

All 1-D bar codes and some 2-D symbologies (including PDF-417 and Code 16K) may be read with simple light-pen style devices. Other 2-D require a scanning type symbol reader. All 1-D and 2-D symbologies, given appropriate software, can be read in the document imaging environment.

In 1-D symbologies, start and/or stop characters are used to aid the scanning mechanism (either hardware or software) in finding the exact location and orientation of bar code symbols. Most 2-D symbologies implement a similar detection/orientation schemes which take the form of “finder patterns” which are characteristic borders or center sections of the 2-D symbol.

## Non-Bar Code 2-D Encoding

There are a variety of ways, generically known as glyphs, to encode machine readable data on paper in ways which don't involve the use of bar codes. Such a technology is the DataGlyph™ developed by Xerox. This technology encodes data using thousands of tiny, diagonal lines - each slanted at a 45-degree angle. Each line represents a binary bit set to 1 or 0 depending on whether it slopes to the right or left.

Glyphs do not have finder bars to identify their location. Further, glyphs can be distributed across the face of a document wherever there is white space. For these reasons, glyphs will require long

search times and are therefore, are generally unsuitable for use in the production imaging environment.

In the document imaging environment, glyphs raise two additional concerns:

- In order to achieve high data density, glyph elements are often small enough to approach the noise level of the scanner. When glyph element size falls below a certain size, the glyphs become unreadable.
- Document imaging applications often explicitly remove image halftone information in order to reduce page image file sizes. This function must be turned off if glyph data may be present since removal of halftone patterns tends to render the glyph data unreadable.

To date, no glyph technology has been released to the public domain and adopted as a standard by AIIM, ISO, or ANSI. Therefore, their use has been sporadic and is generally subject to licensing fees.

## Bar Code Creation

### *Overview*

The creation of new bar codes expressly for use in a document imaging application provides the opportunity for the user to exert a great deal of control over their quality and consistency. This section outlines the requirements for the production of high quality bar codes so that the best throughput and lowest error rate can be obtained in your document imaging application.

A good bar code read technology can find bar codes in documents at arbitrary locations, at arbitrary angles, and to a great degree, tolerate degraded bar code images. However, the very best accuracy and performance will be obtained when bar codes are presented in a consistent fashion on the target documents. Using forms with preprinted bar codes is desirable because their bar codes are always printed in a known orientation relative to the other information on the form. This makes the detection and correction of upside-down and backwards facing forms very reliable.

Appendix D provides a list of several bar code printing packages

If a document imaging application does not lend itself to preprinted bar codes, then the most common way to add a bar code to existing documents is self-adhesive labels. There are a number of third party software packages which can be used to print bar coded labels. These printing packages generally permit full pages of bar code labels to be printed with each label containing a unique sequential number.

## Accuracy and Reliability

In order for document capture to operate with few exceptions and at the rated capacity of the scanner, the bar codes used for job separation and document indexing must be of good optical quality to assure accuracy, reliability, and good performance in reading. It is these issues that need to be understood in order to increase productivity in a batch mode system. When a document image is processed to read bar codes, there are 4 possible results:

Result	Meaning
Result 1	Bar code data is returned accurately and the document is properly accounted for. This is the desired result which can be expected when bar code creation recommendations are followed.
Result 2	An error is returned because some or all of the symbol could not be read, an exception is generated. While the operator must get involved, the document is accounted for. Occasionally, this result is permissible. Result 2 generally occurs when the system knows a-priori that a bar code should be present. For example, when a bar code exists on every page. As long as the frequency of this type of error is low, it is generally acceptable and easily recovered.
Result 3	<p>False data is returned without generating any kind of error notification and resulting in the loss of access to a document. This is called a substitution error which should never be encountered. The proper use of checksums will all but eliminate this result.</p> <p>Result 3 is caused by misreading the bar code data. The image contents are broken or distorted and the resulting bar code width data is interpreted as too wide or too narrow. To compensate, many read tries of the data are performed. Individual characters may be misread from one internal read pass to another.</p> <p>The checksum characters virtually guarantees that the characters read and the characters printed on the page are the same. Any bar code generation system which does not use checksums in the data has to have verification logic external to the bar code to assure data quality.</p>
Result 4	No data generated because, either the bar code is degraded to the point that its data cannot be read from the document image (a read error), or no bar code exists on the document to begin with. Careful generation and selection of bar codes can for the most part, eliminate read errors.

	Result 4 is the most difficult problem. In general, it can only be addressed by careful consideration of the issues surrounding read reliability
--	--

### **Accuracy**

A bar code image on paper may have breaks or fills in the image data. Where such breaks or fills occur, one or more bars and/or spaces may be read as inappropriately too wide or too narrow. The character-level parity checking built-in to most symbologies would be sufficient to catch a single such error. However, because two such errors can cancel each other out, it is possible even with parity checking that a bad symbol can re-read as good. This results in a substitution error (result 3). Thus, the data integrity of the entire symbol must be assured. The checksum mechanism provides the assurance that any combination of individual errors within the symbol will cause it to be rejected.

### **Reliability**

Assuming that checksums are properly used and substitution errors have been eliminated (i.e. result 3 has been dealt with), we are left with issues of insuring reliability (preventing result 4) which concern themselves with the presence and readability of the bar code. Reliability means that the presence of a bar code is accurately ascertained and that its data is properly read in spite of any defects.

The first question that must be answered here is: Is there a bar code on this document? Assuming that your application requires a bar code on each document, then any document for which a bar code can't be found must be treated as an exception. If the goal of high throughput is to be met, then the bar code reading technology employed must be very robust in terms of making this determination. A technology that permits the implementation of a variety of search strategies and which can provide information about the nature and location of any bar codes found is absolutely necessary if one is to accurately answer this question for each and every document encountered.

### **Factors Affecting Accuracy and Reliability**

Once the bar code is found, it must be read reliably. When a bar code can't be read, then a read error and hence, an exception is generated. Again, if the goal of high throughput is to be met, unreadable bar codes must be kept to a minimum. Unreadable bar codes can be traced to a number of causes:

- Poor print quality of the original symbol
- The use of an inappropriately small bar code symbol
- Size reduction of the symbol during an intermediate document copy process
- Loss of symbol data during the faxing of a document
- Distortion of bar code metrics (blooming) by the scanner during scanning
- Scanning at too low of a resolution
- Violation of the quiet zone
- Dynamic changes in bar code element sizes or element size relationships
- Use of a poorly maintained scanner

### **Dynamic Changes within a Symbol**

Dynamic Changes in Element Size - A basic capability of a bar code reader should be its ability to adapt to the variations in element sizes and size relationships. This feature allows the system to reliably read symbols regardless of variations in bar/space density caused by ink spread during printing and distortions of the paper due to environmental factors.

Any reader technology that uses fixed input values for minimal element size or for narrow/wide relationship is inherently unreliable. Many document image bar code reading technologies ignore this important characteristic of bar code symbols; either because their underlying technology does not permit this level of discrimination, or because the vendor is not aware of this issue. When this characteristic is ignored, subtle reliability problems can occur, even in applications which have at times appeared to run flawlessly. In addition to ink spread and paper distortion factors, some bar codes (such as those found on airline tickets) are printed in two sections. The first is printed on the ticket form itself, the second is printed when the ticket is issued by the airline or travel agency. This second portion is added by a computer- driven dot-matrix printer whose print quality, alignment and ink density can vary significantly.

The figure below illustrates a bar code which exhibits dramatic dynamic changes in both character density and skew angle within a

single symbol. The sample bar code used for this figure was taken from an actual airline ticket.



**Figure 9 - A Bar Code Exhibiting Dynamic Changes Within a Symbol**

While a bar code reading technology may appear to be reliable in a particular application, if it is not tolerant of element size variations, sooner or later, a subtle change in the nature of the bar coded symbols encountered could cause the application's read error rate (and hence, the exception rate) to skyrocket. Factors which could trigger this kind of event include:

- A change in ambient relative humidity causing documents to expand or contract - thereby distorting the bar codes
- In the case of commercially printed forms or bar code labels, a change in printing vendors, or even a change in ink density from one press run to the next
- In the case of desktop bar code label printing, a change in hardware, software, ink formulation, or ribbon

The important thing to remember here is that any bar code reading technology which expects fixed bar code element widths or fixed element width relationships is inherently unreliable. No single test of a bar code reading technology will reveal the effects of these environmentally caused distortions of bar codes. Only a bar code reading technology which can dynamically determine bar code element widths can insure freedom from this very subtle but disastrous form of application failure.

### **Quiet Zone Violations**

A violation of the quiet zone is another potential source of read errors. For example, the initial implementation of a form might follow all the rules for bar code generation. Later, a new forms designer with little or no understanding of bar code technology will add something to the form. If this new element is added within the boundaries of a bar code's quiet zone, bar code reading reliability will be greatly reduced. Similarly, if forms are initialed or stamped in the conveniently blank

area of the quiet zone, reliability will again be impaired. The figure below shows a bar code with a typical quiet zone violation.



*Figure 10 - Bar Code With Quiet Zone Violation*

However, if the bar code reading technology employed by your application can read bar codes in multiple directions, then a single impacted quiet zone will not affect the reliability of bar code reads. This feature can frequently eliminate read errors caused by quiet zone violations.

### **Scanner Maintenance**

An often overlooked source of read errors is a poorly maintained scanner. Scanner lamps degrade with use over time, exhibiting reductions in both output intensity and color purity. This degradation reduces the contrast of scanned images and increases the bar code read error rate.

### **Scanner Settings**

The image stored in a file is the image from which the bar code read software must extract data. If the scanner settings used during the scan of a particular document are set at an inappropriate level for that page image (for instance, too light or too dark), then the bar code read software may not be able to properly read the bar code data. Therefore, when setting scanner brightness and contrast controls, always view a few sample documents after they have been scanned and test bar code read software on various settings to determine which settings yield reliable results.

### **Symbology**

Whether printing self-adhesive labels or designing bar codes into a form, a symbology must be specified. Here are some things to consider in choosing a symbology:

- Choose a symbology whose character set supports all of the characters required in the document image index number. For example, if letters and numbers are required, then CODABAR will be inappropriate for your application
- Choose a symbology which allows a printing density (in characters per inch) which allows the production of symbols small enough to fit within the constraints of the documents to be scanned

### Symbology Characteristics

The table below presents a summary of the symbology characteristics which may be of interest to the document imaging user.

Symbology	Character Set	Checksum Support	Overhead Characters	Confidence <sup>1</sup>	Coding Scheme	Minimum X-Width
Code 93	Alphanumeric	Required	4	Excellent	C	4
Code 39	Alphanumeric	Optional	2	1/45 Trillion	D	2
Code 128	Full ASCII	Required	1	Excellent	C	4
CODABAR	Digits & Punctuation	Optional	2	1/50 Million	D	2
Int. 2 of 5	Digits	Optional	1	Unknown	C	2
Airline 2 of 5	Digits	Optional	1	Unknown	D	2
UPC A	Digits	Required	2 <sup>2</sup>	Excellent	C	4
UPC E	Digits	Required	2	Excellent	C	4
EAN	Digits	Required	2	Excellent	C	4
Postnet	Digits	Required	2	Unknown	D	1

### Horizontal Size

When a bar code is affixed to a document, there is a tendency to make it as small as possible. Often this is necessary because the document is already mostly filled with human readable data which must be preserved. However, inappropriately small bar codes, pose distinct problems when they are read by software.

Successful bar code reading relies on the reading computer's ability to distinguish the difference between bars and spaces, both of which (depending on symbology) come in at least two, and sometimes as many as four different widths. Accurately preserving these differences in widths through printing, reproduction, and scanning of

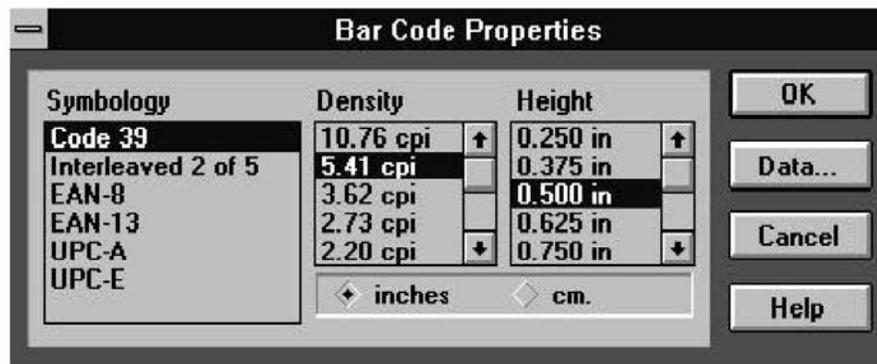
---

<sup>1</sup> Refers to the inherent reliability of a symbology's checksum mechanism. 1/45 Trillion means that properly printed bar codes will exhibit 1 substitution error in 45 trillion reads. Excellent means better than 1/50 Trillion, Unknown means no published data is available for this symbology.

<sup>2</sup> Includes the number system character which is used to indicate the retail domain of an item. Depending on the application, this character may be used for data.

the bar code symbol is essential to their reliable use. So much so, that a major portion of each symbology specification is devoted to printing standards.

Most symbologies may be produced at several densities (Point sizes). The figure below, illustrates how bar code character printing densities are generally set. Notice that in the example below, bar codes cannot be printed at arbitrary densities. Instead, several “legal” densities offered for each symbology.



**Figure 11 - Setting Bar Code Density in a Typical Printing Application**

Densities are measured in characters per inch of bar code width. For every allowable density, the symbology’s specification calls for a specific size and tolerance of the narrowest element. For example the specification for the Code 39 symbology specifies that, at 9.4 characters per inch (the maximum allowable density), the narrow bars and spaces must be .0075 inches wide, give or take .0017 inches.

Since many imaging systems are based on document scanning at 200 dpi, there is a limit to how dense bar codes can be and still be effectively scanned. Beyond a certain density, a 200 dpi scanner can’t capture the fine details of a bar code well enough so that the bar code reading software can accurately read it. The table below outlines these limits:

Symbology	Maximum Density <sup>3</sup> @ 200 Dpi Scanning	Maximum Density @ 300 Dpi Scanning
Code 93	8.4	14.8
Code 39	5.4	9.4
Code 128 (Digits only)	13.8	24.2
Code 128 (Alphanumeric)	6.9	12.1
CODABAR	7.3	12.8
Interleaved 2 of 5	10.3	18
Airline 2 of 5	10.3	18
UPC/EAN <sup>4</sup>	10.0	10.0
Postnet	5	9

The numbers in this table were arrived at based on the assumption that a scanner must "see" at least two pixels of the data in the narrowest bar code element encountered for the subject symbology and character density in order to reliably read the symbol. The formula used to determine the number of pixels of data "seen" by the scanner is the smallest element size in inches multiplied by the density of scanning in dpi.

Let's use the example of a Code 39 symbol printed at 9.4 characters per inch and scanned at both 200 and 300 dpi. For this symbology at this density, the smallest element (a narrow bar or space) is .0075 inches wide.

In the case of a 200 dpi scan, we multiply .0075" by 200, the result is 1.5 pixels -not enough data to reliably read under all circumstances. In such a case, one would have to use bar codes printed at a lower character density or scan the target document at a higher resolution in order to achieve reliable bar code reads

In the case of a 300 dpi scan, we multiply .0075" by 300, the result is 2.25 pixels which is above the minimum criteria for readability and we can assume that a bar code symbol in this instance could be reliably read.

### Vertical Size

The minimum vertical height of a symbol must be .25" or 15% of the symbol's width, whichever is greater. This is a standard requirement for codes 39, 93, Codabar, interleaved 2 of 5 and code 128. The UPC and EAN codes have a nominal height of 1.0" with a minimum

---

<sup>3</sup> Includes overhead characters

<sup>4</sup> For UPC, EAN and Postnet, density is a function of the minimum defined size

height of 0.8". Bar codes in document imaging should generally be at least 0.5" high and never less than 0.375" high.

The US Post office code (Postnet) has a standard short bar height of 0.1". Finding this very short symbol on a page image is problematic as it would be hard for software to distinguish it from a line of 7 - 10 point type. Postnet code reading should only be attempted in special case conditions where the location of the code is known in advance and the skew angle is at a minimum.

### **Orientation**

If all of your documents include bar codes of a known orientation relative to page contents, a strategy may be developed to detect and automatically correct document image orientation by detecting whether the bar code read is upside-down or is turned sideways. Once the bar code orientation is known, simple image processing functions can be used to flip or rotate the document image to its proper orientation. With a double sided scanner, this strategy can be extended so that even backwards facing pages can be accommodated automatically.

### **Symbology and Image Quality**

The ability to read a bar code image of a given quality will depend greatly on the symbology being read. Symbologies with variable bar widths like Code 93, Code 128, UPC-A, UPC-E, and EAN will require better quality printing and scanning to obtain reliable reads than will symbols printed in symbologies that employ discrete coding of bar and space widths.

## Printing Bar Codes on the Desktop

Just as scanning densities can limit the amount of data available for reading bar codes, desktop printing technologies present some of the same kinds of limitations on the printing side of the equation. While commercial offset printing can easily meet the tolerance limits for high density bar code printing, many desktop printers (including laser printers) may not print finely or accurately enough to produce readable bar codes.

For this reason, some third party bar code printing software packages specify that only certain printers (for example, dedicated bar code label printers) may be used and/or that bar code printing densities be limited in some cases. Failure to heed the limitations of these packages may result in unreadable or unreliable bar code symbols.

No matter which symbology, character density or printing method is chosen, always test the readability of bar codes produced by any proposed solution before making a purchase commitment. Testing should first be carried out on the printed bar codes using a bar code verifier. This is usually a hand-held, bar code reader specifically designed to test the validity and readability of printed bar codes. Then test printed bar codes by affixing them to sample documents, scanning them at the desired scanning resolution, and verifying that your document image capture and read software can indeed read the bar code data accurately.

### Warning: Bar Code Fonts

Appendix D lists several bar code printing solutions which will produce properly printed bar codes

Bar code fonts are available for use with most computer operating systems and laser printers. These fonts allow bar code elements from various symbologies to be typed into documents created in word processing, page layout and other programs. While printed output from these fonts may look like a bar code, these pseudo bar codes, will probably not be seen by scanners and bar code read software packages as valid. The reasons for this include:

- Distortion of the bar and space widths due to font scaling or printer resolution limitations
- Failure of the user to insert the proper start and/or stop characters

- Failure of the user to properly calculate and insert checksum characters

**The author does not recommend the use of bar code fonts.**

## ***FAX Input***

Faxing a bar code image can degrade its quality significantly. The figure below illustrates what can happen to a bar code image during faxing. The original bar code appears on the left and the faxed image on the right.



***Figure 12 - A Bar Code Before and After Faxing***

Most fax transmission is performed at a horizontal resolution of about a 200 dpi, and a vertical resolution of either 100 dpi (normal mode) or 200 dpi (fine mode). Therefore, for the purposes of determining the maximum permissible character density for bar codes in faxed documents, it is safe to assume that the image quality of a bar code in the horizontal orientation within a faxed document will be no better than one contained in a document scanned at 200 dpi regardless of fax mode. Bar codes faxed in the vertical orientation can be assumed to be no better than one contained in a document scanned at 100 dpi for normal fax mode, and at 200 dpi for fine fax mode.

## ***Special Symbology Considerations***

### **UPC-E**

UPC-E is a special version of the UPC code whose symbols contain six digits of data and are therefore physically smaller than the usual 12 digit UPC symbol. While each symbol is only 6 digits long (net of checksum and overhead), when a UPC-E symbol is read, a 12 digit result is returned as if a standard (UPC-A) symbol had been read. When a UPC-E symbol is read, its data is padded with zeros at the center so that the UPC-E symbol which encodes the digits 123456 will return 123000-000456. This zero suppression process is specific to the UPC-E message format.

### **Codabar**

The Codabar symbology requires that the implementer specify which of A, B, C or D will be used as the start and stop character. One must know which start/stop character is in use in order to properly read Codabar symbols. Therefore, when reading Codabar symbols obtained from outside sources, care must be taken to determine exactly which start/stop character is in use. Failure to properly determine this attribute will result in read errors. In addition, some bar code printing software packages do not automatically insert the required start/stop codes. In this case, data must be padded with the appropriate start/stop codes.

### **Postnet**

Postnet is unusual in that it uses fixed bar and space widths and that its information is encoded in the height of the bars. As a result, Postnet codes cannot be automatically discriminated on a page with mixed symbologies.

# Automatic Bar Code Reading

## Overview

Automatic bar code reading systems vary significantly in their capabilities and can be divided into three distinct generations based on these capabilities. First generation systems require the user to input the widths or relative widths of the bar codes to be read. While this is a tedious and error prone process, it is necessary since these width numbers are used to determine both the potential existence of a bar code and to determine actual bar code data by comparing these numbers to the widths of the bars and spaces actually encountered in the document under scrutiny.

Second generation bar code readers provide "learning" modules which set bar code width parameters by reading example documents known to contain representative bar code samples. This is a big improvement over user input of these parameters. For precisely printed bar codes, such as those found on job separator pages this system works quite well. For general input however, the true value of bar code widths can and does change from page to page. Factors which contribute to this variability include; scanner age and settings, paper condition, ink type and color, and most importantly page skew. In essence, any system which requires width parameter input either via manual entry or by learning module cannot be used to reliably scan bar codes over time.

Third generation products do not require predetermined bar width information to find bar codes. In addition, variations in widths can be accommodated from one bar code to the next. This means that skew, ink, paper and scanner variations can be accommodated with reliability and without user intervention. Reliability here means that a successful test case will result in long-term success of the bar code reading application.

## ***Why Grocery Store Bar Codes Scan So Well***

It has been remarked that grocery clerks do such a good job of scanning bar codes, why can't computers do as well? At first glance, using a scanner to ring up retail purchases may seem like a rather

simple proposition. In fact, it takes an excellent symbology, plus application knowledge, error checking and feed-back loops to make this system work reliably. Further, the system is reliant on the high quality supervisory functions provided by the grocery clerk.

Let's look at each of these elements to see how they contribute to a reliable outcome (as defined by the retailer who wants every item properly rung-up in a reasonable amount of time). When the actions and ramifications of the grocery clerk are understood, the parameters for successful bar code reading become self-evident.

### **Does a Bar Code Exist?**

Here is where our grocery clerk comes in. First of all, grocery clerks have implicit knowledge of what they are doing. They know (through observation or training), that nearly every packaged item that comes across their check-stand will have a single bar code. They also in the same manner know that a head of lettuce cannot have a bar code. This kind of implicit knowledge allows the clerk to instantly do a first pass sort as to whether or not to scan a particular item or to ring it up by hand, on the scale, etc.

In a document imaging system, we have to assume that any page might have a bar code so all pages should be searched. We can provide similar information when designing our imaging application such that one, and only one bar code will appear on each page. In this way, search times can be limited to the time necessary to find a single bar code.

Because the computer can only make programmatic assumptions about the presence or absence of a bar code, the use of checksums is very important. Not only do checksums validate the data itself but also validate the computer's assertion that the data is indeed a bar code.

The clerk also can rely on the fact that this bar code will be a UPC bar code. In the same way, a document imaging application can be designed such that only one bar code symbology is used. If the symbology is known in advance, then search algorithms specific to that symbology can be used rather than a slower, multiple symbology search algorithm.

## **Zoning**

Assuming that a bar code exists, the clerk will flip the package around until the bar code has been spotted and properly oriented for scanning. Human pattern recognition is nearly instantaneous and is used in conjunction with contextual knowledge. Not only would the clerk be able to spot the bar code patterns on the package, but she would know what a package looks like (even though they come in arbitrary shapes and sizes), would know package boundaries and, would know the difference between text and a bar code. All these kinds of knowledge contribute to the clerks ability to very quickly find the bar code.

This kind of instant pattern recognition is not something we can teach a computer to do yet. All we can do with a computer is program it to match patterns of document image pixels against programmatic descriptions of what bar codes look like at the pixel level. If a match is found that correlates to the description and it falls within programmed tolerances, it might be a bar code.

Imaging applications can be designed such a way that bar codes always appear at the same place on the page. When the imaging application knows where to look for bar codes and searches are confined to that small area, system performance will benefit. The action of the clerk placing the bar code over the scanner is called zoning in document imaging.

## **Scanning, Error Detection/Correction, Skew Correction**

Once the package is properly oriented, the clerk passes the package over the scanner with one of three outcomes;

- The item is properly rung-up and the clerk goes on to the next item
- The item's bar code is not readable and an audible error message is issued by the point of sale terminal. In this case, the clerk has the option to retry scanning (error correction) or to look at the human readable number imprinted below the bar code and manually type-in the number (back-up error correction).

In the case of a retry, the clerk will repeatedly scan at varying angles to achieve a good scan. This is the imaging system equivalent of making sure that bar codes appear in the page image with little or no skew.

- The other possible outcome is that the bar code is read but the system does not have a pricing database entry for the item. In this case the clerk falls back to manual entry. This is not a scanning error, but is another loop in the feed-back mechanism between the clerk and the retailer's computer system.

### **UPC - The Easy to Read Bar Code**

We mentioned that grocery store scanning employs an excellent bar code symbology. Nearly all grocery store items are coded with the Universal Product Code or UPC symbology. UPC is successful because it is an inherently a robust symbology which has been carefully implemented in a consistent fashion by most producers of retail packaging.

UPC embodies many attributes which make it an easy to read symbology document imaging. This discussion is not intended to recommend UPC for use in document imaging but rather to convey an understanding of why UPC has been so successfully applied. UPC's attributes include:

- The mandatory use of checksums
- A fixed number of data characters (12 for UPC-A)
- A fixed symbol height of 1" and width of 1.2"
- Minimal overhead
- A fixed aspect ratio for skew immunity
- Consistent observance of print tolerances and quiet zones by packaging industry

### **Summary**

Imaging systems don't have the luxury of a grocery clerk to punch-in bar code data if it is missed by the bar code read software. Therefore, attention to implementation will ensure that the bar code read success rate is high. When designing your application, if possible, use the following grocery store guidelines:

- Use only one bar code symbology
- Make sure that there is always one bar code per page Bar Codes in

- Make sure that every symbol includes a valid checksum and that the application verifies the data
- Make sure that every symbol has a fixed number of data characters
- Make sure that the quiet zones are strictly observed
- Make sure that the print quality meets the standards set for the symbology
- Make sure that the print density is conservatively chosen
- Make sure that the all symbols have a fixed aspect ratio that insures good skew immunity

Bar code read technologies today, if given enough time to exhaustively analyze every page image, can overcome violations of any or all of these guidelines. It is a simple trade-off of time versus quality. Careful attention to the points outlined above will pay off in fast and robust reading of bar codes from your document images.

## ***Optimizing Bar Codes For Automatic Reading***

While a good bar code reading technology can read bar codes in arbitrary locations, using arbitrary symbologies, and in arbitrary quantities (more than one to a page), the best results and fastest performance are achieved when bar code searches are optimized by supplying what is known about the target documents to the bar code reading application. In general, the more you know about the location, orientation, symbology, number of characters per symbol, size and number of bar codes in your document, the faster that processing can occur.

**Symbology** If known, the symbology should always be specified. While some technologies can read arbitrary symbologies, they generally run faster when searching for a particular symbology.

**Number of Symbols** If known, the number of bar code symbols appearing in each document should be supplied to the technology. If each document will have at most, one bar code symbol, then this should be specified. The bar code read software can stop processing as soon as the expected number of symbols has been encountered.

Otherwise, the software must continue to process the page image until all possibilities have been exhausted.

Similarly, if each page will have exactly one bar code (or any specific number of bar codes), then this should be specified. This way, when a page with fewer symbols is encountered, the page can be treated as an exception.

**Number of Characters Per Symbol** If you know the number of character that will be encountered in every symbol, specify the value when calling bar code read functions. When you design bar codes into a form or print bar code labels, be sure that all bar codes contain at least 6 data characters, that all symbols have that same number of characters, and that the bar code read functions in your tool kit are apprised of this value.

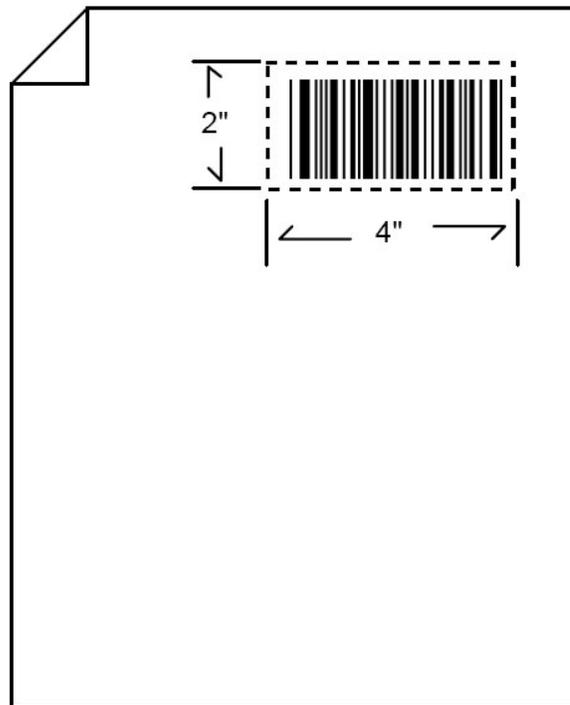
**Bar Code Skewing** While some bar code reading software can deal with a reasonable amount of skewing, bar codes that are badly skewed ( $>20^\circ$  from horizontal or vertical orientation) will impact reliability and will slow down processing. If you are using forms with preprinted bar codes, then skewing will generally not be an issue. However, if bar code labels are affixed to documents prior to scanning, make sure that the selected orientation is carefully observed when the labels are affixed.

**Orientation** If all of the bar code symbols in your documents will be horizontal (this is to say generally horizontal and right-side up or upside down), then specify that this will be the case. If most will be horizontal but a few will be vertical, then specify that horizontal orientation should be searched for first and that vertical symbols should be sought if a horizontal search fails. This will reduce bar code search and read times significantly. In a similar manner, if all or most documents will contain a vertically oriented bar code, reverse the strategy. If you will be checking bar code orientation to detect upside-down documents, then it is vitally important that bar codes be affixed to every document in the proper orientation.

**Relative Size** Regardless of the appropriate character density for a particular application, many bar code detection technologies require that bar code symbols be at least 2 times the height of a document's body text font, and that the bar code be at least 1.5 times wider than it is high. This insures that the bar code reading software will easily be able to distinguish bar codes from text.

**Location** Specifying the approximate location of the bar code(s) within a document image will speed processing. Bar code search time is proportional to the area of a document that must be searched. Therefore, if you know exactly where and how big bar code symbols will be, specify this information to speed up bar code operations.

For example, in the figure below, the 2" by 4" area surrounding the bar code symbol could be searched almost 12 times faster than the entire 8.5" by 11" page.



*Figure 13 - Page Schematic with Bar Code*

## Choosing a Development Tool Kit

When choosing tools for any document image capture development project, be sure that the tools you select are compatible with the language that you want to develop in. For example, not all Windows tool kits are created equal. While all DLLs are callable from the C language, many do not meet the structural conventions for calling from Visual Basic, Gupta, Power Builder, and other high-level languages under Windows 3.1.

A batch mode strategy that reads bar codes from within the document image is much more flexible and confers some distinct advantages to the system. However, when choosing a tool kit, the following bar code reading capabilities must be part of any candidate solution:

Automatic Location	Automatically finds bar codes anywhere on the document
Orientation	Horizontal and vertical
Skewing	Can read bar codes skewed to arbitrary angles
Multiple Symbols	Can read multiple bar codes on a page stacked vertically, horizontally, or in combination
Multiple Symbologies	Can read documents with multiple symbologies on a page
Mixed Sizes	Can read documents that include bar codes of arbitrary and multiple sizes on a page
Variable Element Sizes	Can read symbols whose bars and spaces vary in width and relative width within a symbol
Checksum Handling	Read a bar code's checksum
Search Strategy	Allows the specification of bar code search strategies according to location, orientation and size to speed up processing
Orientation Handling	Can implement automatic detection and correction of upside-down, rotated, and backwards facing document images
Reporting Detail	Reports back the location, size, symbology, read direction, and checksum of all bar codes encountered
Debug Support	Reports the results of partial read and generates an expected checksum for data read
Multiple Read Direction	Can read bar codes front to back or back to front in case of quiet zone violations
Symbology Support	PDF-417, Code 93, Code 39, Code 128, CODABAR, Interleaved 2 of 5, Non- interleaved 2 of 5, Airline 2 of 5, UPC-A, UPC-E, EAN, Postnet

# Bar Code Imaging Applications

## Several Bar Codes of Different Lengths

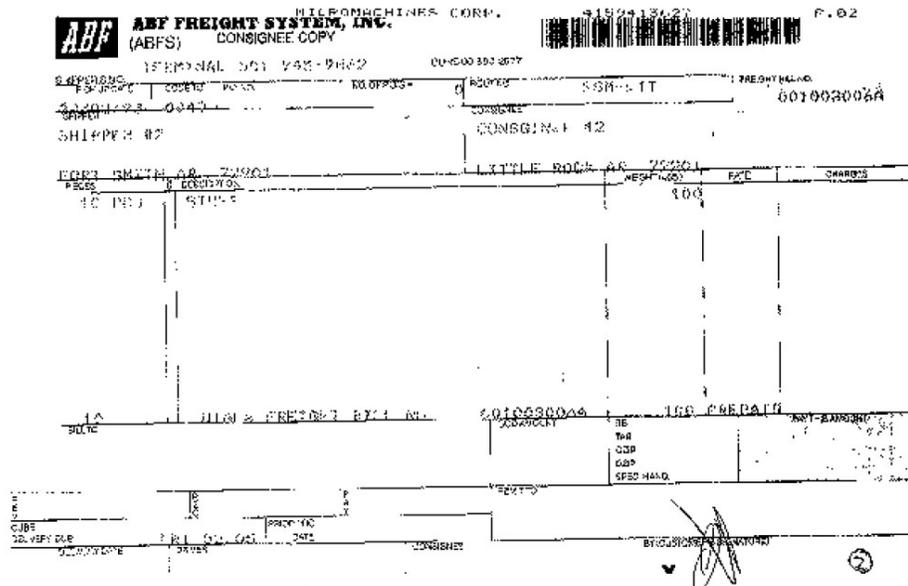
The Department of Defense (DOD) uses a standardized waybill form similar to the one shown below. Much of its pertinent data is contained within four bar codes. In order to automate processing of this form, a bar code reading technology must be able to read all of the bar codes on the form returning not only their contents, but their length and location within the document as well. With this information, meaningful data can be extracted from this form without human reading or keying.

000 000 000 000 000 000 000 000 000 000 000 000 A0ANNZSEA00015Y04030AKZ 9DGJ112 2L AARD000615 00009225												TOTAL BRZ 00.000000		INCHES 00.000000		SHIP TO V20635	
M/F USS EMORY S LAND AS 39												7302		049810		J 29 U	
												032		000		000	
FREIGHT CLASSIFICATION GLOVES & MITTENS NOI												MITTEN, HEAT		000		000	
												000		000		000	
												V20635		000		000	
333-952-13-3 01-092-0039 00015 EA A A 00000000												3		000		000	
21746614 LINE 001 OF 001												000		000		000	

Figure 14 - DOD waybill

## Automated High-Speed Forms Acquisition

Freight companies can make millions of deliveries each day. A proof of delivery document like the one shown below is used to capture a delivery signature for every shipment. Scanning of these proof of delivery documents with decoding of bar code data provides a way to link an image of the delivery signature to the shipment's database record. This creates a positive and easily retrievable proof of delivery which can be produced on customer demand, with no further physical document handling.



**ABF FREIGHT SYSTEM, INC.**  
 (ABFS) CONSIGNEE COPY  
 MICROMACHINES CORP. 9150413427 F.02  
 DU-600 303 3677  
 SHIPPER #2 CONSIGNEE #12  
 FROM: MICRO MACHINES CORP LITTLE ROCK AR 72201  
 TO: 10 INJ STU-5  
 WEIGHT: 100  
 VOL:   
 CHARGES:   
 BILL TO: MICRO MACHINES CORP  
 BILL FROM: ABF FREIGHT SYSTEM, INC.  
 SIGNATURE: [Handwritten Signature]

**Figure 15 - A typical proof of delivery document**

## Multiple Symbols Stacked Vertically & Horizontally

Large volume prescription users such as hospitals and rest homes need a method to accurately and rapidly record prescriptions. Pharmacies include a pull off bar code on the prescriptions. As the refill need arises, the bar code is peeled off the bottle and affixed to a re-order form. The re-order form is sent to the pharmacy via a fax. The bar code information, which tells what the prescription is by code number is read and re-order is filled without the need for manual re-keying of the data.

**Date:** 2-28-93      **Homecare Pharmacy Refills**      **Facility:** XYZ FACILITY

AC4502041T000962-RX238174 02/09/93 CRISGER, DORR X-DUR 20MG TAB 	HOURS AFTER MEALS AC9003061T001314-RX250102 02/09/93 HENDERSON, MARTA ... 	... AC9003221T012823-RX266113 02/09/93 HENDERSON, MARTA ... 
AC4502721T232079-RX188110 02/09/93 ELLIS, JOSEPHINE ZANTAC 150MG # 	... AC9003061T000872-RX266110 02/09/93 HENDERSON, MARTA ... 	... AC9003061T000872-RX266110 02/09/93 HENDERSON, MARTA ... 
AC4701201T012815-RX266077 02/09/93 LEFEN, SIR, M. NORR ... 	... AC9003131T034502-RX266120 02/09/93 ... ... 	... AC7150153T000285-RX227483 02/09/93 ROORE, MARTHA ... 

Figure 16 - Pharmacy reorder form (partial)

## Dynamic Changes in a Bar Code Symbol

Airlines handle a huge number of tickets which have value in the same way that currency does. In order to track both forms usage and specific ticketing information, the form serial number, is encoded into a partial bar code when the blank form is printed. Later, when the form is issued as a ticket, additional information is added to the bar code. The first half of the bar code is printed as part of the form and is therefore usually of good quality. The second half of the bar code is printed at the issuing ticket office or travel agency. The alignment of and the consistency of character density between the two halves of the bar code is apt to be noticeably different - yet the composite bar code must be read as a single symbol. Reading these bar codes requires a technology that does not depend on consistent character density or skew angles of elements within a symbol. Shown below is a typical airline ticket with a two-part bar code.



**Figure 17 - Typical airline ticket with two-part bar code**

## ***Scanner Quality Assurance Test Page***

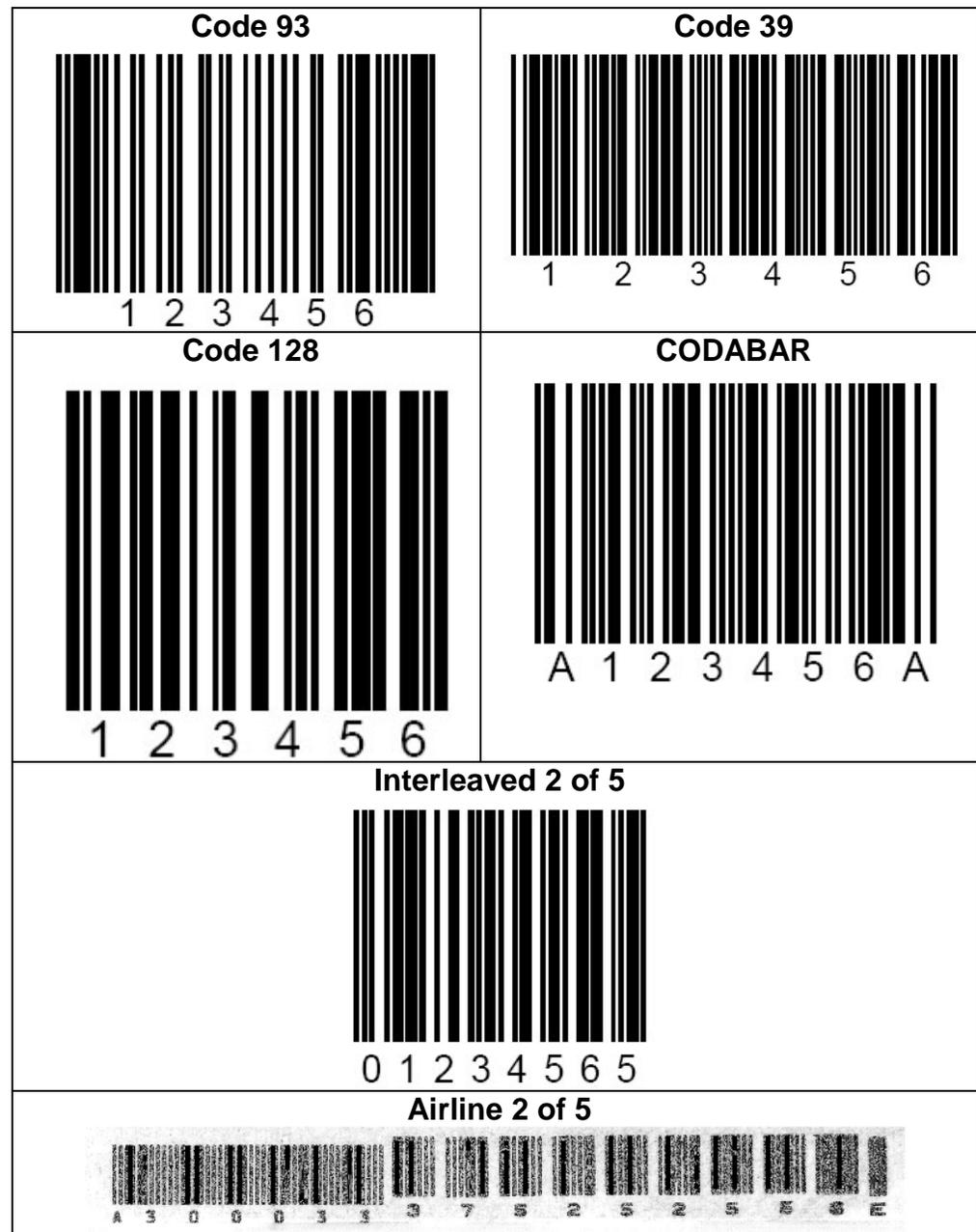
*Figure 18 - scanner quality assurance test page*

## *Typical Separator Page*

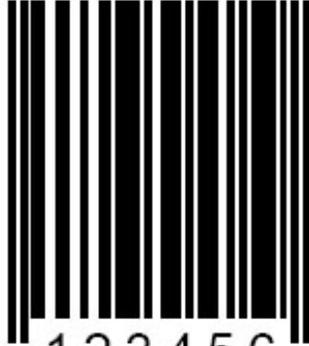
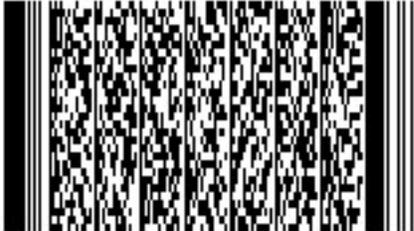
*Figure 19 – Typical Separator page*

## Appendix A - Bar Code Samples

This and the next page illustrate some of the visual characteristics of several popular symbologies.



*Figure 20 - Bar Code Symbol Samples*

<p style="text-align: center;"><b>UPC-A</b></p>  <p style="text-align: center;">0 00001 23456 5</p>	<p style="text-align: center;"><b>UPC-E</b></p>  <p style="text-align: center;">0 123456 5</p>
<p style="text-align: center;"><b>Postnet</b></p> <p style="text-align: center;">(Message: 94501-3511)</p> 	<p style="text-align: center;"><b>EAN-8</b></p>  <p style="text-align: center;">0123 4565</p>
<p style="text-align: center;"><b>EAN-13</b></p>  <p style="text-align: center;">0 000001 234565</p>	<p style="text-align: center;"><b>PDF 417</b></p>  <p style="text-align: center;">Message: PDF-417 can encode up to 1,850 ASCII characters or up to 2,725 digits in a single symbol</p>

## Appendix B -About All My Papers

### *Company*

All My Papers is the developer and distributor of software toolkits and applications that resolve issues with data integrity and data interoperability in Image Cash Letter exchange.

All My Papers has developed the products it sells and in addition it has acquired all the products, contracts, source code, patents and trademarks of Seaport Imaging, Xionics XipPrint Division, GGX and Idata International.

### *Contact All My Papers*

13750 Serra Oaks  
Saratoga, Ca 95070  
PHONE: (408) 366-6400  
FAX: (408) 366-6406  
[www.allypapers.com](http://www.allypapers.com)

[sales@allmypapers.com](mailto:sales@allmypapers.com)

[support@allmypapers.com](mailto:support@allmypapers.com)

## Appendix C - Further Reading

### ***Reference Materials***

You may find the following titles useful for reference when designing or implementing a document image capture and indexing system:

#### **Document Capture for Document Imaging Systems**

by David Black - Association for Information and Image Management (AIIM) (301) 587-8202, ISBN: 0-89258239- 1

#### **Reading Between The Lines**

An introduction to bar code technology by Craig K. Harmon and Russ Adams - Helmers Publishing (603) 924-9631, ISBN: 0-91126100- 1

#### **The Bar Code Book**

Printing, Reading, and Specification of Bar Code Symbols by Roger C. Palmer - Helmers Publishing (603) 924-9631, ISBN: 0-1911261-05-02

### ***Standards Organizations***

For further information on symbology standards, contact one of the organizations listed below:

#### **AIM -- Automatic Identification Manufacturers Inc.**

1326 Freeport Road, Pittsburgh. PA 15238 ANSI -- American National Standards Institute, 1430 Broadway, New York, NY 10018

#### **UCC -- Uniform Code Council**

7051 Corporate Way, Suite 201, Dayton, OH 45459-4924 EAN -- European Article Numbering Association, Rue des Colonies 54, Kolonienstraat, Bruxelles 1000, Brussels, Belgium

#### **FACT -- Federation of Automated Coding Technologies**

c/o AIM, 1326 Freeport Road, Pittsburgh, PA 15238

## **Appendix D - Bar Code Printing & Tagging Suppliers**

### ***Avery LabelPro For Windows***

Avery Dennison Corp. 20955 Pathfinder Rd. Diamond Barr, CA  
91765-4000 Toll Free: (800) 462-8379 Main Number: (909) 869-7711  
FAX: (909) 598-2905 Technical Support: (214) 888-2699

### ***B-Coder Pro for Windows***

T.A.L. Enterprises 2022 Wallace St. Philadelphia, PA 19130 800-722-6004; 215-763-2620 FAX: 215-763-9711 Generates bar codes on-screen and copies them to the Windows clipboard for pasting into files or documents created with other Windows programs. Supports all common bar code symbologies including PDF-417. DocuTag

### ***American Reprographic Systems, Incorporated***

220 West Congress Detroit, MI 48226  
(313) 965-5090 FAX: (313) 965-4883 A system to produce and dispense up to 30,000 sequentially numbered, bar coded labels per hour for litigation support applications. The system includes DOS software, a printer and dispensers which greatly speed up the process of affixing document labels. Bar Codes in Document Imaging

## Appendix E - Imaging Suppliers and Integrators

### ***Document Control Solutions***

616 South State College Fullerton, CA. 92831 800-800-6131  
mac@DOCsolutions.com (Steve MacWilliams)  
sales@DOCsolutions.com (General Sales Contacts) Document Control Solutions provides the following products and services:

### ***ImageTrax***

ImageTrax is a lifecycle tracking (records management) software for physical record stacks. ImageTrax is a database management, bar code and reports printing application, and a physical records inventory tracking system provided as a single package. ImageTrax integrates with Document Control Solutions' other two products which are described below.

### ***ColorBar***

ColorBar is a patent protected, color banding production system which produces color-coded, and bar-coded strip labels on demand to match existing file room banding formats, or to add color-coded banding to unbanded physical file stacks. ColorBar integrates with most records management software products.

### ***Quick-Filer***

Quick-Filer is a patented system of pocket-style folders and folder printing software for document image conversion projects. Each Quick-Filer application is custom tailored for the needs of a particular conversion project or client. Each Quick-Filer pocket folder can serve as a scanable document separator, and includes peel-off bar coded labels for each document type found in the imaging application. Quick-Filer folders incorporate black and white folder edge symbol banding, as well as many other features to make document image conversion and filing as painless as possible. Quick-Filer integrates with most records management software.