

---

# Bar Code Integration - a case study

David Taylor

Principal

*Asparona.com*

## Abstract

Organisations are constantly looking to drive costs down and improve efficiencies. This often results in a review of current processes and either refining or designing new process to realise some of these goals. Having invested in Oracle Applications it is important to try and find ways of making use of the features within the application suite to implement these processes.

This paper will discuss the opportunities and options for integrating bar code readers into business process flows. It will present information on how to integrate the readers to allow for the transfer of data to and from the readers.

Finally the paper will use a real life case study as an example of what can be achieved. The case study will cover bar code device selection, process design, implementation approach and current project status.

## Background

Asparona are working with a number of organisations who are looking to implement bar code solutions to integrate with their ERP systems. The system requirements differ significantly depending on the business processes to be implemented. It is very much a "horses for courses" approach that needs to be taken. The solution requirements depend on the process being automated, the frequency of the process, the level of knowledge/training of the users and so on.

## Overview of Barcode Hardware and Software.

### Barcode readers

#### Wand

The 'wand' is the simplest and least expensive input device available. It is durable and contains no moving parts. It must, however, come into contact with the bar code, which can present a challenge. If a bar code must be read more than once, it may become smeared or damaged and, in essence, unreadable. Also, a wand is "human powered," which means it must be held at the proper angle and moved at the proper speed. For these reasons, a wand is not recommended.

#### Charge couple device

The CCD, or "Charge Couple Device", is another common input device. A CCD is a very "aggressive" instrument, with a high ability to read bar codes quickly and easily. But it has two primary limitations. First, it has a short "read" range, and must be held no more than 1 to 3 inches from the bar code. Further, the CCD has a limited width, and will not read bar codes that are wider than the face of the input device. Therefore these devices are becoming obsolete and are not recommended.

#### Laser Scanner

The laser scanner is perhaps the most popular bar code input device. A laser scanner need not be close to the bar code to do its job. A standard range laser scanner can read a bar code from about 6 to 24 inches away, and a long range scanner can read one from perhaps 2 to 8 feet away. An extra long-range device can even read a bar code 30 feet from the device. Laser scanners vary in price. However due to the fact that these devices are attached to the host computer via a cable and therefore not suitable for remote applications, they are not recommended.

## Portable Data Terminal

Sometimes you must bring the computer to the bar code, particularly in an environment where there are multiple locations to be covered and a completely mobile device is required. A portable data terminal (PDT), a fully programmable hand-held computer, is necessary in such instances. Choosing the correct PDT is very important. When implementing a solution based on PDT's time and cost will need to be allocated to the design and build of the software that will run on the PDT itself.

PDT's run on either a version of MS DOS or Windows CE. More advanced PDT's provide touch sensitive screens and a full windows operating environment.

## Environmental Considerations

When selecting a reader or PDT there are a number of considerations that need to be taken into account. Listed below are some of the criteria that you can consider:

**Drop Height** – Height from which the reader can be dropped repeatedly without damage. A reader that is used at a work station does not need to be as rugged as one being used in a warehouse say.

**Temperature Range** – Manufacturers quote temperature ranges for their devices. The main limiting factor is the display (for PDT's). Below a certain temperature the display will either fog up or stop working altogether.

**Memory** – PDT's require memory to hold the program and the data files. Typically this would be around 5MB for program memory and 2MB for data memory. Additional memory can be added as an option.

**Display** – PDT's have displays. For simple devices this would be a 20 character by 8 line LCD display. More advanced PDT's have fully functioning miniature computer screens.

**Weight** – obviously a consideration if the device is to be carried around and used all day.

## Connecting the Reader to the Application

### Host Connection

Finally, bar codes are used to input data into a host computer system. There are three basic methods for achieving this.

- 1) Laser scanner directly attached to computer via a cable. This device is ideally suited to point of sale applications.
- 2) Portable data collection terminal remote device that can upload information via a docking station, infrared port or serial cable. This device is ideally suited for mobile application.
- 3) Wireless device. This type of device is suitable for a large warehouse environment and requires its own infrastructure which may be expensive.

### Communications Summary

Collected data must be transferred back to the host computer from the PDT, which can be accomplished in one of two ways. A radio frequency communications link provides online real-time data communications, immediate updates to databases and feedback to the operator. However, this solution could require a complex connection to the host, and you may need a less difficult or expensive method.

Mobile Computing data collection is a simpler and easier process.

In this method, data is collected in small batches and transferred from the portable into the host computer by a cable connection.

## Programming Readers

For programmable readers (or PDT's) it will be necessary to develop software to meet the specific requirements of the process. Two main methods for developing the programs are provided.

## **Rapid Development Tools**

Most manufacturers offer a rapid development tool. This tool allows non-technical users to build up the programs screen by screen, allowing for simplistic validation and control. Depending on the task at hand this can be a very good way to quickly develop the programs required.

## **Low Level Programming**

Programming for bar code readers is a specialist task. It is advisable to get input from an experienced organisation to help with the design of the software. This helps to ensure that the program flow will function efficiently and that the program is well written. Programs are written in a number of languages depending on the reader and the requirement. Examples include Basic, C, Visual Basic, C#.

## **Printing Bar Code Labels?**

Bar codes can be printed with existing dot matrix or laser printers, but with varying results. Oracle applications reports can be written that include the capability to print bar code labels. Thermal label printers, on the other hand, were designed specifically for the job and are built to produce high-quality text and graphics. They print at fast speeds and can be used to print one label at a time or an entire roll.

There are three basic printing methods.

### **Thermal transfer printing**

In this method, the print head transfers ink from a ribbon onto standard paper. The thermal transfer printer brings greater consumable costs because it utilizes a ribbon, but there is less wear and tear on the print head.

### **Thermal Direct Printing**

In this method, the print head is in direct contact with treated paper, and no ribbon is used. As a result, consumable costs are smaller, but the print head undergoes substantially more wear and tear. A ribbon produces less friction than paper, so a print head lasts approximately four times longer when printing in thermal transfer mode than in thermal direct mode. Print heads are considered consumable items and must be included in the overall cost of operation. Further, the width of the print head determines the maximum width of printed labels. Print heads are generally 2, 4, 6 or 8 inches wide. Every thermal label printer is driven by a proprietary programming language, which can make the bar code printing process challenging. However, bar code label software can make it easier by allowing you to create labels on the screen and print labels with data from various sources. Label software can be generic to all printers or specific to one manufacturer.

### **Laser Printing**

Most laser printers can also print bar code labels. There are a number of issues with using laser printers that are not dedicated to the job. Firstly the printers must be loaded with label stationery and often the labels get caught up in the mechanism. The printers are no good for printing one off labels. The quality is not so high and is unlikely to meet the requirements for 'A' grade quality labels.

## **Printing Labels from Oracle**

Again there are a couple of approaches to printing labels from Oracle Applications:

### **Direct using Oracle Reports**

Reports incorporating bar codes can be designed using Oracle Reports. The issues with using this approach are:

- Requires specialist skills to write the reports
- May require special printer drivers
- Inflexible.

### **Indirect using specialist label printing software**

Another approach is to use a software program specifically designed for printing bar codes. These programs take a raw data file as input (CSV or fixed format), or even direct connection to the database, to generate the label. The software handles all known bar code formats, supports all known label sizes and allows easy design

of custom labels. They also can support a wide range of bar code printers and their associated driver files. These products allow an end user to create a new label design using a simple drag and drop interface.

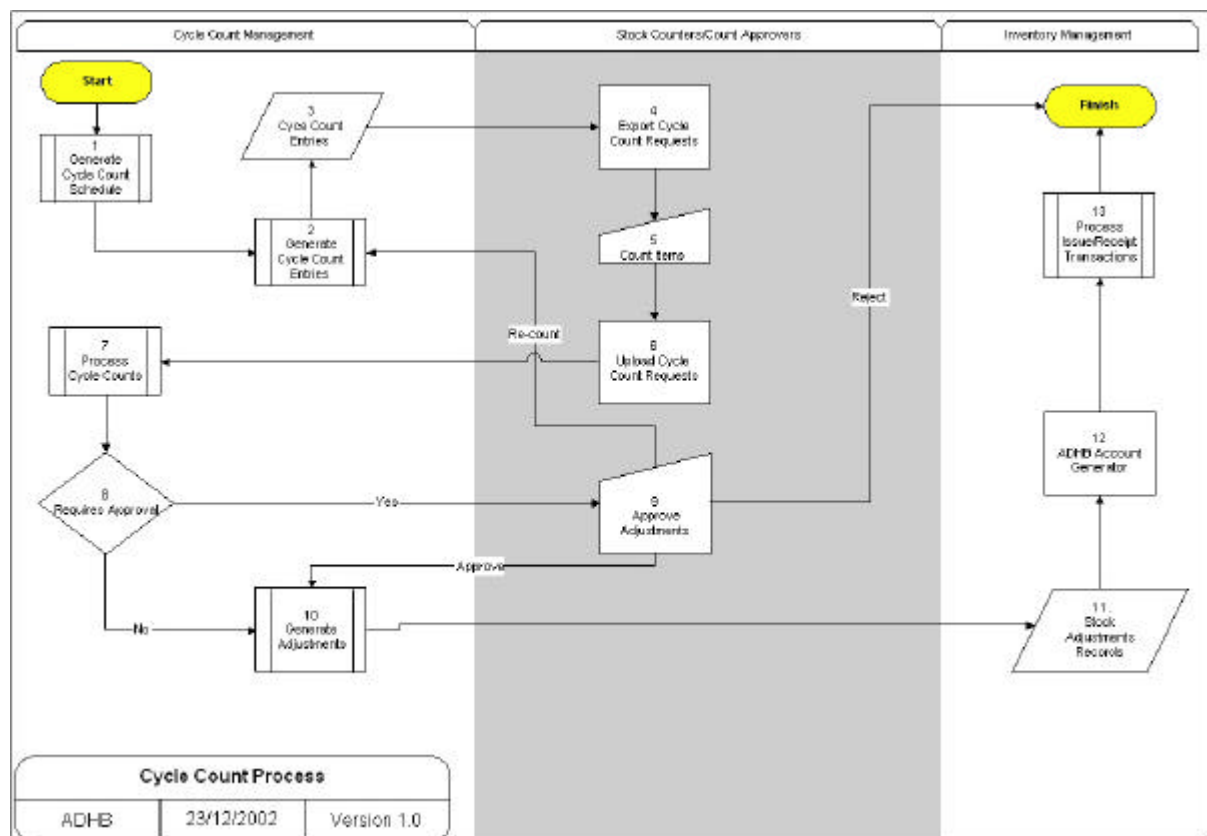
One example is a product called Bar Tender. This tool provides a software component that automatically triggers printing when a file (of a certain name or type) appears in a given directory a trigger fires and the associated label is printed.

## The Case Study

### Overview

The overall project scope was the full automation of the stock count to purchase order process. The organisation wished to implement a process that would enable stock administrators to count stock levels of a regular basis. The stores to be counted were to be managed using Oracle Inventory, stock receipts would be recorded but issues would not be. The stock counting would enable the organisation to track the items being issued and to generate re-order requests using the min-max planning function. There were 120 stores (subinventories) that needed counting and at the peak 12 stock clerks who would be doing the counting.

Bar Code readers were to be implemented allow the store clerks to download stock count requests to their reader, for them to go out and count the items and finally to upload the counted items.



## **Implementation Overview**

### **Cycle Counting**

We decided to implement the standard Oracle Inventory cycle count feature. This will allow the items in each store to be scheduled for counting on a regular basis. The system would be configured to automatically schedule count requests for items depending on their 'ABC' assignment. A items would be counted every day all the way down to 'E' items which were to be counted monthly.

The only change to the cycle count process was how the adjustment transactions were treated. When an item is counted and the stock level is different from the count quantity, if the count is lower then an issue transaction should be generated, higher and receipt transaction generated. Instead of the standard cycle count adjustment and its associated accounting implementations, we were to "intercept" the transaction and change it to one of two new transaction types representing the issue or receipt. The transactions would also control the GL account that was to be debited or credited.

### **Bar Code Reader**

The process required the stock clerks to be out and about at the various stores. The solution required some sort of PDT that allow the operator to roam freely, guiding them to the stores and items that needed to be counted. The program requirements were basic so the readers only needed to offer a simple screen (8 by 20 screen), less than 6MB of program memory and, we calculated, around 0.5MB of data memory. Full wireless technology was not required.

The environment the reader was to operate in was generally kind but some of the stores were refrigerated. Simple tests were performed with the three vendors selected, including a test of operation in a freezer. Selection of the final device came down to its ability to survive the temperature range, drop height, size and ease of use. Just as important was the ability and experience of the supplier when it came to designing and developing the software to run on the reader.

Three suppliers were evaluated for the project, each offered bar code readers, printers and software development skills. We selected the Dolphin 7200 reader from DataCol Solutions. This rugged PDT met all our requirements and we were confident of the ability of the local supplier to help us develop the software and to provide ongoing support.

### **Bar Code label Printing**

A single label printer was purchased for the project. This will handle the initial printing of the several thousand labels in the stores and the ongoing ad-hoc requirements. It was decided to implement the specialised bar code printing tool, Bar Tender, to handle design and printing of labels. A simple concurrent program was written to generate a CSV formatted file containing all relevant details that might be required for the labels. The file produced is saved to the PC and the Bar Tender software picks it up and prints the labels. This software allows anyone with appropriate access to set up a new label design or change an existing one.

### **Communication with Oracle**

We looked at a number of ways of connecting the reader to Oracle Applications. The interface needed to be simple to use, reasonably fault tolerant and as integrated as possible. We decided to develop a client application written in Java that would run on the PC's that the readers will connect to. The client application allowed the operator to identify themselves to the system and then to download the cycle count requests assigned to them. The program connects to Oracle Applications directly (using JDBC, which means no other Oracle software is required on the PC) to access the data. The downloaded data is formatted into a CSV file and then transferred to the reader using a commonly used protocol called Y-Modem. The entire operation is seamless and the operator only has to enter their operator code, plug the device into the PC docking station and press 'GO'. Upload of the counted items works in much the same way.

The Java client uses the standard inventory cycle count API's to export the cycle count data and to upload the counted items. A third party java class was purchased to provide the Y-Modem file transfer and serial IO handling required to communicate with the bar code reader (PDT).

## **Current Project Status and Summary**

### **Current Status**

The project is on schedule to go live at the end of March. The full scope of the project includes min-max planning, full automated purchasing and EDI with two key suppliers. I should add we are also upgrading from 11.5.3 to 11.5.8 at the same time! The 11.5.8 upgrade has completed which, by the way, was very smooth and relatively trouble free.

The Java client is almost complete. Unit testing of the integration with the bar code device and connection to Oracle has completed and was required before final sign off of the PDT purchase. System testing of all processes is in process currently. Design of the software for the PDT has completed and, now the PDT purchase has been approved, development is underway. We expect the full solution to be ready for acceptance testing within the next week.

### **Summary**

This project involved what could be described as tight integration between bar code reader and Oracle Applications. A purpose built software client was developed to connect the application with the reader. This approach was taken because of the process being implemented, and the type of users involved (ie not required to log into Oracle Applications).

It is also possible to integrate Oracle Assets with a bar code solution using ADI and removing the need for an interfacing application. Here ADI is used to download the asset inventory to the reader. A file is generated and then a client tool is used to download the data to the reader. Count results are uploaded as a file to the PC and then ADI is used again to upload the data.

### **About the Author**

David Taylor has been working with Oracle Applications for 11 years and has been involved in numerous implementation and general consulting assignments in New Zealand, Europe, and Asia Pacific. Within the Oracle arena David is recognised as one of the leading authorities on Oracle Applications. David is a principal member of Asparona Limited an independent IT services and consulting organisation.

Originally founded by two ex-Oracle Corporation consultants, each with more than 12 years experience of Oracle Applications, Asparona now employs over a dozen professional consultants. Our consultants have at least 5 years experience working with Oracle Applications and many more years' business or industry experience. With over 30 implementations of Oracle Applications in New Zealand, the UK, Europe and Asia Asparona is recognised as a leading independent specialist Oracle Applications consulting organisation in New Zealand.

These skills cover the full range of roles required for a successful implementation of Oracle Applications including; project management, business consulting, customisations, interfaces, specialist product knowledge, technical infrastructure planning, training and change management. To this skill base, Asparona has now added the necessary infrastructure and capability to host and serve applications as an Application Service Provider or ASP.

[www.asparona.com](http://www.asparona.com)