Inventory Applications

Research Report Volume II:
Bar Code Application Pilot Projects

Mozart Menezes,
Bob Blackmon
and
Lansford Bell

Department of Civil Engineering
Clemson University
December 1997

Technical Report Documentation Page

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
FHWA-SC-97-03		

4. Title and Subtitle	5. Report Date
Inventory Applications Research Report Volume II Bar Code Application Pilot Projects	December 1997 6. Performing Organization Code
7. Author(s) Mozart Menezes, Bob Blackmon and Lansford Bell	8. Performing Organization Report No.
9. Performing Organization Name and Address Department of Civil Engineering Clemson University Clemson, SC 29634-0911	10. Work Unit No. (TRAIS) 11. Contract or Grant No.
12. Sponsoring Agency Name and Address South Carolina Department of Transportation P.O. Box 191 Columbia, SC 29202	13. Type of Report and Period Covered 14. Sponsoring Agency Code

15. Supplementary Notes

Prepared in cooperation with the US Dept. of Transportation
Federal Highway Administration

16. Abstract

The South Carolina Department of Transportation (SCDOT) performs numerous engineering, accounting, and management functions that are data intensive. Two pilot projects were initiated by Clemson university to demonstrate the use of bar code technology as a data acquisition tool within a state transportation agency. The pilot projects were related to asset inventory control within a research and materials laboratory, and materials inventory control and reporting within a central supply depot. Both applications involved the use of a portable programmable bar code scanner. Computer programs were developed capture and display data in the scanner unit, transfer of data between the scanner and a desktop computer, and generate useful reports.

Bar code labels were placed on assets in the laboratory and shelf bins in the supply depot. Demonstration tests indicated that the bar code systems developed by Clemson University would permit inventory applications to be performed approximately 15 times faster than what was required using a manual process.

17. Key Words	18. Distribution Statement
bar code technology inventory applications asset control	No restrictions. This document is available to the public through the National Technical Information service, Springfield, VA 22161

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price
Unclassified	Unclassified	48	

TABLE OF CONTENTS

LIST OF FIGURES

Chapter I

INTRODUCTION

During the 10 May 1995 Workshop sponsored by SCDOT and Clemson University on the use of bar code technology, participants identified two activities where bar coding technology could be used to simplify inventory processes. The potential inventory applications were in the SCDOT Research and Materials Laboratory (RML) and the SCDOT Supply Depot. Clemson University was awarded a research contract to develop pilot projects at both locations to demonstrate the use of bar coding technology in facilitating inventory control and improving operations.

Research and Materials Laboratory Inventory System

The Research and Materials Laboratory in the South Carolina Department of Transportation has the primary responsibility for quality control of the materials used in highway construction. The Laboratory inventory consists of equipment used to support the testing of soil, asphalt, concrete and aggregates as well as communications and computer support hardware.

All physical property in the South Carolina Department of Transportation (SCDOT) Research and Materials Laboratory (RML) is inventoried annually to satisfy two complementary systems. Each system requires an annual report. All items with an original value of $500 or more are maintained in SCDOT records. RML is accountable to Headquarters SCDOT for such property which must be annually inventoried and reported. Less expensive property controlled by RML is usually inventoried at the same time. Property accountability is recorded with reference to its value and its physical location within the RML or at an external location. RML retains property accountability for all equipment loaned to district offices and for performing field tests. Loaned

items may include truck-mounted communication and computer equipment. A significant effort is made each year to locate, identify, and record the existence of all physical property. The inventory is executed by surveying each space in the RML facility and manually marking up a hardcopy of the previous inventory. A search is then made to locate all missing items which have been relocated to other spaces during the past year. If surplus property is found in a space, a search is made of the inventory to determine where the property should have been. If property has been relocated, appropriate changes are made in the inventories for the two spaces involved. If an item is missing from a space, a search is made based on advice of personnel working in the space. When the missing property is located, appropriate changes are made in the inventory and the general inventory can continue. Equipment stored outside are inventoried in the same manner. Individuals using RML property off-site are contacted and asked to confirm that they still have the property. District Offices are asked to inventory and confirm that the property is still under their control. Data collected during the survey is keyboarded into a desktop computer. There are opportunities for introducing errors in the keyboarding operation and marking up the inventory pages. The process takes several weeks to accomplish. Labels on construction equipment and test equipment exposed to weather and construction conditions are often difficult to read.

Supply Depot Inventory System

SCDOT has a central procurement department that prepares contracts and acquires commodities for all districts within the state. The Supply Depot is part of this central procurement chain, and it is the supply central point for all SCDOT district warehouses within the state.

Operation of the Depot requires a complete physical inventory of all items on an annual basis. This generally requires closing the Depot for a week while four two-men teams make the

inventory. Teams are given sections of the master inventory list and they inventory commodities in order of the sequence in which they appear on the list. Stock is placed on the shelving by type of commodity (e.g. oil filters) but not stored sequentially by stock numbers. Overstocks for small items are generally stored on the top shelf but overstocks for large items may be stored in several places in the Depot. The inventory process requires locating the various places in the Depot where the stock is stored and making a count at each location. Much of the time required to complete the inventory is spent walking, searching, and moving scaffolding to be able to make a physical count of items on the shelves in plastic bins. The master list is marked-up and changes are keyboarded into a computer. Data are transferred through the network to headquarters. Data processing maintains the official inventory which reflects receipt of new commodities and shipping out commodities to District Offices. All differences in the inventory and the official inventory are noted and returned to the Depot for reconciliation. The challenged inventory items are then reinventoried which may require a more thorough search. This process is repeated until all challenged items are reconciled. Sources of error in the process include misreading the marked-up inventory sheets, data entry, and forgetting where overstocks are stored if in multi-locations. The Depot houses about 1350 different items either on pallets or in plastic bins; tires are stored by size in an open area. Shelving may have 3 to 5 shelves per rack depending on the size of the commodities being stored. Large volume commodities are stored on pallets in open areas.

Research Objectives

The primary objective of the research described herein was to demonstrate the use of bar coding technology in inventory management systems to improve operations in the RML and the Supply Depot .

Research Methodology Overview

Research and Materials Laboratory

The research began with the definition of the system's proposed operating characteristics through a series of meetings between Clemson University representatives and SCDOT RML personnel. A review was made of SCDOT current inventory practices including the related computer software. A labeling scheme was developed that would satisfy RML and SCDOT requirements. Criteria for a new database procedure was also developed. As the software development work was accomplished, it was demonstrated to RML personnel for evaluation and feedback. Pre-printed labels were obtained for the two different environments. RML personnel were given a demonstration of the final software package. A performance test was then made at RML to determine the time required to survey a sample of the physical property. Documentation was completed and furnished to RML.

Supply Depot

Meetings were held with Supply Depot personnel to determine system requirements including the information to be stored on labels. Individual items should be labeled, if possible, to simplify preparation of shipping documents. Large labels should be attached to the shelving edge or edge of plastic bin for inventory purpose. Labels should include stock numbers, nomenclature of items, and unit of measure. Labels are now printed in the Depot by linking into main frame, identifying the labels needed by stock number, and printing the labels in the warehouse office. Management wants to retain the ability to print labels for attaching to individual items as they are received. However, they do not have the ability to print bar codes on the labels. Data Processing at

Headquarters has the responsibility of incrementing and decrementing the official inventory as commodities are received and shipped on a daily basis.

Report Organization

Chapter II briefly addresses bar code technologies applied in this project. Development of the RML system is addressed in Chapter III and the Supply Depot system in Chapter IV. Conclusions of the research project are presented in Chapter V.

Chapter II

BAR CODE TECHNOLOGY

A bar code is a graphical representation of data in a machine readable format. The coded data may correspond to data base entry such as a part number, or it may be comprised of an arbitrary but unique set of numbers, like a license plate, that are used to locate other information in the database. Bar codes utilize combinations of wide and narrow bars and spaces to achieve the coded representation of the data.

Bar code provides a means of rapid data entry. A skilled data entry clerk can enter data at the rate of 2 to 5 characters per second. Bar coded data can be entered at the rate of up to 30 characters per second by unskilled personnel. Also, data entered from bar codes is essentially error free, as compared to manual keyboard data entry which produces on the average one keystroke error for every 300 characters entered.

A wide range of scanning devices and related hardware components are available for bar code data entry. A portable data terminal (PDT) is a small hand held computer with keyboard, display screen, and laser scanner unit. The PDT is fully programmable, but due to memory limitations within the PDT units, the programs are normally compiled as executable code. Clipper programs were developed and compiled for use within the PDT units as part of this research. It should be noted that Janus 2020 PDT used as part of the research was configured with a 2 MB memory card, which is sufficient for the compiled programs and an inventory file of approximately 8500 items. An inventory of an entire facility can therefore easily be completed (the RML has 1388 items) before performing a data transfer operation.

Data can be transferred between the PDT and an office computer in a number of ways. As part of this project it was determined that the best way to accomplish the data transfer was through the use of an installed memory card reader within the office computer.

Bar code labels can be purchased in a preprinted format from third party vendors, or, printed by the user from inexpensive software packages that are compatible with existing computer files and programs. Preprinted labels are available for withstanding most any harsh exterior environment.

Chapter III

THE RESEARCH AND MATERIALS LABORATORY

INVENTORY SYSTEM

System Requirements

A meeting was held with RML personnel to establish a set of overall system requirements. System parameters developed the meeting included the following:

1. The system will utilize the SCDOT Janus 2020 portable data terminal manufactured by Intermec.
2. The data base will be based on Microsoft Access software.
3. A simple method must be available for transferring data between the portable terminal and a computer on the SCDOT network.
4. The SCDOT label numbering system must be retained for all reports to headquarters.
5. The system must be capable of reducing time required for data collection and reconciliation.
6. The system must be capable of handling two separate inventory reporting systems; one for SCDOT headquarters and the other for RML management.
7. Pre-printed labels will be used to meet two different environmental conditions.
8. The location of all inventory items must be recorded.

Originally, RML personnel wanted the capability of printing their own labels as required. This would require a printer with two types of media; one for labels for interior use and one laminated with a protective covering for exterior use or under severe conditions. Many of the labels on property exposed to weather or severe conditions currently need to be replaced. New property items are added and some items are disposed of during the year so the numbering system must be designed to easily accept changes. Based on projected label requirements, it was decided that it

would be uneconomical to establish an on-site labeling capability. Two types of pre-printed labels were purchased using two series of sequential numbers. The approved numbering system was based on two numbering systems. Property on the SCDOT inventory could be identified with labels starting from the high end of each sequence. RML property could be identified with labels starting from the low end of each sequence. The software could then sort the inventory by the appropriate range of numbers.

Data transfer between the portable terminal and computer will be accomplished by physically transferring the memory card between the two units. Data could be transferred by wire but it would require additional software and more knowledge of computer operations.

System Development

SCDOT Data File Alterations

SCDOT originally maintained an equipment inventory data file in Microsoft ACCESS format. This data file was altered by Clemson University for integration compatibility with the compiled dBASE software that resides in the Janus unit. The file alteration consisted of labeling all fields in upper case letters and adding two fields, LABEL and CHECK. The revised data file fields and contents are illustrated in Figure 1. The bar code identification number will reside in the LABEL field. The CHECK field stores codes that indicate the status of items during the inventory process using alphabetic characters as follows:

1. C denotes an item that has been inventory scanned for which record information matches the physical information;
2. W denotes an item that has been scanned but the physical location has been changed;
3. U denotes an item that has not been scanned.

ORG	NUMBER	LABEL	DESCRPT	CHECK	LOC	VALUE	REP_COST	REV	COMMENTS
RML	0090-03-02		HOM. GENERATOR		203	308.88			
RML	0090-23-03		DAYTON GENERATOR		202	427.47			
DOT	0090-24-06		POWER GD GENERATOR		X	1021.65			25-18-1
RML	0158-08-01		TRANSFER PUMP		11 B	435			
RML	0351-00162		TRANSIT & TRIPOD		117	320.9	1287	96	
DOT	0367-00049		FIFTH WHEEL		23	5075	5583	93	
DOT	0367-00075		DIST MEASURING DEV		23	624.75	684	93	3-1-71
RML	0367-00205		DIST MEASURING DEV		23				5-1-741
DOT	0367-00221		DIST MEASURING DEV		23	553.75	554	96	5-1-740
DOT	0367-00222		DIST MEASURING DEV		23	553.75	554	96	5-1-619
RML	0401-00010		DBL EXECUTIVE DESK		15 A	70	420	93	
RML	0401-00098		EXECUTIVE DESK		10	100	420	93	
RML	0401-00126		EXECUTIVE DESK		14	35	420	93	
RML	0401-00173		SECRETARIAL DESK		15 A	25	595	93	
RML	0401-00570		EXECUTIVE DESK		20	97.85	420	93	
RML	0401-00614		EXECUTIVE DESK		113	96.33	420	93	
RML	0401-00627		EXECUTIVE DESK		116	133.03	420	93	
RML	0401-00638		EXECUTIVE DESK		117	140.6	420	93	
RML	0401-00658		EXECUTIVE DESK		6	140.6	420	93	
RML	0401-00664		EXECUTIVE DESK		13 B	140.6	420	93	
RML	0401-00713		EXECUTIVE DESK		15	160.5	595	93	
RML	0401-00734		EXECUTIVE DESK		11 A	159.65	420	93	
RML	0401-00760		EXECUTIVE DESK		113	159.86	420	93	

Figure 1. Revised SCDOT Data File

The SCDOT data file was also modified to establish field type, width, and decimal place designations for dBASE compatibility. The file requirements are shown in Figure 2. If applications are developed for other SCDOT installations, the ACCESS file that resides on the office PC must have these characteristics.

For the Research and Materials Laboratory application, the data files described above have been created. In addition, a macro has been written that permits the exportation of Access files to a dBASE format. The macro can be used by opening an Access form tab, selecting Inventory, and clicking on the export button.

Assignment of Bar Code Label Identifiers: Start.exe

As stated above, the Clemson University developed system permits the assignment of arbitrary bar code identifiers (numbers) to the units of equipment to be inventoried. The system then performs translations relating the DOT or RML number to the bar code identifier.

To initialize the Clemson University System, the user must enter bar code identifiers into the PC Access data file to create the link between the bar code identifier and the DOT or RML number. This is easily accomplished using the Start.exe program that resides in the Janus unit (see Figure 3 for the process flowchart). At the DOS prompt of the Janus unit, enter

start FILE

Where FILE is the name of the file being used. SCDOT may maintain multiple files within the inventory system, for example one file for each building or organizational unit. The file provided as part of the demonstration system is named Inventor.

	Field	Type	Width	Decimal Places
1	ORG	Character	3	
2	NUMBER	Character	10	
3	LABEL	Numeric	11	0
4	DESCRPT	Character	25	
5	CHECK	Character	1	
6	LOC	Character	7	
7	VALUE	Numeric	19	5
8	REP_COST	Numeric	19	5
9	REV	Character	2	
10	COMMENTS	Character	22	
11	ROOM	Character	36	

Figure 2. SCDOT Data File Parameters

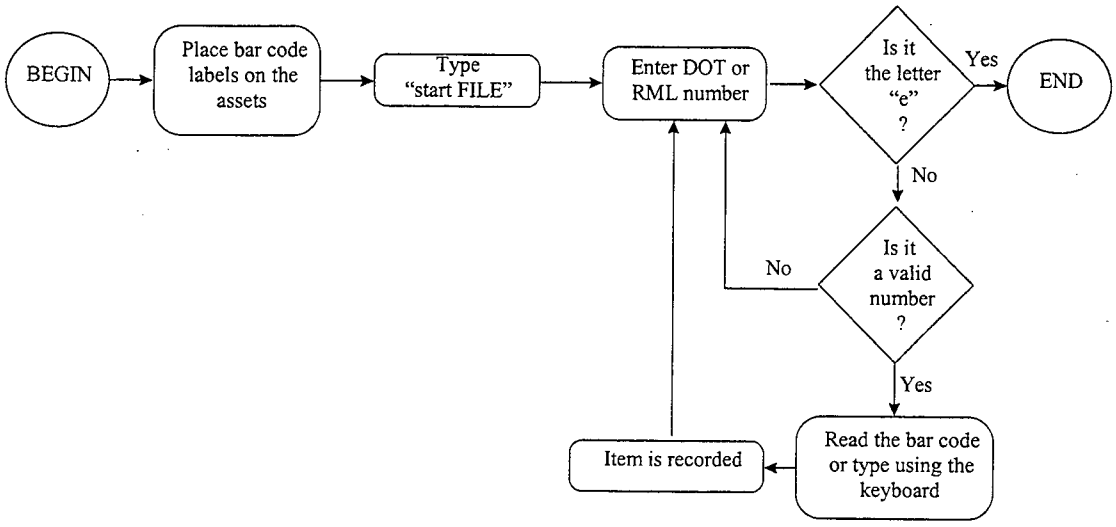

Figure 3. Using Start.exe

After entering the start field command the Janus unit will display a field for entry of the DOT or RML asset number which resides in the NUMBER field of the data file. After entering the asset number the program will check to see if the item exists in the data file. If so, a bar code number will be requested. The bar code number can be entered through either bar code scanning or keyboard entry. If the file does not contain the DOT or RML number the screen will prompt the user for another entry.

This program can be terminated by pressing "e" and the ENTER key. The file residing in the Janus unit can then be imported into the ACCESS office PC data file and exported to the Janus unit at a later date as required. A listing of the CLIPPER commands that constitute the Start.exe program appears as Appendix A of this report.

Asset Inventory: Lab.exe

The asset inventory process is accomplished using the Lab.exe compiled program that resides in the Janus Unit. Figure 4 shows the process flowchart. This program performs the following functions:

1. creates a current asset inventory with physical locations;
2. identifies SCDOT assets that can not be found;
3. identifies items that reside within physical locations but are not in the inventory file; and
4. permits linking of assigned numbers to multiple control numbers.

To begin program use, at the Janus unit DOS prompt enter

Lab FILE I

Where FILE is the name of the data file. The letter "I" (which is case sensitive) is an option and it is used to perform the initial inventory. Using the program in the "I" mode places the code U in the CHECK field for all items in the file, denoting the fact that all items are at this time "unchecked". This option must be used to initialize each overall inventory process.

The program will prompt the user for a room number. This data entry must begin with the letter R followed by the room number. The asset inventory of all items then proceeds without entering the room number until moving to another room. The room number can be entered manually, or by scanning a bar code (printed with the room number preceded with the letter "R") that is affixed to a door jam or other location. After completing the inventory, enter the letter E to end the session.

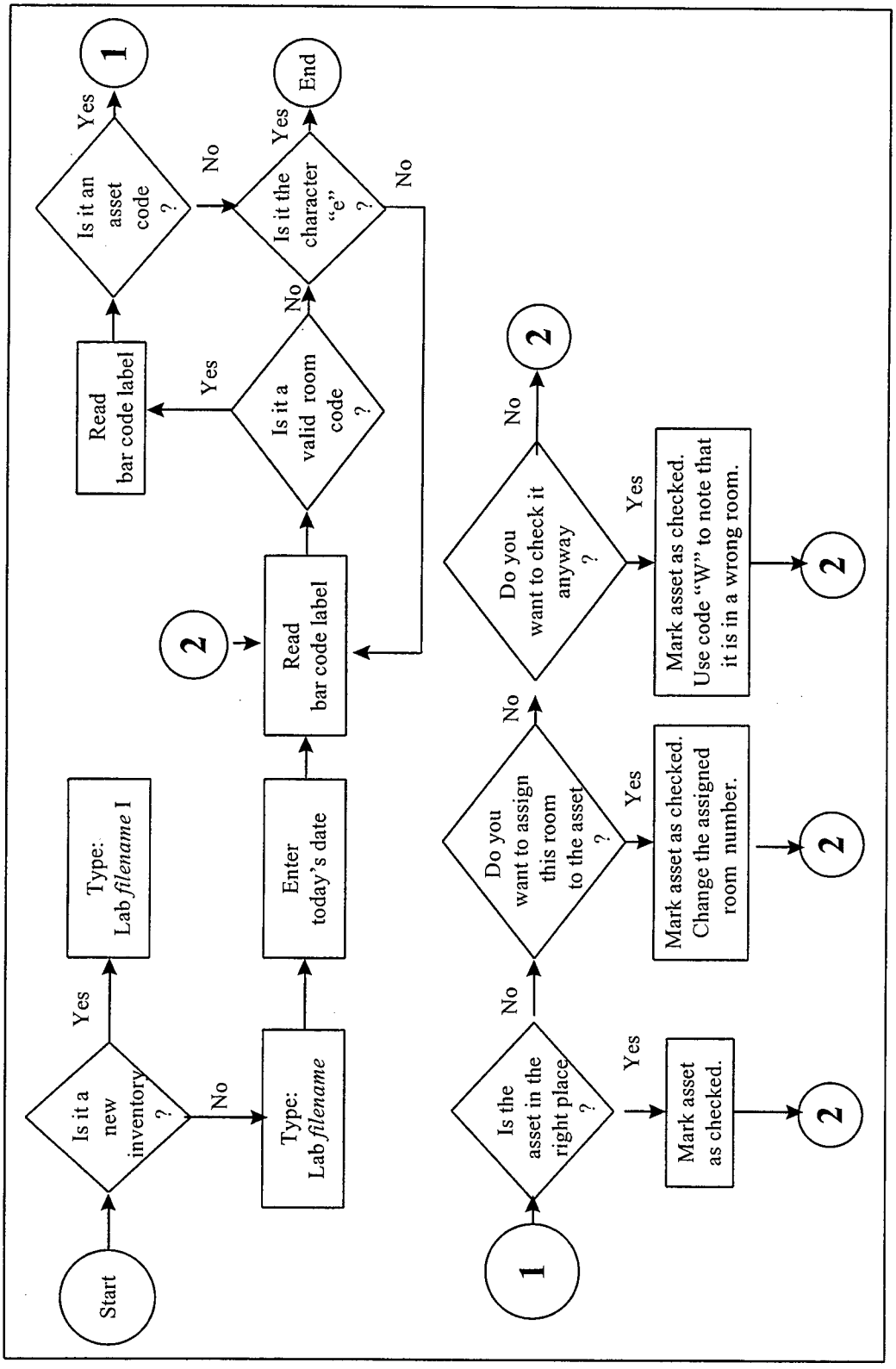

Figure 4. Using Lab.exe

The inventory proceeds by simply scanning all bar code numbers within a room. The program in the Janus unit searches the data file for the bar code number, compares the actual location (LOC field of the data file) to the location stated in the file. The user has a number of options to pursue if the item is not in the location specified by the file:

1. override the previous data entry, denoting a change in location;
2. placing a code in the CHECK field to denote improper location; and
3. leave CHECK status as "unchecked" for later resolution of the problem.

The Janus unit display will continuously display data pertaining to the current room location, including the total items expected to be encountered in that room, the number of items in "checked" and "unchecked" status, and the number of items residing in the current location that were assigned to another location. The source code for the Lab.exe program is contained in Appendix B of this report.

Access Reports

After importing the inventory data file into the office PC in Access format any number of reports can be generated by persons familiar with the Access software package. All data can be manipulated in Access and the import/export options exercised from screens similar to those shown in Figure 5. However, eight reports were developed by Clemson University as a result of requirements stated by SCDOT Laboratory personnel. See Table I for the list of the Research.mdb contents. These reports have capabilities as follows:

1. items inventoried and tagged C and W by number order;
2. items inventoried and residing in the proper location;
3. items sorted by bar code label number;

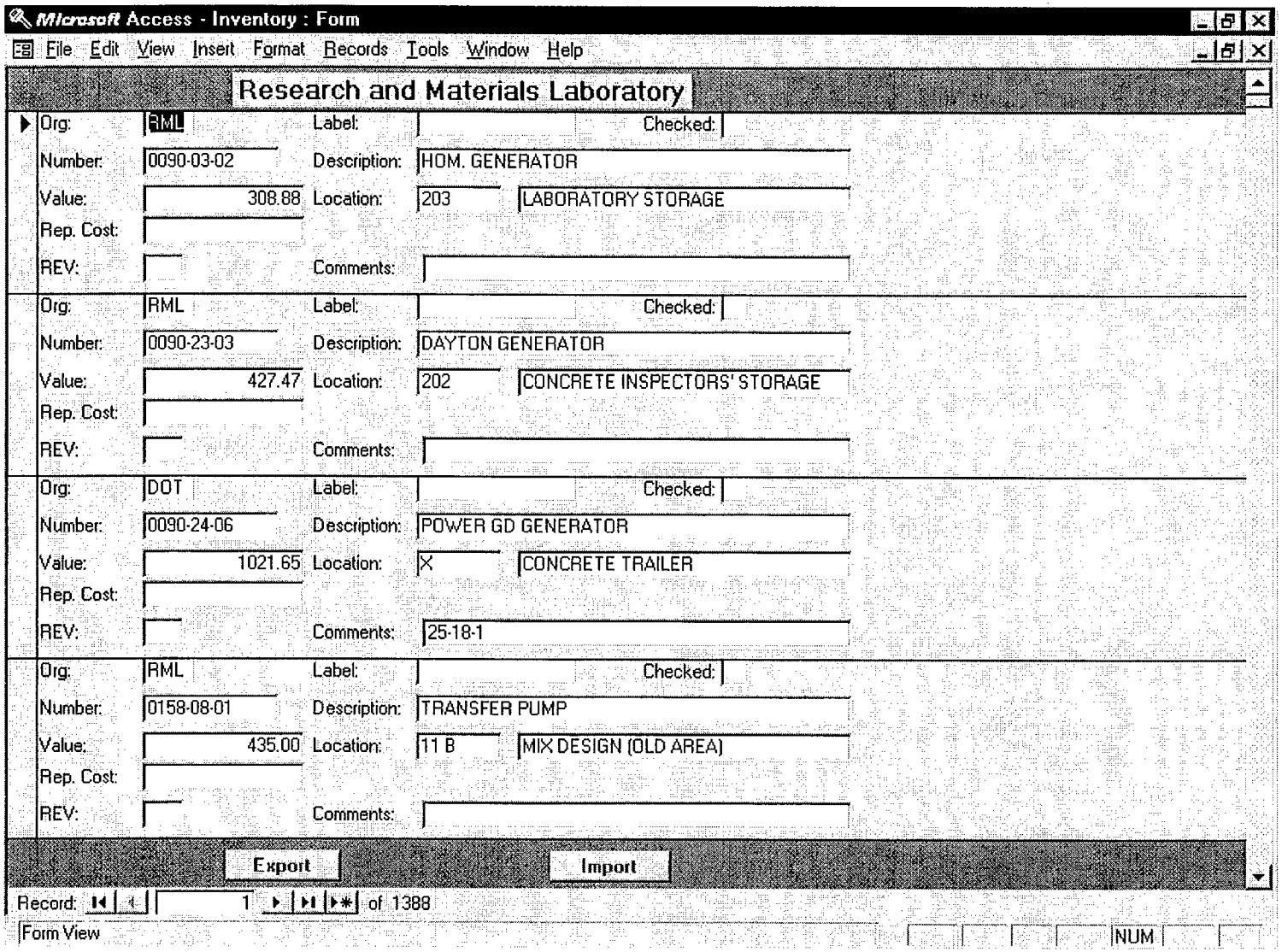

Figure 5. Access Data Screen

File Type	Name	Description
Table	Inventory Backup	Used as a temporary table to store data during the "import" process.
Table	Inventory Imported	Used as a temporary table to store data imported from an external file.
Table	Inventory	The table used to store all information regarding the assets.
Query	Checked & Wrong by DOT Number	Creates a query of checked and unchecked items sorted by DOT codes.
Query	Checked by Room	Creates a query of checked items grouped by locations.
Query	Room	Creates a query sorted by room number.
Query	Unchecked	Creates a query of unchecked items.
Query	Label	Creates a query sorted by the bar code label.
Report	Checked & Wrong by DOT Number	Report utilizing the query Checked & Wrong.
Report	Checked by Room	Report utilizing the Checked by Room query.
Report	Room	Report utilizing the query Room.
Report	Unchecked	Report utilizing the query of unchecked items.
Report	Label	Pre-formatted report that utilizes the query Label.
Form	Inventory	The main screen. It is utilized for entering data and retrieving information activities.
Macros	Export and Import.	Macros that control the database automation.

Table I. Files Contained within RML Inventory System

4. items sorted by DOT/RML number;
5. items sorted by room assigned;
6. items missing from inventory, sorted by DOT/RML number;
7. items sorted by assigned room number
8. items missing or not in proper location, sorted by room number.

Software and Hardware Requirements

The Clemson University Asset Inventory System consists of the compiled Clipper programs Start.exe and Lab.exe with a sample Access data file that is compatible with these programs. The source code for the Start.exe and Lab.exe programs (Appendices A and B) have been provided by Clemson University to facilitate modifications by SCDOT data processing personnel. The system also includes Access reports as described above.

The system requires a PC with Microsoft 95 and a PCMCIA card drive. An Intermec Janus 2020 portable data terminal, with characteristics similar to the Janus unit supplied by Clemson University for the Laboratory inventory pilot (PCMCIA card drive and 2Mb memory), is also required. If labels are to be printed for room locations, a software package such as Labels Unlimited and a laser printer will complete the system components.

System Testing

The final system was demonstrated for RML personnel. Labels were then placed on all property in the file room. Label numbers were linked to existing inventory numbers by entering both label numbers. Then a test was performed to determine how long it would take to inventory the room. An inventory was then made by scanning each label. The inventory of the 24 items was accomplished in about 48 seconds as measured by a RML manager. According to information from

the RML personnel, the same items would be inventoried from 12 to 15 minutes using the manual method. In this sample test it was found that the use of bar code reader was 15 times more faster than the manual process. In that time it is not computed the savings due to keyboarding. The manual process would also require additional time for keyboard data entry and error correction.

After reading the items the memory card was then withdrawn from the portable terminal and inserted in the card drive of a laptop computer. The resulting data base was then visually reviewed to ensure that all data was transferred and correct. The laptop computer used for the system demonstration was not connected into the SCDOT network. Management was in process of upgrading their computers and a computer was not available with both a network card and a memory card drive

Chapter IV

THE SUPPLY DEPOT INVENTORY SYSTEM

System Requirements

The Supply Depot is the central distribution point for commodities distributed to SCDOT district units throughout the state. The Supply Depot 1997 catalog has 1374 items divided under 418 categories. The materials vary from paper towels to tires for trucks. The materials are delivered to the districts on a regular schedule every two weeks. Although the Supply Depot discourages walk-in orders, it occurs with some frequency due to emergency needs.

Meetings were held with Supply Depot personnel to establish a set of overall system requirements. Management's concept was to use the system in the shipping department to ensure that shipping forms matched the actual commodities being shipped. This way the bar code would read a commodity and identify if it was the one ordered in the invoice. This activity would require that the inventory data base be available in the scanner including item nomenclature, SCDOT SWIP number, and Depot Stock number. This application did not require any printed reports or the storage of any files for future reference. This application would require each item, regardless of size, to have an attached bar code.

A decision was made to develop a broader application that would provide the central core of a comprehensive inventory system and provide the needed support in the shipping department. A basic inventory control system could be used to reduce time required to complete the annual inventory. It could then be used as the central core for a complete in-house inventory system, if needed, using the same technology for checking in new commodities and decrementing the

inventory as commodities are shipped. This approach would give Depot management a more flexible system that could satisfy future requirements. System parameters developed during the meetings included the following:

1. The system will utilize the SCDOT Janus 2020 portable data terminal manufactured by Intermec.
2. The data base will be based on Microsoft Access Software.
3. A simple method must be available for transferring data between the portable terminal and a computer on the SCDOT network.
4. Depot personnel would continue to print human readable characters in the labels along with the bar coding.
5. Labels of two dimensions should be used: a small label for individual item labels, and a large label for shelf edges or fronts of bins.
6. The system must be capable of reducing time required for data collection and reconciliation.

Data transfer between the portable terminal and computer will be accomplished by physically transferring the memory card between the two units. Data could be transferred by wire but it would require additional software and a knowledge of computer operations.

Proposed Inventory Method

By using the scanner in the inventory process, a much simpler and quicker inventory method could be used to reduce the time and effort required using the conventional method.

A sequential inventory could be made in the Depot by inventorying commodities as they are arranged on the shelving without regard to the master inventory list. For example, the inventory would start with the commodity in the upper left corner of the first shelving section and continue sequentially shelf by shelf until the lower right hand corner of the section. This process could be

repeated for every shelving unit and bulk storage area in the Depot. The inventory software is designed to add partial totals as data is entered. For example, data on one shelved item is entered and at some time mater the overstock of that commodity is found elsewhere in the Depot. The additional items are entered as new inventory items but the software will add the quantities together. Using this approach the primary concern is ensuring that all commodities along each aisle are included in the inventory. This approach significantly reduces the probability of missing overstocked items and the time required to complete the inventory. Upon completion, the software will provide a complete inventory report of the Depot and report uninventoried items that appears on the master list but not included in the current report.

System Development

Supply Depot Inventory Data File

To execute this pilot project, SCDOT data processing personnel provided Clemson University with a sample ACCESS data file or table consisting of 60 inventory items. This data file was created as a subset of the overall mainframe file that is currently being used to manage the inventory. Clemson University modified the data file such that it now contains six fields as shown with sample data in Figure 6. The Access file contains the item description, a Supply Depot number, an SCDOT SWIP number, inventory quantity, unit of measure, and the date of the last inventory inspection.

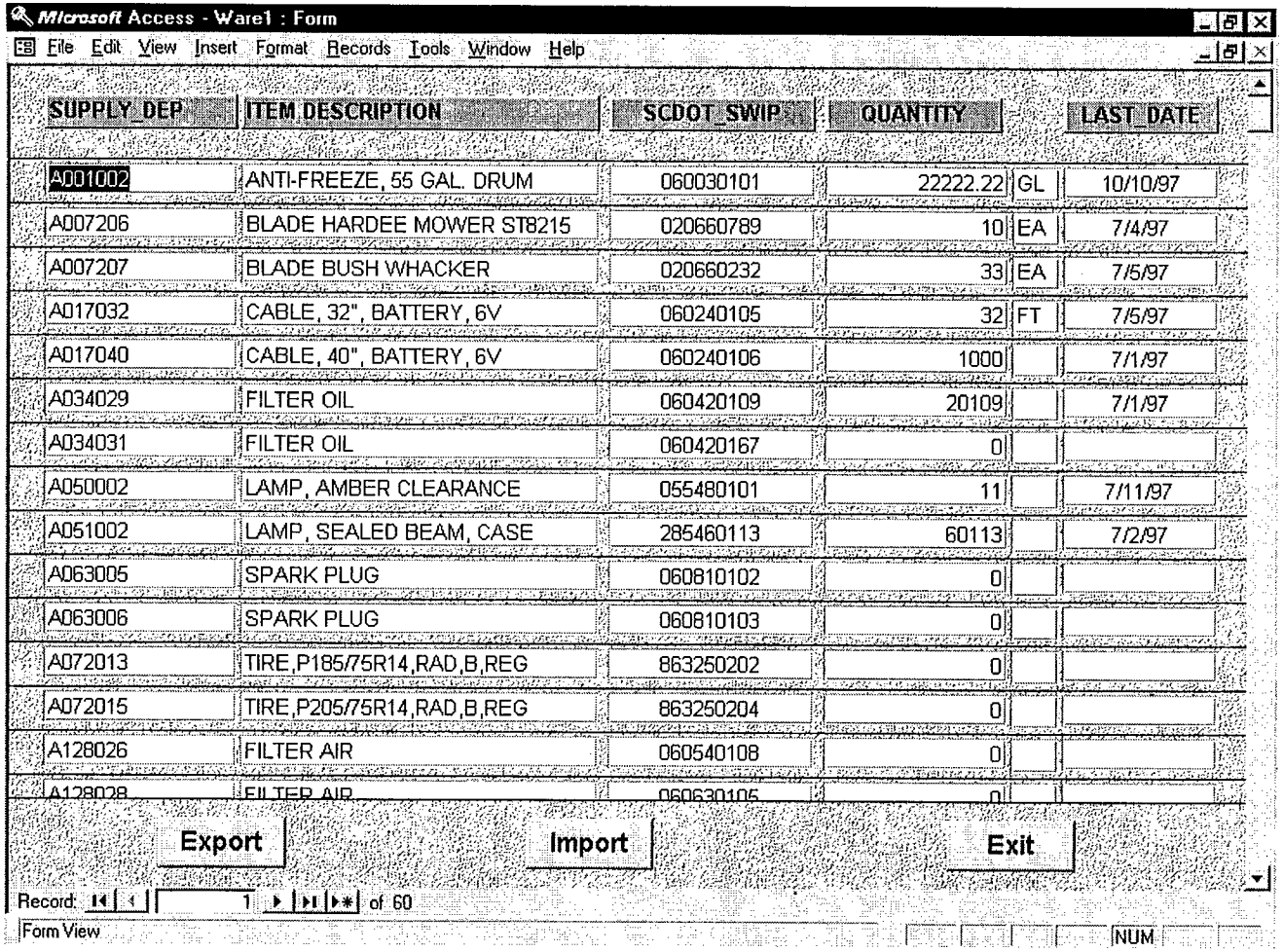

Figure 6. Access Data File

The Access Database: Warehouse.mdb

The Access database, Warehouse.mdb, was developed by Clemson University to manipulate the inventory data file and provide import and export capabilities with the Janus unit, which is used to capture the inventory data. The database also provides mechanisms for editing data, and generating basic reports.

The database opening screen is shown in Figure 7. The three buttons provide options as follows:

1. edit Database to modify the file and import/export data to/from the Janus unit;
2. update File to run a macro that will update the data file for recently scanned data; and
3. exit to exit the database.

Selecting the Edit Database option displays the screen shown in Figure 7. File export is accomplished through the Export button shown in Figure 6. Importing files from the Janus unit is a two step process. Activating the Import button of the editing screen, Figure 6 converts the dBASE file used in the Janus unit to Access format. The user then returns to the opening screen, Figure 7, to select the Update button to merge recently scanned data into the Access file. Pressing the Exit button on the opening screen will close all databases and minimize Warehouse.mdb.

The system, as provided, currently consists of the three standard reports that are illustrated in Figure 8. However system users familiar with Access can generate system reports in any format as required. These reports print the data file inventory by SWIP code, by Depot number, and alphabetically by description. The Access database system also consists of macros that control the database, two screen format forms, and two Access data files or tables. The primary data table is titled Ware1, and the secondary table that is used as an intermediate step during the data import process is titled ImportedWare.

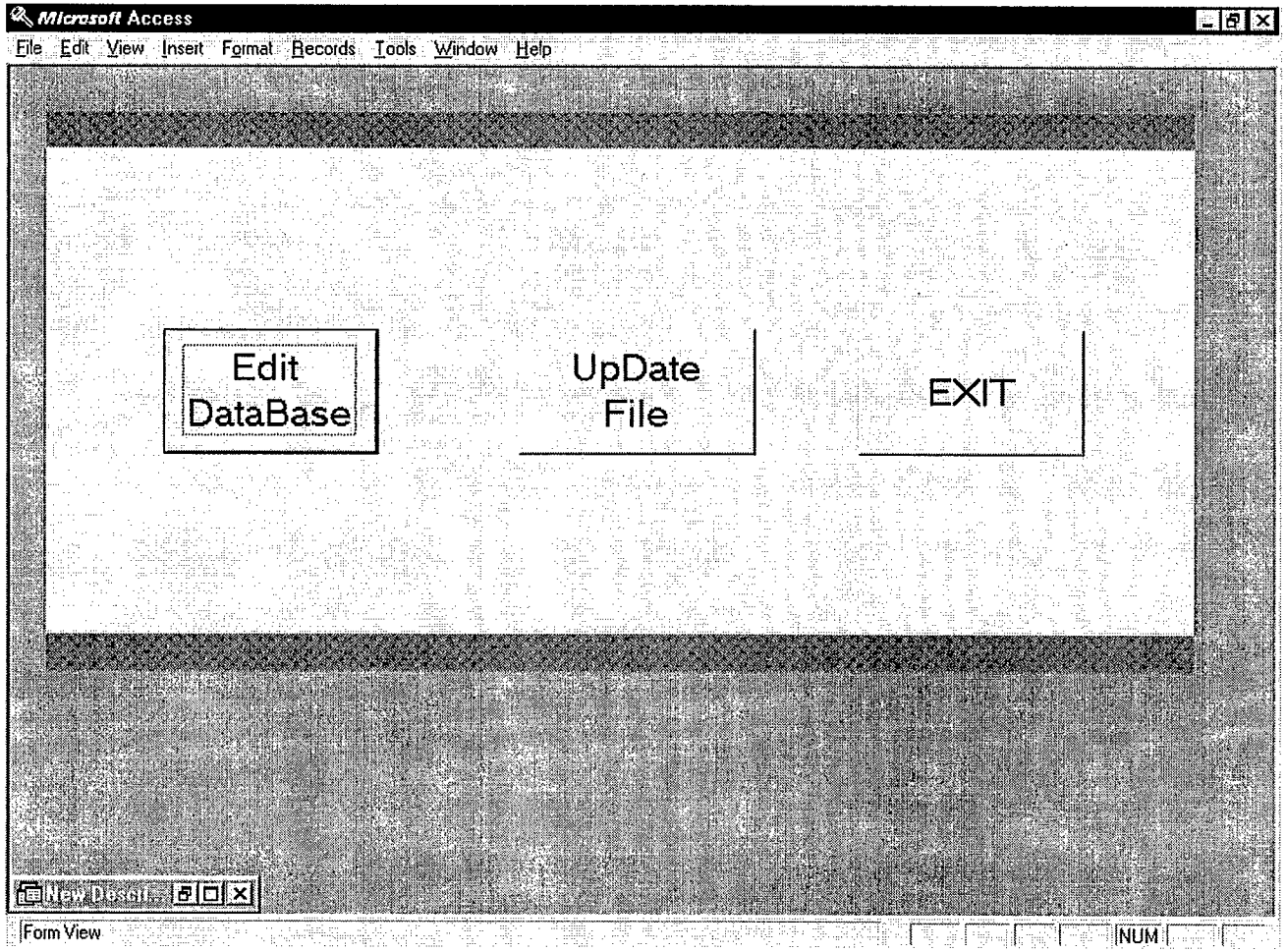

Figure 7. Access Opening Screen

Swip Code Report

SCDOT_SW	*ITEM_DESCR*	*SUPPLY_D*	*QUANT*	*UNIT*	*ST*
020660232	BLADE BUSH WHAC	A007207	0	EA	
020660789	BLADE HARDEE MO	A007206	0	EA	
055480101	LAMP, AMBER CLEA	A050002	0		
060030101	ANTI-FREEZE, 55 G	A001002	0	GL	
060240105	CABLE, 32", BATTER	A017032	0	FT	
060240106	CABLE, 40", BATTER	A017040	0		
060420109	FILTER OIL	A034029	0		
060420167	FILTER OIL	A034031	0		
060540108	FILTER AIR	A128026	0		
060630105	FILTER AIR	A128028	0		
060810102	SPARK PLUG	A063005	0		
060810103	SPARK PLUG	A063006	0		
085400101	BAG, SAMPLE, SMA	G003001	0		
085400102	BAG, SAMPLE, LAR	G003002	0		
200450109	RAIN SUIT, LARGE	G073003	0		
200450110	RAIN SUIT, EXTRA L	G073004	0		
225350101	COOLER, WATER, 2	G046002	0		
225350102	COOLER, WATER, 5	G046005	0		
280240101	WIRE, WHITE, #8	E040008	0		
280240102	WIRE, BLACK, #8	E040009	0		
280300103	CABLE TRAFFIC SIG	S005104	0		
280300105	CABLE TRAFFIC SIG	S005103	0		
280800101	CABLE, #12-2, 1000'	E005013	0		
280800102	CABLE #12AWG, 100	E005014	0		
285230108	FITTING,1",SER.ENT	E016100	0		
285230109	FITTINGS,2",SER.EN	E016200	0		
285460113	LAMP, SEALED BEA	A051002	0		

Figure 8. SWIP Code Report

Supply Depot personnel will exchange information with data processing personnel through the file Warehouse.mdb, which contains the table ware1. A description of the files contained within Warehouse.mdb appears in Table II. Therefore it is important that the ware1 fields sizes, type, and order be structured as shown in Figure 9.

Bar Code Data Entry Using the Janus Unit

The Depot warehouse inventory is performed by scanning bar code labels using the fully programmable Janus unit, entering observed quantities, and then transferring a data file to the Access system. Residing within the memory of the Janus unit is a data file in dBASE format and a compiled Clipper program named WareH.exe. The source code for the program WareH.exe is included as Appendix C. As noted above, format conversions between the Janus dBASE data file (Ware.dbf) and the Access files or tables (Ware1 and ImportedWare) are performed within the Access database system. The structure of the dBASE file Ware.dbf is shown in Figure 9.

The Janus unit accepts DOS commands to run executable programs and perform other functions. Figure 10 shows the process flowchart. At the DOS prompt the inventory program is executed by entering the command:

wareh I

Where the "I" option is used to zero all quantities in the data file prior to beginning an inventory session. The user then enters the current date. The inventory proceeds by scanning an item bar code label (SWIP or Supply Depot number) after which the Janus unit displays the item codes, item description, quantity, and previous inventory date. The inventory quantity is then entered using the Janus unit keypad. The Janus unit display screen is illustrated in Figure 11.

File Type	Name	Description
Table	ImportWare	Used as a temporary table to store data imported from an external file.
Table	Ware1	The table used to store all information regarding the commodities.
Query	Depot Number Query	Creates a query sorted by the Supply Depot control number.
Query	Description	Creates a query sorted by the description in alphabetical order.
Query	Swip Code Query	Creates a query sorted by the SCDOT SWIP code.
Report	Depot Number Report	Pre-formatted report that utilizes the query Depot Number Query.
Report	Item Description	Pre-formatted report that utilizes the query Description.
Report	Swip Code Report	Pre-formatted report that utilizes the query Swip Code Query.
Form	Control Form	It is the opening screen of this Access database.
Form	Ware1	The main screen. It is utilized for entering data and retrieving information activities.
Macros	Autoexec, EditFor, Exit, Export, Import, Maximize, UpDate File	Macros that control the database automation. These macros are automatically executed every time that the Warehouse.mdb is loaded.

Table II. Files Contained within the Supply Depot Inventory System

	Field	Type	Width	Decimal Places
1	*SUPPLY_DEP*	Character	7	
2	*ITEM_DESCR*	Character	25	
3	*SCDOT_SWIP*	Numeric	9	
4	*QUANT*	Numeric	19	5
5	*UNIT*	Character	2	
6	*LAST_DATE*	Character	8	

Figure 9. Structure of Janus Unit dBASE File Ware.dbf

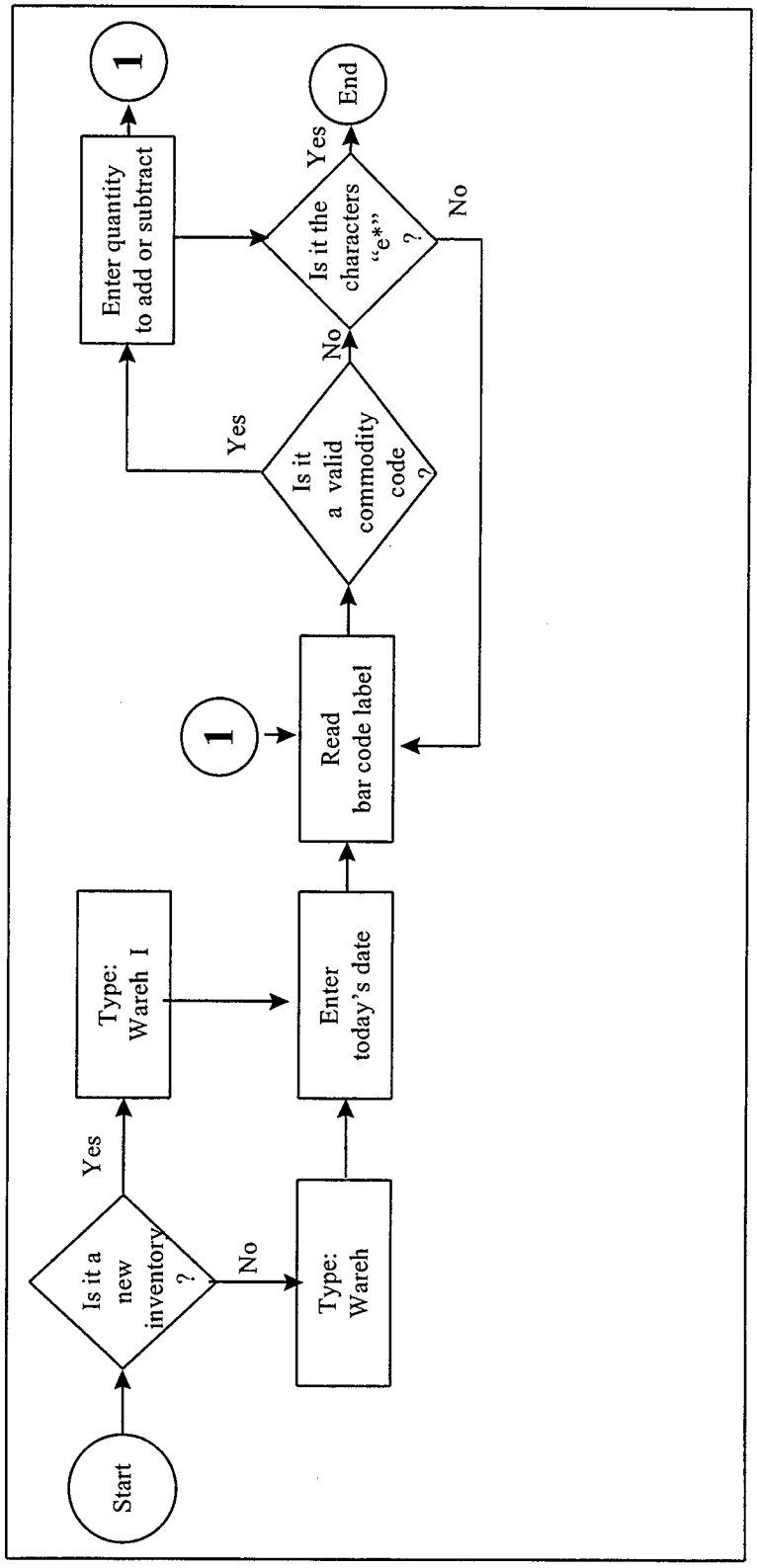

Figure 10. Using Wareh.exe

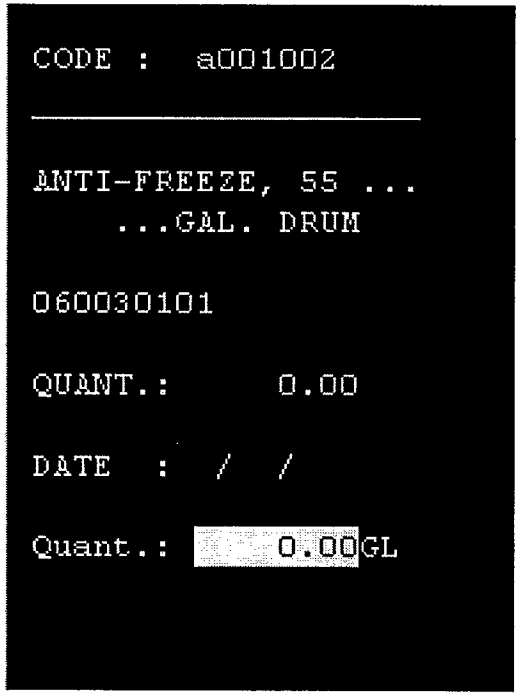

Figure 11. Janus Unit Display Screen

To control program execution, the user can enter optional codes at the prompt to enter the material code. E* will end the inventory session, U* will undo the last data entry, and D* will permit changing the current date.

Bar Code labels

The inventory process utilizes bar code labels that will be attached to items, boxes, shelves, pallet racks or bins. For very large items such as tires the label may be placed on a wall adjacent to the storage location. The bar code labels are printed with the unique SWIP or Supply Depot item code using a bar code printing software package. The software package Labels Unlimited was found to be satisfactory. Labels of any size can be printed, facilitating scanning of the label from a distance of three feet. There are two pre-formatted samples of labels in the system. One is named Labels.job and shall be used for small labels (Avery Pin 4013 - 0.9x3.5" Address), and the other, Bigcode.job, shall be used for big labels (Avery Pin 4166 - 3x5" Index Card).

Hardware and Software Requirements

The Clemson University Depot Inventory System consists of a compiled Clipper program WareH.exe with sample Access data files and routines that permit importing and exporting files to the Janus unit and generating basic reports. The source code for the WareH program has been provided to facilitate modifications by SCDOT data processing personnel if desired. The Access database can also be easily modified to incorporate additional capabilities.

The system requires a PC with Microsoft Windows 95 and a PCMCIA card drive. An Intermec Janus 2020 portable data terminal with PCMCIA card drive and 2Mb memory is also required. Labels can be printed using the Labels Unlimited software and a standard laser printer.

System Testing

The final system was demonstrated using a limited set of bar-coded labels printed at the Depot. A test was performed to determine how long it would take to inventory a representative sample of items. The sample included several sections of four level shelving for palletized storage, and more than a section of five level shelving for small parts in plastic bins. The labels were attached to either the edge of the shelf under the commodity or on the front of the plastic bin. The inventory required the use of a rolling ladder to enable the person making the inventory to read the labels and physically count the items present. Using the sequentially approach, the inventory required 15 minutes to inventory 31 commodities on open shelving and 32 minutes to inventory 29 bins of small parts. A survey was made of the warehouse shelving and about 50 shelves of plastic bins and 213 shelves of storing pallets were found. Based on these results, it is estimated that all items within the Depot could be inventoried by one person using the portable bar code scanner in less than 14 hours. It is a significant difference when compared to four two-men teams working for a week using their regular inventory process.

The shipping department application was not separately tested because of the ability of the Janus unit to display information from the data base was adequately tested during the inventory test program discussed above. In an informal activity, depot personnel became familiar with the Janus unit by scanning a sheet of bar-coded labels and reading the resulting display screen; the same process to be used in the shipping activity. In each case the Janus unit displayed the item nomenclature, depot stock number, and the quantity of the item in the inventory. The benefit from this application will be increased accuracy of the items being shipped and should not have a significant impact on operations in the Depot.

Chapter V

CONCLUSIONS

The objective of the research described herein was to demonstrate the use of bar code technology as a data acquisition tool in inventory operations. Two inventory software packages were developed with system characteristics specified by SCDOT. Both systems interact with a Microsoft Access data base, thus providing the ability to define a wide range of reports without further programming.

The RML system, when fully installed, will significantly reduce the time required to complete the annual inventory. The inventory data can be captured using a hand held scanner and transferred to a desk top computer without rekeying the data. A test indicated that the inventory process using bar code technology is faster than the manual process. The test result was 15 times faster using bar code scanner.

The Supply Depot system, when fully installed, will significantly reduce time required to complete the annual inventory and reduce the time that the Depot must be closed. It is estimated that whereas the manual inventory process required approximately 288 manhours, the automated process using the bar code scanner will require approximately 14 manhours. The system could serve as the core of a comprehensive inventory management system or complement other systems related to reordering receipt of new commodities and decrementing the inventory as commodities are shipped.

The project has demonstrated the value of using bar coding for inventory. The RML inventory system can easily be adopted at other SCDOT facilities.

Appendix A

Start.exe Source Code: Lab Initialization Program

```
/*
PROGRAM: Research Lab Inventory
File Name: Start.prg
Main Module
*/

#define ONE 1
#define TWO 2
#define THREE 3
#define FOUR 4
#define FIVE 5
#define tTAB 9
#define tESC 27
#define DTA CTOD("01/01/97")
#define COR Color "BG+/N,GR+/B+"

#define tUP 5
#define tDOWN 24
#define tLEFT 19
#define tRIGHT 4
#define tPGDOWN 3
#define tPGUP 18
#define tEND 6
#define tHOME 1
#define tINS 22
#define tDEL 7

Procedure  Main(FILE)

Local x:=1, Result:=0
Public  vCODE:=space(8), TItems:=0, ;
  vRev:=0,vComments:=space(22),vRoom:=space(36),;
  vCheck:="U", vLabel:=0

clear
SET EXACT ON
SET WRAP ON
Use (FILE) ALIAS DB New
Index on Number to "Number"
Reindex
Index on LOC to "LOC"
Reindex
Set index to Number, Loc
Set order to 1
clear
```

```
@ 1,1

DO while .T.
  vCODE:=space(10)
  /* vCODE -> is the inventory number, either RML or DOT */
  @ 1,1 Say "Inv.:" get vCODE COR
  Read
  Result:=0
  vCode:=Upper(vCode)
  DO CASE
       Case left(vCode,1)="E"
            CLOSE ALL
            RETURN
       Otherwise
            set order to 1
            seek vCode
            Result:=IF(Found(),1,0)
            If (Result=0 .or. DB->Label>0)
                 vCode:=space(8)
                 loop
            endif
            vLabel:=0
            @ 1,1 Say "Inv.: " + vCODE COR
            @ 3,1 Say "Label:" get vLabel PICTURE "99999999" COR
            @ 5,1 Say "_____"
            read
            If vLabel=0
                 loop
            endif
            DB->Label:=vLabel
  Endcase
ENDDO
RETURN

/*********************************/
```

Appendix B

Lab.exe Source Code: Lab Inventory Program

```
/*
PROGRAM: Research Lab Inventory
File Name: Lab.prg
Main Module
*/

#define ONE 1
#define TWO 2
#define THREE 3
#define FOUR 4
#define FIVE 5
#define tTAB 9
#define tESC 27
#define DTA CTOD("01/01/97")
#define COR Color "BG+/N,GR+/B+"

#define tUP 5
#define tDOWN 24
#define tLEFT 19
#define tRIGHT 4
#define tPGDOWN 3
#define tPGUP 18
#define tEND 6
#define tHOME 1
#define tINS 22
#define tDEL 7

Procedure  Main(FILE,ACTION)

Local x:=1, Result:=0
Public  vCODE:=space(8),TItems:=0,Checked:=0, WChecked:=0, Remaining:=0,;
     vLOC:="     ",vRev:=0,vComments:=space(22),vRoom:=space(36),;
     vCheck:="U"

clear
SET EXACT ON
SET WRAP ON
Use (FILE) ALIAS DB New
Index on LABEL to "Label"
Reindex
Index on LOC to "LOC"
Reindex
Set index to Label, Loc
Set order to 1
If ACTION="I"
```

```
        goto top
        do while .not.EOF()
            Replace DB->CHECK with "U"
            skip
        Enddo
    Endif
    DO while .T.
        vCODE:=space(8)
        @ 1,1 Say "CODE : " get vCODE COR
        Read
        Result:=0
        vCode:=Upper(vCode)
        Do Case
            Case Isalpha(vCODE)
                DO CASE
                    Case left(vCode,1)="R"
                        Result:=RoomCode(vCODE)
                        If Result=0
                            Loop
                        Endif
                        @ 5,1 Say "_____"
                        @ 7,1 Say "Room:"+vLOC+" "+vROOM
                        @ 10,1 Say "Total Items:" + STR(TItems,5,0)
                        @ 11,1 Say "Checked C: " + STR(Checked,5,0)
                        @ 12,1 Say "Checked W: " + STR(WChecked,5,0)
                        @ 13,1 Say "Remaining: " + STR((TItems-Checked-
WChecked),5,0)
                        loop
                    Case left(vCode,1)="S"
                        /* Result:=StatusCode(vCODE) */
                        loop
                    Case left(vCode,1)="E"
                        CLOSE ALL
                        RETURN
                ENDCASE
            Otherwise
                vLabel:=val(vCODE)
                set order to 1
                seek vLabel
                Result:=IF(Found(),1,0)
                If (Result=0 .or. vLoc=" ")
                    vCode:=space(8)
                    loop
                endif
                @ 3,1 Say "Inv.:" + DB->NUMBER
```

```
                IF (DB->LOC=vLOC)
                    Do Case
                        Case DB->CHECK="W"
                            Checked=Checked+1
                            WChecked=WChecked-1
                        Case DB->CHECK="U"
                            Checked=Checked+1
                        Case DB->Check="C"
                        OTHERWISE
                            Checked=Checked+1
                    Endcase
                    Replace DB->CHECK with "C"
                Elseif (Upper(DB->CHECK)="C" .or. Upper(DB->CHECK)="W")
                    @ 3,1 say "*Already Checked*"
                    @ 4,1 say "* in other room *"
                    WAIT " "
                ElseIf (Upper(DB->CHECK)="U")
                    @ 3,1 Say "** WRONG PLACE **"
                    @ 4,1 Say "C / W / U"
                    WAIT "Press choice..." to vCheck
                    Do Case
                        Case upper(vCHECK)="W"
                            Replace DB->CHECK with "W"
                        Case upper(vCHECK)="U"
                            Replace DB->CHECK with "U"
                        OTHERWISE
                            Replace DB->CHECK with "C"
                            Replace DB->ROOM with Upper(vROOM)
                            Replace DB->LOC with Upper(vLOC)
                            Checked:=Checked+1
                            TItems:=TItems+1
                    Endcase
                Endif
                CLEAR
                @ 1,1 Say "CODE :  " + vCODE COR
                @ 3,1 Say "Inv.:" + DB->NUMBER
                @ 5,1 Say "_____"
                @ 7,1 Say "Room:"+vLOC+" "+vROOM
                @ 10,1 Say "Total Items:" + STR(TItems,5)
                @ 11,1 Say "Checked C: " + STR(Checked,5)
                @ 12,1 Say "Checked W: " + STR(WChecked,5)
                @ 13,1 Say "Remaining: " + STR((TItems-Checked-WChecked),5,0)
    Endcase
ENDDO
RETURN
```

```
/**********************************/

Function RoomCode(CODE)
    Local x:=0
    x:=Len(CODE)
    vLOC:=Right(CODE,(x-1))
    vLOC:=Alltrim(vLOC)
    TItems:=0
    Checked:=0
    WChecked:=0
    Set order to 2
    goto top
    seek vLOC
    If FOUND()
        x:=1
        vROOM:=DB->ROOM
    Endif
    count to TItems WHILE DB->LOC=vLOC
    seek vLOC
    count to Checked WHILE DB->LOC=vLOC for upper(DB->Check)="C"
    seek vLOC
    count to WChecked WHILE DB->LOC=vLOC for upper(DB->Check)="W"
Return (x)
```

Appendix C

Wareh.exe Source Code: Depot Inventory Program

```
/*
PROGRAM: Supply Depot Inventory - Warehouse
File Name: WareH.prg
Main Module
*/

#define ONE 1
#define TWO 2
#define THREE 3
#define FOUR 4
#define FIVE 5
#define tTAB 9
#define tESC 27
#define DTA CTOD("07/01/97")
#define COR Color "BG+/N,GR+/B+"

/*
#define tUP 5
#define tDOWN 24
#define tLEFT 19
#define tRIGHT 4
#define tPGDOWN 3
#define tPGUP 18
*/
#define tEND 6
#define tHOME 1
#define tINS 22
#define tDEL 7


Procedure  Main(ACTION)

Local Result:=0
Public  vSupply:=space(7), vItem:=space(25), vSwip:=space(9), vQuant:=0.00000,;
     vLDate:=CTOD("07/01/97"), vCode:=space(100), x:=1

clear
SET EXACT ON
SET WRAP ON
Use "WARE" New
Index on SUPPLY_DEP to WSUPPLY
Index on SCDOT_SWIP to WSWIP
Set index to WSUPPLY, WSWIP
Set order to 1
Copy Structure To "WWWTEMPO"
```

```
Use "WWWTEMPO" ALIAS TEMPO NEW
SELECT WARE


/*
     I=> Initialize.
     When running the program with this option the quantity of items stored
     on field QUANT will be zero (0). This option should be used if a new inventory is
     ready to begin.
*/

If ACTION="I"
     goto top
     do while .not.EOF()
          Replace WARE->QUANT with 0
          skip
     Enddo
Endif
CLEAR
@ 1,1 Say "TODAY'S DATE"
@ 2,1 get vLDate
READ
DO while .T.
     CLEAR
     vCODE:=space(10)
     @ 1,1 Say "CODE : " get vCODE COR
     Read
     Result:=0
     vQuant:=0.00000
     vCode:=Rtrim(Upper(vCode))
     IF (ISALPHA(vCode))
          DO CASE
               Case left(vCode,2)="E*"
                    CLOSE ALL
                    RETURN
               Case left(vCode,2)="U*"
                    If (x=1)
                         loop
                    Endif
                    UndoLast()
               Case left(vCode,2)="D*"
                    CLEAR
                    @ 1,1 Say "TODAY'S DATE"
                    @ 2,1 get vLDate
                    READ
```

```
                loop
            OTHERWISE
                set order to 1
                seek vCode
                result:=IF(Found(),1,0)
                if (result=0)
                    loop
                endif
                @ 2,1 Say "_____"
                @ 4,1 Say left(WARE->ITEM_DESCR,16)+"..."
                @ 5,5 Say "..."+right(WARE->ITEM_DESCR,9)
                @ 7,1 Say WARE->SCDOT_SWIP
                @ 9,1 Say "QUANT.:" + STR(WARE->QUANT,9,2)
                @ 11,1 Say "DATE :" + DTOC(WARE->LAST_DATE)
                @ 13,1 Say "Quant.:" get vQuant picture "99999.99"
                @ 13,17 Say WARE->UNIT
                READ
                If !(Updated())
                    loop
                endif
                Update_File()
        ENDCASE
    ELSE
        set order to 2
        seek vCode
        result:=IF(Found(),1,0)
        if (result=0)
            loop
        endif
        @ 2,1 Say "_____"
        @ 4,1 Say left(WARE->ITEM_DESCR,16)+"..."
        @ 5,5 Say "..."+right(WARE->ITEM_DESCR,9)
        @ 7,1 Say WARE->SUPPLY_DEP
        @ 9,1 Say "QUANT.:" + STR(WARE->QUANT,9,2)
        @ 11,1 Say "DATE :" + DTOC(WARE->LAST_DATE)
        @ 13,1 Say "Quant.:" get vQuant picture "99999.99"
        @ 13,17 Say WARE->UNIT
        READ
        If !(Updated())
            loop
        endif
        Update_File()
        set order to 1
    Endif
ENDDO
```

```
RETURN

Function Update_File()
    Select TEMPO
    DELETE ALL
    PACK
    Append Blank
    Replace TEMPO->SUPPLY_DEP WITH WARE->SUPPLY_DEP
    Replace TEMPO->ITEM_DESCR WITH WARE->ITEM_DESCR
    Replace TEMPO->SCDOT_SWIP WITH WARE->SCDOT_SWIP
    Replace TEMPO->QUANT WITH WARE->QUANT
    Replace TEMPO->UNIT WITH WARE->UNIT
    Replace TEMPO->LAST_DATE WITH WARE->LAST_DATE
    Select WARE
    vQuant:=vQuant+Ware->QUANT
    Replace WARE->QUANT with vQUANT
    Replace WARE->LAST_DATE with vLDate
    x:=0
Return

Function UndoLast()
    Select TEMPO
    GO 1
    Select WARE
    Seek TEMPO->SUPPLY_DEP
    Replace WARE->QUANT WITH TEMPO->QUANT
    Replace WARE->LAST_DATE WITH TEMPO->LAST_DATE
    x:=1
Return
```