A RELATIONAL DATABASE MANAGEMENT SYSTEM FOR A ROK ARMY INFANTRY DIVISION WITH PROBABILISTIC INVENTORY CONTROL MODEL

by

Taeyong Park

December 1987

Thesis Advisor Y. K. Mortagy

Approved for public release; distribution is unlimited.
A RELATIONAL DATABASE MANAGEMENT SYSTEM FOR A ROK ARMY INFANTRY DIVISION WITH PROBABILISTIC INVENTORY CONTROL MODEL

The purpose of this thesis is to provide an inventory database management system for a ROK Army Infantry Division.

Since, 1973, the ROK Army has significantly improved its capability in logistics management by using computer systems. However, the operational level of command (the division) has some difficulties in meeting higher command requirements because of the unavailability of computer hardware and software. As of last year, the computer hardware was installed at the division level. User friendly software development has become another requirement for effective use of this computer hardware. To help meet this objective, this thesis provides a database management system for the manager who works at the division logistics section, and for the Quartermaster battalion which is the major unit to handle material within the division.
To meet the objective, the author of this thesis concentrated on writing user friendly interface programs within the current logistics management system.

By applying the proposed system, the Army can improve one of its logistics objectives, i.e., automatic data processing. This can contribute to the logistics management system implementation by allowing the division to generate more accurate reports in less time, and to improve inventory management by shortening the administrative process.
A Relational Database Management System
for a ROK Army Infantry Division
with Probabilistic Inventory Control Model

by

Taeyong Park
Major, Republic of Korea Army
B.S., Korea Military Academy, 1978

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN MANAGEMENT

from the

NAVAL POSTGRADUATE SCHOOL
December 1987

Author: Taeyong Park

Approved by: Y.K. Mortagy
Y.K. Mortagy, Thesis Advisor

Thomas P. Moore, Second Reader

David R. Whipple, Chairman,
Department of Administrative Science

James M. Fremgen,
Acting Dean of Information and Policy Science
The purpose of this thesis is to provide an inventory database management system for a ROK Army infantry division.

Since 1973, the ROK Army has significantly improved its capability in logistics management by using computer systems. However, the operational level of command (the division) has some difficulties in meeting higher command requirements because of the unavailability of computer hardware and software. As of last year, the computer hardware was installed at the division level. User friendly software development has become another requirement for effective use of this computer hardware. To help meet this objective, this thesis provides a database management system for the manager who works at the division logistics section, and for the Quartermaster battalion which is the major unit to handle material within the division.

To meet the objective, the author of this thesis concentrated on writing user friendly interface programs within the current logistics management system.

By applying the proposed system, the Army can improve one of its logistics objective, ie automatic data processing. This can contribute to the logistics management system implementation by allowing the division to generate more accurate reports in less time, and to improve inventory management by shortening the administrative process.
# TABLE OF CONTENTS

## I. INTRODUCTION ........................................... 13
A. INTRODUCTION ........................................... 13
B. THESIS ORGANIZATION .................................. 14

## II. ARMY LOGISTICS STRUCTURE AND ADPS IN THE KOREA ARMY ................................. 15
A. ARMY LOGISTICS STRUCTURE .................................. 15
   1. Overview ........................................... 15
   2. The Army Logistics Unit Organization .................. 15
   3. Management Issues .................................. 16
B. SELECTIVE MANAGEMENT .................................. 17
   1. Overview ........................................... 17
   2. The ABC Classification system ...................... 17
C. THE EXISTING SYSTEM .................................. 19
   1. The Planning-Programming-Executing-Evaluating System (PPBEES) ................................ 19
   2. Resource flow in the infantry division ................ 20
   3. Inventory operation and control in divisions ........ 21
   4. Current documentation and report forms .............. 26
D. CURRENT USE OF COMPUTERS FOR LOGISTICS IN ROK ARMY ................................. 27
   1. Overview ........................................... 27
   2. ADP Support to logistics ............................ 28
E. SUMMARY ........................................... 29

## III. RELATIONAL DATABASE MODELS ........................................... 30
A. INTRODUCTION ........................................... 30
B. WHAT IS DATABASE? ...................................... 30
C. RELATIONAL DATABASE MODEL ............................ 32
   1. Relation ........................................... 32
2. Keys .......................................................... 33
3. Record relationships ..................................... 33
4. Database Management System ....................... 33

IV. THE PROPOSED SYSTEM .................................................. 35
A. OVERVIEW ....................................................... 35
B. MANUAL SYSTEM ............................................... 35
C. THE PROPOSED SYSTEM STRUCTURE .................. 35
D. THE PROPOSED SYSTEM FUNCTIONS .................. 36
   1. Transaction tracking .................................... 36
   2. Report Generation ...................................... 37
   3. Limited analysis ....................................... 39
   4. Use friendly interface .................................. 40
E. SYSTEM LIMITATIONS ........................................... 41
   1. Demand forecasting .................................... 41
   2. Statistical analysis of transactions ............... 41
   3. ASL request ............................................. 41
   4. Fund Accounting ....................................... 41
F. ADVANTAGES OF THE SYSTEM .............................. 42

V. CONCLUSIONS ............................................................ 43
A. CONCLUSIONS .................................................... 43
B. FURTHER STUDY .................................................. 43

APPENDIX A: STATISTICAL CONSIDERATION ............... 45

APPENDIX B: DATA STRUCTURE ............................... 49
1. OVERVIEW ..................................................... 49
2. DATA FILES ..................................................... 49
   a. Master .................................................... 49
   b. Property file ............................................ 50
   c. ASL file .................................................. 50
   d. Customer file .......................................... 51
   e. Stockout file .......................................... 51
   f. Batch file .............................................. 51
   g. Normal file ............................................ 51
APPENDIX C: APPENDIX SYSTEM STRUCTURE
1. OVERVIEW ........................................ 54
2. PMMAIN ........................................... 54
3. MANAGFL ......................................... 54
4. TRANSACT ........................................ 56
   a. StockRD ........................................ 56
   b. TRbatch and TRissue ............................ 56
   c. TRreqlst ........................................ 58
   d. TRstock ......................................... 58
   e. TURTOLC, TRTURNIN and TRTUNLST ............ 59
   f. Query ........................................... 59
5. REPORTS ......................................... 65
6. ANALYSIS ......................................... 69

APPENDIX D: USER'S MANUAL .................................. 70
1. INTRODUCTION .................................... 70
   a. Requirements .................................... 70
   b. Organization ..................................... 70
2. GETTING STARTED ................................ 70
   a. Configuration .................................... 71
   b. Summary of the proposed system disk .......... 71
3. WORKING WITH THE PROPOSED SYSTEM ........... 73
   a. Routine transaction .............................. 73
   b. File management ................................ 76
   c. Reports generation .............................. 83
   d. Analysis ....................................... 84
   e. Query .......................................... 87

APPENDIX E: PROGRAM LISTING ............................ 89
1. MAIN MENU ........................................ 89
   a. PMAIN ......................................... 89
   b. PMMENU ....................................... 91
2. ROUTINE TRANSACTION ............................. 93
a. TRANSACTION MENU ........................................ 93
b. TRSTOCKRD ........................................ 95
c. TRBATCH .............................................. 97
d. TRISSUE .................................................. 99
e. TRREQLIST ............................................ 102
f. TRSTOCK .............................................. 105
g. TRTURN-IN ............................................ 110
h. TRTUNKLIST .......................................... 111
i. TRCANCEL .............................................. 113
j. TRACKING ............................................... 114

3. REPORTS .................................................. 116
a. REPORTS MENU ....................................... 116
b. RPTSR .................................................... 117
c. RPOST ..................................................... 121
d. RPEIS ..................................................... 125
e. RPSOR ..................................................... 128

4. ANALYSIS ............................................... 132
a. ANALYSIS STARTUP ................................... 132
b. ANLEADT ............................................... 133
c. ANEOQ ................................................... 135
d. ANPROC .................................................. 137

5. MANAGEFL .............................................. 139
a. MANAGEFL MENU .................................... 139
b. ADDASL ................................................. 140
c. ADDPROP ............................................... 142
d. ADDCUST ............................................... 144
e. EDITASL ............................................... 144
f. EDITBAT ............................................... 145
g. EDITCUST ............................................... 146
h. EDITMAST ............................................. 147
i. EDITPROP ............................................... 148
j. MANAGEQ .............................................. 149
APPENDIX F: SCREEN FORMAT .............................................. 151

1. TRANSACT ...................................................... 151
   a. BATCH ............................................... 151
   b. MAST ............................................... 151
   c. TURNIN ............................................. 152
   d. CANCMAST ......................................... 152

2. MANAGEFL .................................................. 153
   a. ASL .................................................. 153
   b. CUST ............................................... 153
   c. PROP ............................................... 154
   d. EDITBAT ........................................... 154
   e. EDITCUST .......................................... 155
   f. EDITASL ........................................... 155
   g. EDITPROP .......................................... 156
   h. EDITCUST .......................................... 156

3. ANALYSIS .................................................... 157
   a. EOQ ............................................... 157

LIST OF REFERENCES ................................................. 158

INITIAL DISTRIBUTION LIST ....................................... 159
LIST OF TABLES

1. BUDGET RATE OF MATERIALS ........................................ 20
2. DEMAND FREQUENCY RATE ........................................... 24
3. DATA STRUCTURE .................................................. 50
4. DATA DICTIONARY ................................................. 52
5. INDEX FILES ...................................................... 53
LIST OF FIGURES

2.1 ABC Classification ........................................... 18
2.2 Division Supply Procedure ...................................... 22
2.3 Army Logistics Computer System ................................. 27
3.1 View of database ............................................. 31
3.2 Relation of property ............................................ 32
3.3 Record relationship ............................................. 34
4.1 Examples of transaction output ................................ 37
4.2 Main data flow diagram ......................................... 38
4.3 Transaction data flow diagram .................................. 38
4.4 Report data flow diagram ....................................... 39
4.5 The output from analysis process ................................ 40
C.1 Detailed system structure ...................................... 55
C.2 Pmmain screen ................................................... 56
C.3 ADDPROP module flowchart .................................... 57
C.4 Example of MANAGEQ module ................................ 58
C.5 Transaction menu screen ........................................ 59
C.6 TRSTOKRD (Request to division) module flow chart ....... 60
C.7 TRBATCH (batch process) module flow chart ................. 61
C.8 TRSTOCK (Issue from LSC) flow chart ......................... 62
C.9 TURN TULC (Turn-in to LSC) module flowchart ............. 63
C.10 Example of Query on transaction module ....................... 64
C.11 Report module menu screen ................................... 65
C.12 RPTSR (transaction status report) module flow chart ...... 66
C.13 RPOST (Order shipping time report) module flow chart ...... 67
C.14 RPSOR (Stockout report) module flow chart ................. 68
C.15 Analysis module screen ........................................ 69
D.1 Main menu(PMAIN) ............................................... 73
D.2 Routine transaction menu screen ................................ 74
<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>D.3</td>
<td>Request for issue from customer screen</td>
</tr>
<tr>
<td>D.4</td>
<td>Batch process examples</td>
</tr>
<tr>
<td>D.5</td>
<td>Request for turn-in from customer</td>
</tr>
<tr>
<td>D.6</td>
<td>Receive stock item from LSC</td>
</tr>
<tr>
<td>D.7</td>
<td>Cancellation of request screen</td>
</tr>
<tr>
<td>D.8</td>
<td>File management screen</td>
</tr>
<tr>
<td>D.9</td>
<td>Customer addition screen</td>
</tr>
<tr>
<td>D.10</td>
<td>Property list addition screen</td>
</tr>
<tr>
<td>D.11</td>
<td>ASL list addition screen</td>
</tr>
<tr>
<td>D.12</td>
<td>Customer record editing screen</td>
</tr>
<tr>
<td>D.13</td>
<td>ASL list record editing screen</td>
</tr>
<tr>
<td>D.14</td>
<td>Master file record editing screen</td>
</tr>
<tr>
<td>D.15</td>
<td>Batch file record editing screen</td>
</tr>
<tr>
<td>D.16</td>
<td>Reports generation menu screen</td>
</tr>
<tr>
<td>D.17</td>
<td>Transaction status report option menu</td>
</tr>
<tr>
<td>D.18</td>
<td>Examples of Transaction status reports</td>
</tr>
<tr>
<td>D.19</td>
<td>Analysis start-up screen</td>
</tr>
<tr>
<td>D.20</td>
<td>Variable entering for analysis</td>
</tr>
<tr>
<td>D.21</td>
<td>Transaction tracking</td>
</tr>
<tr>
<td>D.22</td>
<td>Query on stock number</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Page Numbers</th>
</tr>
</thead>
<tbody>
<tr>
<td>75</td>
</tr>
<tr>
<td>76</td>
</tr>
<tr>
<td>77</td>
</tr>
<tr>
<td>77</td>
</tr>
<tr>
<td>78</td>
</tr>
<tr>
<td>78</td>
</tr>
<tr>
<td>79</td>
</tr>
<tr>
<td>79</td>
</tr>
<tr>
<td>80</td>
</tr>
<tr>
<td>81</td>
</tr>
<tr>
<td>81</td>
</tr>
<tr>
<td>82</td>
</tr>
<tr>
<td>83</td>
</tr>
<tr>
<td>84</td>
</tr>
<tr>
<td>84</td>
</tr>
<tr>
<td>85</td>
</tr>
<tr>
<td>86</td>
</tr>
<tr>
<td>87</td>
</tr>
<tr>
<td>88</td>
</tr>
<tr>
<td>88</td>
</tr>
</tbody>
</table>
I. INTRODUCTION

A. INTRODUCTION

This thesis proposes an inventory control database management system for use at
the ROK Army division.

Supply functions are critical in achievement of economic operations of armed forces, and the management of defense resources. Republic of Korea (ROK) Army logisticians have developed better ways to build support process. One improvement is the recent computerization of Planning-Programming-Budgeting-Executing-Evaluating-System (PPBEES) at high level command (above division).

However, implementation of PPBEES below the division level requires a great deal of manual paperwork, and much overtime for lower level managers.

Fortunately, as of last year, the Army divisions have acquired computer hardware. Still, some usage difficulties remain because of shortages of the computer-related personnel. Thus the development of well structured, user friendly computer systems is urgently needed.

This thesis will introduce a logistics database system which will reduce the amount of paperwork, decrease order processing response time and generate statistics which are now too cumbersome to compute. The system must be well structured and user friendly for easy implementation by novice users.

Therefore this thesis will deal with the application of computer based routine transactions, generation of reports, and analysis of transactions for the ROK Army division logistics management section.

The software developed for this thesis is microcomputer based because of price and availability to the Korea Army, especially at the division level and below. The software performs the supply distribution function, which is the most costly and labor consuming phase at the operational level.

It handles authorized storage list (ASL) as well as property items, to be defined at chapter II. The fixed order quantity with probabilistic demand model will be applied for inventory control.

\footnote{PPBEES is a newly developed system for the national resource management in Korea. The objective is to accomplish the economic management of the Army by evaluation of managers with budget expended and the designated material readiness condition.}
B. THESES ORGANIZATION

The thesis is divided into the following sections: As the background of the thesis, the second chapter covers the current ROK Army logistics structure, supply functions, and current computers in the Korean Army.

The third chapter describes the structure and the capability of the proposed system and its benefits.

Future research needs will be stated in the concluding chapter. Appendices include statistical considerations for inventory control measures in Appendix A, data structure of the proposed system in Appendix B, the proposed system structure in Appendix C. The user's manual is provided in Appendix D, and program listings and menu screen formats are in Appendix E and F.
II. ARMY LOGISTICS STRUCTURE AND ADPS IN THE KOREA ARMY

A. ARMY LOGISTICS STRUCTURE

1. Overview

The Korea Army is the largest of the three armed forces and is responsible for providing general-purpose forces to meet any threat to the Korean national security. In order for the Army to carry out its mission, it has to insure an uninterrupted supply of weapons, equipment, supplies and other items to the combat forces.

The Army logistic unit was established to enhance national security by providing a reliable and supportable supply of equipment and other necessary items to the various army units.

To achieve its objective, the logistic unit has adopted the following principals;

- **Logistics intelligence;** Commander must have accurate and timely logistics information in order to provide effective logistics support.
- **Objective;** Logistics endeavors must be directed toward a clear and attainable objective.
- **Generative logistics;** The professional application of initiative, knowledge, and ingenuity, and innovative exploration of technical and scientific advances are fundamental to the generation of logistics systems improvements.
- **Interdependence;** Logistics system efficiency requires integration with other functions of the system.
- **Simplicity;** Simplicity is essential at all levels of the logistics system.
- **Timeliness;** Logistics support must be provided in the right quantity at the proper time and place for accomplishment of the mission.
- **Impetus;** The impetus of logistics support is forward to support the combat mission.
- **Cost-effectiveness;** Efficient management of resources is essential.
- **Security;** Security must be maintained to preserve resources and ensure sustained combat capability [Ref. 1: p. 34]

2. The Army Logistics Unit Organization

The Army logistics unit is organized into three levels, each responsible for certain functions. The next two sections will outline the organizational structure and the functions of the logistics units.
a. Organizational Structure

The organizational structure is divided into three levels. The top level is
the wholesale echelon which includes depots, maintenance points, plants and factories
associated with special army activities controlled by the Army headquarters. The
intermediate echelon, i.e. retail sale echelon, is the next level and serves as an interface
between the top level and the direct support and use echelon. It is responsible for
providing general support function. The third and final level is the direct support and
user echelon which includes field units and provides direct support functions to the
various units in the field.

b. Support Functions

Logistics Units perform the following functions

- **Supply**: which includes procurement, distribution, maintenance while in storage,
  and salvage of all commodities necessary to equip, maintain and operate the
  armed forces.
- **Maintenance**: which focuses on repair and restoration of fielded weapons and
  equipment systems. Maintenance is classified into three level, i.e. unit, field and
  depot.
- **Transportation** of troops and supplies.
- **Services**: including food stores, clothes stores, laundry, grave services, fire
  fighting, etc.
- **Facilities**: including real properties such as depot, maintenance factories, and
  barracks, etc.

3. Management Issues

Four managerial issues were identified by the Korean Army Logistics
Command as critical to the performance and execution of the logistics tasks. These
elements are;

- Highly reliable communications between the logistics units.
- Retention of sufficient defense resources.
- High speed movement of combat support supplies.
- Utilization of automatic data processing systems in order to effectively manage
  the process. [Ref. 2: p. 636]

This thesis is an effort to support the four elements, i.e. the use of data
processing in developing a system to improve the retention of sufficient defense
resources; to reduce communication complexity and paperwork; to decrease the
response time for requests; and to allow management more time for important
decisions.
The next sections outline a specific management problem followed by, in later chapters, a proposed implementation of a computerized information system, within the constraints of the Korean data processing environment system.

B. SELECTIVE MANAGEMENT

1. Overview

Material management involves thousands of individual transactions each year. To do their job effectively, material managers must be able to effectively use their time, concentrate on critical items and avoid the distraction of less critical details.

In reality, it is difficult to achieve this without the following prerequisites:

- A classification system to identify critical items.
- Well established and structured procedures to deal with those items that are considered less critical.
- A computer system which assists managers in carrying out the less critical activities and which can identify changes that may effect an activity classification.

The Logistics unit has adopted a classification system which satisfies the first prerequisite above, i.e. inventory control procedures that isolate those items requiring precise control from those items that do not.

2. The ABC Classification system

This system is based on the fact that only a small percentage of inventory items account for most of the total inventory value. Thus considering the cost of management time, it is more cost effective to purchase a sufficient supply of low cost, low demand items and maintain little control over them.

Before discussing the classification system, it must be stated that before an item is given low priority classification, it is evaluated by the Army to determine how critical it is for combat. If it was classified as combat critical, it must be treated like high priority item even if its demand classification is still low one. Thus a low priority item has to satisfy two criteria; an item shortage will not disrupt the combat operations and its annual demand in terms of must be low.

The selective management, more commonly known as the ABC system, was adopted by the Korean Army to meet these supply function considerations in material handling. Each item is given one of three classifications, i.e. A, B or C (see Figure 2.1). Class A consists of items whose dollar value of total annual demand typically accounts for 50% of the total dollar value of the inventory, while representing only 7% of the
number of inventory items. The B class consists of 18% with 35% of inventory items. The C class consists of items whose annual dollar value accounts for only 15% of the total dollar value of the inventory but represents 65% of the inventory items.

![ABC Classification](image)

**Figure 2.1 ABC Classification.**

The entire inventory is listed in descending order from the largest value of the annual demand to the smallest and break points are between class A and B and between class B and C.

The ABC classification exists to direct attention to those inventory items that represent the largest annual expenditures. If inventory levels can be reduced for class A items, a significant reduction in inventory investment will result.

"The purpose of classifying items into groups is to establish appropriate levels of control over each item. ABC analysis is useful for any type of independent demand system (continuous review, periodic review, and so forth). With the periodic system, the ABC analysis can be subdivided so high usage items receive a short review and low usage items receive a much longer review." [Ref. 3: p.439]

As previously mentioned the degree of control is classification dependent, thus class A items require and are given the greatest attention while the class C items are paid the least attention. Class A items are managed using an economic order quantity
model. A review of the inventory position would occur each time an item is issued to a customer. Class B items could use an economic order quantity (EOQ) based requisitioning objective. Class C items require no special calculations, since they represent a low inventory investment. The order quantity might be a one year supply with an annual review of the inventory position.

The success of the ABC system is dependent on several factors, among them is the validity of the assumption that class A items constitute a small percentage, 7%, of the total items and around 50% of the total cost. Figure 2.1 depicts the distribution of the cost and quantity for the inventory in the Korean army.

The system proposed in this thesis applies continuous review on class A and some of class B items, which are included in the authorized storage list (ASL). ASL will be discussed in section three inventory control means.

C. THE EXISTING SYSTEM

1. The Planning-Programming-Executing-Evaluating System (PPBEES)

The Planning-Programming-Budgeting-Execution-Evaluation System (PPBEES) is a newly developed system for the national resource management in Korea. It evaluates managers based on their budget expenditure and their material readiness condition.

A number of plans were developed to implement this system. For instance Accounting System for Fund, and Fund Management Comparison and Evaluation System. They had less success than originally envisioned. It is my opinion that the effectiveness of the system can be increased by widening its implementation to include Army divisions and by establishing a vehicle for division managers to easily obtain accurate information needed in managing inventory. At the present time performance and operating statistics must be manually prepared, which is a time consuming process normally avoided by managers. The proposed system allows managers to acquire the needed information in a timely fashion and to support both planning and internal performance audits.

The PPBEES measures performance based on a subset of the supplied items. The following sections outline the division supply system and the items included in the PPBEES system.

---

2These two names are literally translated from Korean.
2. Resource flow in the infantry division

The resources (or fund) flow in the division is divided into two major categories, materials and cash. The materials are about 55% of total value, and are the subject of the proposed computer system in this thesis, while the remaining 45% is cash.

The materials issued to the division are divided into three groups resource control number (RCN), automated supply items, and non monetary valued items, see Table 1.

<table>
<thead>
<tr>
<th>TABLE 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>BUDGET RATE OF MATERIALS</td>
</tr>
<tr>
<td>-----------------</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Item</th>
<th>Contents</th>
<th>Percent</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Total</td>
</tr>
<tr>
<td>PPBEES Materials</td>
<td>Parts, Gasoline, Medical Equipment, General Supply</td>
<td>15</td>
</tr>
<tr>
<td>Automated Issue Items</td>
<td>Food, Clothing, Heating Fuels, Tactical Construction Materials, Ammunition, Assemblies, End items</td>
<td>40</td>
</tr>
<tr>
<td>Non Monetary Valued Items</td>
<td>Clothing(Officer, NCO) Field manual, Technical manual Target</td>
<td>Unknown</td>
</tr>
</tbody>
</table>

The RCN grouped material constitutes around 15.3% of the resources, i.e. 30% of material, and are issued by the Logistics Support Command with RCN. These are the only items tracked by the PPBEES system and used in evaluating managers' performance.

The remaining materials, 40% of the budget, are issued automatically without user's request. We can anticipate that this results in excess stock in some units while the other unit run out. The division has no evaluation system for these automated issued line items. [Ref. 4: pp.194-196]

3RCN (Resource Control Number) is the numeric symbol which is used for accounting, execution of supply activities, that connects the cash budget and material supply.
The division has only basic manual means for recording material consumption. The inventory report card contains only the present amount of stock for each item in the organization unit. The division should have a system to record material consumption and evaluate support performance.

In the present PPBEES there are no means to evaluate the ranking of the units in a given period, as well as a number of other problems which make an automatized system highly desirable. For example:

- For certain items, the logistics support command sets a ceiling (maximum allowance) on the number of units of the item which may be consumed by a division each calendar year. The performance of the division with respect to this item is judged based upon the fraction of the maximum allowance which is actually used during the year. Obviously divisions with larger maximum allowances use items more freely, while divisions with smaller maximum allowance may suffer from lack of material. This may induce a division commander to delay the maintenance of equipment, to continue to use substandard material, and to occasionally make inappropriate substitutions.
- There are too many elements to compare. For each RCN item there are 20 criteria used in the evaluation. Each division has 20 units and each regiment has 23 companies. Division managers do not have the required resources to do this comparison manually.
- The system requires too much paperwork. Headquarters must calculate costs by equipment and by subordinate units. Both are difficult manual efforts which may include significant errors.

3. Inventory operation and control in divisions

The supply procedure in divisions is shown in figure 2.2. The principal actions between supplier and customers are: requests for issue; turn-ins; and cancellations. These actions take place between a unit and a division and between a division and the logistics support command. For example when a company sends a "request for issue" to a regiment, the regiment passes it to division. The division issues the item if it is on hand; if not, a request is sent by division to the logistic support command. The logistics support command processes the requisition through their computer system, and generates an issue list containing all the items requested by the division. After the division receives the list and material, all due-outs to division customers are filed and the remaining material is stored pending the next request. All requests for issues from lower units are handled daily. The proposed system follows this basic procedure.

The control measures in these procedures are the authorized storage list (ASL), requisition objective (RO), safety level (SL), and order shipping time (OST).
A. Supply procedure from division

1. Request for issue or cancel
2. Request for issue or cancel
3. Receive from Logistics Support Command
4. Issue to organization
5. Turn in to Logistics support command
6. Turn in from organization

B. Current Transaction record procedure

<table>
<thead>
<tr>
<th>Company</th>
<th>Regiment</th>
<th>Division</th>
<th>Logistics support command</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Telephone</td>
<td>Transaction list #1</td>
<td>Transaction list #1</td>
<td>Request for issue</td>
</tr>
<tr>
<td></td>
<td>list #2</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Transaction Record book</td>
<td></td>
<td>Request for issue</td>
</tr>
<tr>
<td></td>
<td>Transaction list #2</td>
<td></td>
<td>Computer Process</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Issue List</td>
</tr>
<tr>
<td>Property book</td>
<td>Property book Issue and voucher list</td>
<td>Integrated voucher records</td>
<td>Stock accounting record</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Fund accounting record</td>
</tr>
</tbody>
</table>

Figure 2.2 Division Supply Procedure
a. **Authorized Storage List (ASL)**

Authorized storage list defines all supplies which can be stored by the supply unit to meet expected demand. This is based on selective management similar to the ABC classification. The ASL considers economic and support performance objectives. If support performance was the only consideration, all items will be stored. This obviously would be very costly, would increase the difficulty handling the material and decrease the maneuverability of the unit. On the other hand if we consider only economics the unit performance would be degraded and there would be a serious increase in unnecessary stock. Therefore the stock should be the minimum required to perform the mission.

To get the minimum required stock, we have to consider measures of effectiveness (MOEs): Let's assume that total cost = \( f(x) \) and support performance = \( g(x) \). We have to minimize \( f(x) \) and maximize \( g(x) \), but as \( x \) increases both \( f(x) \) and \( g(x) \) increase. There is conflict! So what is usually done is:

1. Choose a specific minimum value for \( g(x) \), i.e., \( g(x) \) must be greater than this value (call it \( b \)).

2. Minimize \( f(x) \) subject to \( g(x) \geq b \).

A ranking similar to the ABC system was developed for the demand frequency of each item. As shown in Table 2 - (A), 85% of total demand frequency includes only 15% of all items required by the supported unit; even if we increase this latter percentage to 50% and reexamine the associated fraction of total demand frequency, the increase of demand frequency is only 3-4%.

This fact suggests that by storing 15-20% of all items it is possible to meet 85% of the support requirement. This is the basis of ASL, i.e., satisfying 85% of all requisitions. The achieved DFR is the ratio of demand for ASL items to the total effective demand. Total effective demand is the demand of all items less cancellations.

ASL is divided into 6 groups: active items, stand-by and essential items, Prescribed Load List (PLL) of Organization Unit, the supported units ASL, direct exchange and repair parts for new equipment, and substitution of maintenance.

- The active items are the items included in 85% of the demand frequency rate.
- Essential items for any future emergency are decided by the Army Commander.
- Prescribed load list of organization unit are repair parts and tools intended to give worth of support 15 days and must be stored in each organization.
- When two or more items have functional and physical characteristics that cause them to be equivalent in performance, reliability and maintainability, only one of the items will be on the ASL.
### TABLE 2

#### A). DEMAND FREQUENCY RATE

<table>
<thead>
<tr>
<th>Demand frequency rate (%)</th>
<th>ASL</th>
<th>Non ASL</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>14</td>
<td>5</td>
</tr>
</tbody>
</table>

**Graph:**
- **Demand frequency**
- **Annual dollar value**

#### B). AUTHORIZED STORAGE LIST

<table>
<thead>
<tr>
<th>ASL Items</th>
<th>Non ASL Items</th>
</tr>
</thead>
<tbody>
<tr>
<td>Immediate issue</td>
<td>Delayed issue</td>
</tr>
<tr>
<td>Total Annual effective demand</td>
<td>85%</td>
</tr>
</tbody>
</table>

**Demand flexibility rate = \( \frac{\text{ASL}}{\text{Total effective demand}} \times 100 \)**

#### C). ANNUAL DEMAND FREQUENCY FOR ASL

<table>
<thead>
<tr>
<th>Chemical Items</th>
<th>6times</th>
<th>Quatermaster Items</th>
<th>6times</th>
<th>Engineer Items</th>
<th>3times</th>
<th>Communication Items</th>
<th>3times</th>
<th>Ordnance Items</th>
<th>6times</th>
<th>Transportation Items</th>
<th>6times</th>
<th>Medical items</th>
<th>6times</th>
</tr>
</thead>
</table>
When stock changes or when catalog data are updated, all additions or deletions are made in the ASL. Other changes include inter-changability and substitution between items. The deputy chief of staff logistics (DCoSLOG) in the Army HQ reviews the ASL semiannually for potential range reductions. Inclusion on the ASL depends on the minimum number of times an item is ordered as shown in Table 2-C.

b. Requisition Objective (RO)

Each ASL item must have an RO recorded in the stock accounting record. The RO is the maximum quantity of the item authorized to be on hand and on order at any time. Retention of assets above the RO is authorized under certain conditions. The RO computations are made in a days of supply (DOSs) mode or economic order quantity (EOQ) mode.

1. Days of Supply (DOS). The DOS mode will be used by nonautomated accounts when computing stock for items that are critically short, seasonal, highly perishable, or have a shelf life of less than 3 years. A DOS RO will be computed at least semiannually, or when the balance on hand is equal to or less than the reorder point (ROP), or when the balance on hand equals zero.

The DOS RO is the sum of the operating level (OL), safety level (SL), and order shipping time (OST) in days; multiplied by the quantity demanded during the control period (one year for division), divided by the number of days in the control period. The ROP is the sum of the SL and OST in days, multiplied by the quantity demanded during the control period divided by the numbers of days in the control period.

2. Economic order quantity (EOQ). The EOQ RO is the sum of the EOQ and the ROP quantities which will minimize the total variable cost of stock for a specified performance goal. The performance goal is based on how essential the item is. The amount of stock may be constrained by mobility and fiscal limitations. The total variable cost consists of order cost (replenishment and wash actions), change cost (cost to add and delete an item), cost to maintain an item in stock, storage holding cost, and the implied shortage cost. The implied shortage cost is not a measurable cost (to calculate this cost refer to the Appendix A statistical consideration.)

The implied shortage cost will be used in automated systems as a variable to adjust a stock level to meet a specified performance goal. Stock criteria will be variable when the EOQ is used by automated systems. The EOQ mode will be used by automated accounts (higher than division level so far).
c. Order shipping time (OST)

OST is used computing both EOQ and DOS. It is the average number of days that elapse between the document date of the requisition and the date the receipt is posted to the stock accounting record.

When OST is not available, as in the case of the manual system used in DOS, it has to be computed. The computation is based on the average OST of the six most recent replenishment receipts and rounded to the next higher number of whole days. In computing OST, requisitions are excluded if they have long delays from wholesale backorder, unusual circumstances, or lack of funds. A cumbersome manual process!

OST is updated each time the RO is recomputed. Because of the fact that most of the system is manual, updates to the OST is done annually.

4. Current documentation and report forms

The documents maintained by division logistics departments are classified by contents and include:

- Inventory status for each item
- Evaluation worksheet for the PPBEES
- Documents for TAMMS
- The record of consumable item consumption

These report mainly focus on the historic data and are difficult to use in computing statistics and information needed by managers. For example the current documentation is inadequate to use in material planning. Although it shows the current stock level of items, funds, etc. This data can not be used by a computer and is difficult to retrieve, sort or manually manipulate to calculate the needed information.

The format of the recording form is inconvenient. The documents or forms should be able to prove the transaction history and also be user friendly. In the current documents, transactions are recorded by date sequence, when any cancellation of requisition occurs, it is very difficult to locate and correct all previous records. Records written in pencil are hard to use as vouchers.

---

TAMMS is a abbreviation of The Army Maintenance Management System which contains the historic records of maintenance, and operation of all equipment. This data will be used in requisitioning of the new equipment requisition.
D. CURRENT USE OF COMPUTERS FOR LOGISTICS IN ROK ARMY

1. Overview

In March, 1967 the first computer was introduced in Korea to assist with a census of the Korean population for the Economic Planning Board. The Korean Army installed its first computer system in the Army HQ for the management of military personnel in 1972. The next year another computer was installed at the Logistics Command to assist with logistics management. Subsequently several computer centers were established. All these computer centers are directly controlled by the staff of the Army HQ. [Ref. 5: p.14]

![Diagram of Army Logistics Computer System]

In the late of 70's, an integrated software development center was established at the Central Automatic Data Processing Center (ADPC) to develop software for the mainframe computer used by the logistics command and Army HQ. The second center, the Logistics Management Information Center, was established to support MIS development activities in the G-4 of the Army. It endeavors to improve MIS capability and has gained the attention of high level managers.

The hardware installed in the various computer centers is different. IBM 370, and UNIVAC 90/30 and 1100 series machines are very common. These are batch system and are not connected with each other. The application software, which is run periodically by users, was developed using old technology, i.e. flat file system.
The file system and the fact that the computers are not connected results in a high degree of redundancy. Each computer center, even those under the control of the Department of Computers at Army HQ, has several files containing the same data elements. For example the Army personnel system, payroll system, and medical system all contain common data elements which are updated independently, a potential source of data integrity.

A second problem which has affected the development of the data processing field in the army is a shortage of qualified personal.

Recently, high level managers have recognized the need for the standardization of hardware and unification of application software. This has resulted in an ongoing effort to bring the data processing systems in Korea to the leading edge of technology. As a result, a number of mini- and microcomputers have been installed at the division level.

2. ADP Support to logistics

It is necessary for the commanders to have adequate forecasting capability for the effective command and control his unit. The use of ADP systems has significantly increased the commander’s visibility and has had an effect on logistics operations. The Automatic Data Processing Center (ADPC) within the logistics structure has provided significant support. The ADPC, dedicated to logistics operations, supports its own internal functions such as stock control within the responsible area and routine jobs such as reports generation for higher commands. Additionally the center provides to other departments the logistics information. In the division level, reports are manually generated.

One concern of this thesis is an important report generating function of ADPC; the inventory status report. This report is presently created by division G-4 for logistics support command manually.

This reporting system will be designed in this thesis to provide up-to-date accurate inventory status data for major items pertaining to each division. In addition to the inventory status report there are many other reports which are needed to control the inventory. These reports provide information to the division commander and higher logistics support units so that readiness can be evaluated. These reports also indicate the shortage and overage of material and, when integrated at higher levels, allow the command to determine new procurement needs, prepare budgets, redistribute assets and take disposal actions.

28
E. SUMMARY

This chapter presented a description of the existing inventory control system and data processing capabilities. Several issues, stated in this chapter, were paramount in the decision to develop the proposed inventory control system. For example:

- The existing system has limited capability in the development of effective material management.
- The PPBEES implementation will benefit from a computerized Management Information System.
- The documentation in the present system is cumbersome and inadequate for managers.

A database management system can be used to resolve these problems. A well-developed one will provide the information needed, maintain records of transactions and exchange the information with other computers.

This system has to be user friendly and reduce to an acceptable minimum the dependency on the data processing personnel. Its implementation, including installation, must be simple and must use the microcomputers available in the various units.

The problem the system resolves is well structured to insure the system's success. At the same time the system design has a degree of flexibility to expand at a later point. The existing system is labour intensive and the proposed system will save managers precious time for more important activities.

The system described in this thesis was designed with these facts in mind. As such it resolves a problem by using microcomputers, and a user friendly interface, and allows for electronic communication of data.
III. RELATIONAL DATABASE MODELS

A. INTRODUCTION

The proposed system is a relational database management system which uses dBASEIIIplus. In this chapter, a brief description of relational database models is given.

B. WHAT IS DATABASE?

A Database is basically a computerized record keeping system, that is, a system whose overall purpose is to record and maintain information for future retrieval. D.R. Howe, the author of Data Analysis for Database Design, defined a database as "a collection of non-redundant data shareable between different application system". He extended his definition by saying that 'non-redundant means unnecessarily duplicated data adds no new information', sharing data as 'a multiple usages in multiple applications'. [Ref. 6: P.1]

A database should be structured so as to provide a foundation for future application development. It can be manipulated into information for management purpose.

A database model is an abstract representation of data. It defines the way that data items are organized and related. There are two major classes of database representation, the physical and logical models. The physical model represents the actual structure of the data in the computer. The logical model represents how a user perceives the data organization. For example, in an inventory system, we may have the following case;

A number of parts are located in a number of warehouses. shown on Figure &str-A.

The physical structure may consists of the following files

- A part file which consists of the part number and all the attributes which specifically describe this part, such as part name, price, weight, and the vendor supplying this part (assuming that a part is supplied by one vendor only), etc.
- A warehouse file which consists of a list of all warehouses and those attributes unique to each warehouse.
- A relationship file which consists of the attributes that describe the relationship between parts and warehouses, e.g. number of units of part 1234 (QUANTITY) in warehouse A-12 and location of the part in this warehouse (bin 27AC)
The logical structure is a user view, and as such it is dependent on the user. For a warehouse manager, the logical view may be similar to the one shown in Figure 3.1-B, i.e. the warehouse data and the parts available in that warehouse and the vendor of each part.

A second user view may be that of a worker on a machine. In this case the person is interested in seeing the part data and attributes as well as the warehouses where this part is. He is not interested in knowing the vendor or the location of the part in a warehouse.

This type of file structure facilitates adding new views based on user requirements without changing the physical structure of the files and reduce the number of redundant elements and the update effort. Programs share the data instead of having the same data duplicated for each program, creating an update nightmare.

The database management system is responsible for managing the physical storage of data. Thus if a data element physical characteristic changes, no programming changes are required since each program requests a data element by name and is, to a great extent, independent of the element's physical characteristics.
C. RELATIONAL DATABASE MODEL

The database model discussed in the previous example is called the Relational Model. A basic quality of the relational model is its simplicity.

It was introduced by Dr. E. F. Codd in a seminar paper in 1970. The paper have been written concerning the most appropriate way to express relations. He stressed the importance of the relational representation and physical computer implementation such as ordering on physical devices, indexing, and using physical access paths. Although the relational model has many desirable characteristics, it was, until recently, a subject of theoretical interest only. In the last few years commercially viable database management system (DBMS) became available and in the early 1980s, several important DBMS products were introduced. SQL, DB2 (provided by IBM) and ORACLE (provided by Relational Software Incorporated) are two examples. Since these announcements, the relational model has come to be of greater practical significance.

[Ref. 7: p. 242]

The following definitions introduce a number of terms that will be used in describing the proposed material control model.

1. Relation

A relation is a table of data (a file) and consists of rows (tuples) and columns (attributes). The data table provides a simple data structure.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Attribute</th>
</tr>
</thead>
<tbody>
<tr>
<td>SN</td>
<td>NM</td>
</tr>
<tr>
<td>UNIT</td>
<td>ONHAND</td>
</tr>
<tr>
<td>Tuple 1</td>
<td>Test item1</td>
</tr>
<tr>
<td>Tuple 2</td>
<td>Test item2</td>
</tr>
<tr>
<td>Tuple 3</td>
<td>Test item3</td>
</tr>
<tr>
<td>Tuple 4</td>
<td>Test item4</td>
</tr>
<tr>
<td>Records</td>
<td>Test item5</td>
</tr>
</tbody>
</table>

Figure 3.2 Relation of property.

Figure 3.2 shows an example of a table (or relation); the table contains six columns (each represents an attribute) and five rows (tuple, each represents a specific record). The intersection of each row and column in the table contains a value. For instance,
Test item1 is an attribute value of the attribute type “NM” for the record belonging to
stock number 1111-11-111-1111.

There are a number of constraints that must be observed when building tables:

- The ordering of rows is not important because the rows can be interchanged
  without affecting the information content of the table.
- The ordering of columns is not important either, for the same reason.
- Each row and column intersection contains a single attribute value. Multiple
  values are not allowed.
- Each row in a table must be distinct; no two rows can have the same attribute
  values throughout. (The significance of this rule is that a row can always be
  uniquely identified by quoting an appropriate combination of attribute values.)

An attribute value may be null, but a null value does not mean that the
attribute is blank, e.g. the unit cost of test item4 may be entered in the table after the
record has been created.

Each attribute has a domain, a set of values that the attribute can have. For
example, the domain of unit cost is a positive eight-digit number with 2 decimal points.

2. Keys

We want to be able to identify each tuple (record) in a relation by the value of
at least one of its attributes. In Figure 3.2, the stock number (SN) is a unique
identifier since no other row may have the same stock number.

If there is no one unique attribute, then a combination of more than one may
have to be used. The table data structure shown in Appendix C includes examples of
keys which have more than one attribute.

3. Record relationships

The essence of a database is the representation of record relationships. The
relationships can be specified in a variety of ways.

The relationships are identified intuitively. The designer considers potential
relationship among records that have been defined. A relationships may exist among
three or four or more records. For example with records of the proposed system,
MASTER records have many property records, while one PROPERTY record has only
one ASL record. Many STOCK-OUT records have many MASTER records.

4. Database Management System

A database management system (DBMS) is a software system which performs
the functions of defining, creating, revising, and controlling the database. It provides
facilities for retrieving data, generating reports, revising data definitions, updating data,
and building applications.
Several relational DBMS are available. The one used in developing this inventory control system is Dbase III Plus. The version used is the Microcomputer version. However, an extension which includes SQL, a facility which allows sharing of data with mainframe computers is under development and will be introduced in the very near future.

DbaseIIIPlus is the market leader among DBMSs in the United States. Several supporting products have been introduced such as compilers to increase execution speed and application generators to optimize the code execution. It has network capabilities which allow sharing of data, and security procedures not available in the single user version. It also has its own programming language, which facilitates the development of user friendly interfaces.
IV. THE PROPOSED SYSTEM

A. OVERVIEW

This chapter describes the existing manual procedures and the proposed computerized system.

B. MANUAL SYSTEM

Presently, a logistics unit in a division receives a request for issue of an item, an item turn-in, or cancellation of a requested item from lower level organizations. The same set of requests occur between the division and the LSC. Transactions are recorded in property books and this generates a second type of process, that is the editing of erroneous entries.

Transactions are aggregated to produce management reports, which are used during the analysis for the next planning cycle and for evaluating the performance of managers.

Two sets of books exist. The first book is the transaction log in which records of each request are kept. Each entry includes the transaction number, date, item ordered and quantity. The second set is the property book which has a page for each stock number and the on-hand quantity of this item. When a request is satisfied, both sets of books are updated.

C. THE PROPOSED SYSTEM STRUCTURE

As requests arrive at the division, either from outside entities or internal personnel, the end user enters the request data using a screen identical in its format to the request form. Entries are edited by the system for errors using hard coded criteria, and stored.

The system consists of several files and modules. The Batch and Master files are similar in structure to the Transactions Record Book, while the Property File and the ASL file are similar to the Property Book. An item is included in one of these latter files based on its classification and whether the item is PPBEES trackable or not. The Stock Out file combines elements from both books and the Customer file contains customer's attributes such as address, customer ID, zipcode, etc.
Modules in the system perform several functions. For example, when transactions are processed the system updates the quantity on hand, creates a transaction record, the completion date of a transaction is generated and the transaction record is moved to the MASTER file. The date field is used for performance measurement purposes. For example the date field will help answer questions about turn around time for a specific item.

If the stock on hand after satisfying the current demand is less than the reorder point, an attribute for each item, the stock on hand field, is updated to reflect the current onhand amount and a request for issue to the supplier will be generated automatically. When the amount is less than zero, a stock-out record is added to the STOCK-OUT file, and the system sends a delivery delay warning to the user. This process is illustrated in a flow chart in Appendix C.

One issue list per customer is generated, regardless of the number of items requested in one day, under one receipt voucher number. The same is true for a request for issue to LSC. Figure 4.1 shows examples of the issue list and request for issue. The processing of the other type of transactions follows almost the same procedure.

When the supplier issues stock to the division, the stock is examined to determine whether it is new, i.e. initial supply, and if it is, the program will ask the user to enter it in the property file, and in the ASL file if necessary.

The fact that the system does not require more than one entry per request and that it interrogates all of the appropriate files, without end user intervention, drastically reduces the amount of end user time per request.

The system is flexible and can accommodate a number of potential user requests. For example, If a user wants to extend the output information to the type of operation, he can add one attribute to Master file for the operation type, the query program can easily support such requests.

Figure 4.2 depicts the data flow in the proposed system and Figure 4.3 shows the transaction data flow diagram.

D. THE PROPOSED SYSTEM FUNCTIONS

1. Transaction tracking

The main purpose of the transaction tracking is to record each customer's request for inventory planning and management purposes. The proposed system generates reports on the status of each customer's or stock item. Transaction records are saved in the Master File for analysis of trends. The ASL file provides criteria such as Reorder point, requisition objective, and safety level.
ISSUE LIST

From: The 150 Infantry Division
To: The 150 Div 1591 regiment

Date: 11/13/87

Voucher No: 1500QM-0029-7317

No Stock number Description Onhand Unit Quantity Price Total

1 1111-11-111-111 Test item SORRY! Delay delivery
2 2222-22-222-222 Test item SORRY! Delay delivery
3 3333-33-333-333 Test item SORRY! Delay delivery

Grand Total: $900

Material Management NCO: ____________________________ Date: __/__/___

Material Management Officer: __________________________ Date: __/__/___

REQUEST FOR ISSUE

To: The 3333 Logistics Support Command
From: The 150 Infantry Division

Request No: 1500QM-0014-7317

Date: 11/13/87

No Stock number Description Unit Quantity Price Total

1 1111-11-111-111 Test item Ea 105 10 1050
2 2222-22-222-222 Test item Ea 22 22 440

Grand Total: $1490

Material Management NCO: ____________________________ Date: __/__/___

Material Management Officer: __________________________ Date: __/__/___

Figure 4.1 Examples of transaction output.

For management purposes, a manager may ask specific questions on a transaction record, a customer's transaction history, or the status of a certain stock item. However, the system stock item queries are limited to the status of the item. An extension to determine the location of an item may be added with minor changes to a database file. This programming effort may be done by a knowledgeable user.

2. Report Generation

One of the most important functions of the proposed system is to generate reports required by the logistics support command and the internal division manager. The system has a number of hard coded reports and a simple ad hoc reporting facility. The hard coded reports are described in Appendix C.
Figure 4.2 Main data flow diagram

Figure 4.3 Transaction data flow diagram
Reports may be classified, into two groups, internal use and external reports. Internal use reports are those used for performance analysis. The external reports are the transaction output, e.g. the issue list resulted from request for issue of the organizational unit, request for issue, request for turn-in are for Logistics Support Command.

Figure 4.4 shows the data flow for report generation. More reports may be added to the system, if requested, this will bring the system closer to a complete inventory planning management information system.

![Diagram]

Figure 4.4  Report data flow diagram.

3. **Limited analysis**

   The proposed system uses the fixed order size probabilistic model for determination of reorder point, safety level, and economic order quantity.

   The assumptions involved in the proposed system are:
   - The demand is continuous and normally distributed,
   - Annual demand can be obtained through the transaction history record,
   - There are no seasonal effects on demand,
   - The lead time is constant,
• Service level for each item is known or determined by managers,
• The stock-out cost is known.

Other statistical considerations are presented in appendix A.

The annual demand is determined by summing the effective demand (total demand - canceled demand) and the monthly demand is calculated by dividing annual demand by 12. The standard deviation of demand is calculated from the transaction record history. The lead time is obtained by computing the arithmetic average of the order shipping times from historical data for a specified period of time.

After all parameters are determined, the reorder point, safety level, and requisition objective are calculated and updated.

You selected stock number: 8888-88-888-8888
(Test item, Unit: Ea, Class: S)
Purchasing price (P): 22 $/Ea
Annual Demand (R): 100 Ea/year
Lead time (OST) in Month: 0 Month(s)
Ordering cost (C): 0.00 $/order
Holding cost unit per year: 0.00 %
(Select one of these)
Stockout cost (If Known): 0.00 $/unit Select? N
Service Level in year: 0.0000 % Select? Y

[PRESENT] Reorder point: 15 Safety Level: 10
Requisition objective: 50 Lead Time: 14

Is this record right? (Y/N): X

Figure 4.5 The output from analysis process.

4. Use friendly interface

One of the purposes of this thesis is to design a user friendly computer program. Novice computer users can use it without special training. The proposed system provides a user friendly interface by minimizing the possibility of errors. The user does not have to know about the disk operating system. And a manager who is not familiar with a division inventory system can use it, since most of the transaction processes proceed automatically.
E. SYSTEM LIMITATIONS

The proposed system is not a perfect system. An objective of this system is to improve logistics management capability. This is a difficult task and requires many additional software applications. For example, better system would include accounting and inventory tracking functions. The proposed system addresses just one main logistics function. Constructive comments and remarks are welcomed and solicited.

1. Demand forecasting

Demand forecasting forms the basis of managerial decisions and if it is accurate, managers don't have to worry about safety level or lead times.

The proposed system does not forecast demand. The annual demand quantities are based on one year historical records or as specified period. Managers can retrieve historical demand data in various forms, by item, period, unit, or combinations, and use it in forecasting.

An extension to the software is needed to provide demand frequency information by unit by type e.g. training, exercise, routine job, or special purpose.

2. Statistical analysis of transactions

Statistical analysis was not fully covered in this system. Only two models were presented for analysis: the known stock-out cost model and service level model under normally distributed demand, were presented. There are a number of other models for inventory management. The system should be enhanced by adding various models and allowing organizations to choose the model appropriate for their needs.

3. ASL request

As was mentioned in chapter II, the division commander can ask for additional items for the ASL for his division. The basis of for this request is also described Table 2 in chapter II. The query menu in the software written for this thesis allows the division logistics manager to quickly determine if a particular item will qualify for stockage in the ASL based upon its demand history. The software for this thesis does not, however, include the facility to automatically generate a request to add a particular item to the division ASL.

4. Fund Accounting

The accounting of the fund (of RCN) is important to managers, so the proposed system is partially prepared to do this function. The resource control number (RCN: an attribute of an item) is based on the current PPBEES. While the proposed system, includes all the needed elements, it must be extended to perform accounting functions.
F. ADVANTAGES OF THE SYSTEM

By applying the proposed system:

1. The Army can improve one of its logistics objectives: automatic data processing.

2. The system contributes to the PPBEES implementation by generating timely and accurate reports for various supply units.

3. It allows for more effective utilization of personnel time by shortening the administrative process.

4. More accurate records can be maintained.

5. The system can be extended, with minimal effort, to include fund accounting.
V. CONCLUSIONS

A. CONCLUSIONS

The proposed system is a user friendly, menu-driven, modularly designed database management system focusing on the supply distribution function of the ROK Army division. It will reduce investment in personnel, and material and generate timely and accurate reports serving division managers. The system may be operated by novice as well as experienced personnel.

To develop this system, the ROK Army logistics structure, current logistical management methods, and problems were described in chapter II. The same chapter describes current computer usages in the ROK Army.

The third chapter includes the structure of the database, its major capabilities, and benefits. The chapter presents the fundamentals of databases and the database management system used in developing the system, i.e. dBASEIII PLUS.

The fourth chapter describes the proposed system: what it can or can’t do for the logistics control management in the ROK Army division. The chapter also addresses the benefits of the implementation of the proposed system.

Through the development of this system, the author achieved the primary goal of this study by combining the knowledge of the ROK Army logistics management, relational database, design of the database management system, and programming techniques. The implementation of the DBMS in ROK Army division should contribute significantly to the supply effort.

By using this program, the ROK Army division can reduce the current volume of paperwork and overtime devoted to routine tasks. The proposed system can save personnel and material resources, thereby freeing these resources to other vital objectives such as enhancement of the combat power.

B. FURTHER STUDY

In this thesis, the author deals only with the distribution function from among many other logistics functions. A more comprehensive system would include accounting, maintenance tracking, and provisions for analyzing transaction records system.
Different inventory control models would be applied to enhance the database management system. Only two of the many models available have been implemented in this thesis.

Networking and interconnecting this system with available mainframe systems is another issue not addressed in this thesis. A final objective is to tailor this system for use in combat situation.
APPENDIX A

STATISTICAL CONSIDERATION

The material in this appendix is extracted directly from Tersine [Ref. 3: 131-165]. It is given here because the decision of safety level, reorder point in the proposed system is based on.

According to Tersine, when the demand is probabilistic (not deterministic), it is necessary to minimize the expected total cost for inventory management. If the demand distribution is continuous, the minimum expected total cost expression is obtained by taking the derivative of the total expected cost with respect to the decision variable and then setting it equal to zero.

If leadtime demand is distributed according to a continuous probability distribution \( f(M) \) then mean leadtime demand is given by:

\[
M = \int_{0}^{\infty} M f(M) dM \quad (eqn \ A.1)
\]

and the standard deviation of leadtime demand is the square root of the variance which is given by:

\[
\sigma^2 = \int_{0}^{\infty} (M - M)^2 f(M) dM \quad (eqn \ A.2)
\]

Where:

- \( M \) = random variable for lead time demand
- \( f(m) \) = probability density function of lead time demand
- \( \sigma \) = standard deviation of lead time demand
- \( M \) = mean lead time demand

The probability of a stockout for a given item is simply the probability that the demand during the lead time will exceed the reorder point. The stockout probability is the first definite integral of the probability density function of demand during the lead time from the reorder point to infinity.

That is:

\[
P(M > B) = P(s) = \int_{B}^{\infty} f(M) dM \quad (eqn \ A.3)
\]
Where: \( B = \) Reorder point
\( M = \) Random variable for leadtime demand

The expected stockout quantity during the lead time is the second definite integral of the probability density function of demand during the lead time from the reorder point to infinity.

That is

\[
E(M > B) = \int_{B}^{\infty} (M-B)f(M)dM \quad \text{(eqn A.4)}
\]

When demand is normally distributed, the reorder point can be obtained from the following formula;

\[
B = M + S = M + Z\sigma \quad \text{(eqn A.5)}
\]

Where
- \( M = \) Mean lead time demand in units
- \( S = \) safety stock in units
- \( Z = \) Standard normal deviate corresponding to the desired stockout probability
- \( \sigma = \) Standard deviation of lead time demand.

**Known stockout cost per stockout event**

The proposed system assumes that demand is normally distributed and lead time is constant (that is realistic for many items). It is also assumed that a fixed shortage cost is incurred one time only in each reorder cycle which experienced a stockout situation. This cost is assumed to be unaffected by the number of units or requisitions which are backordered during a reorder cycle. Since the historical distribution of demand is available, the safety stock can be determined by selecting a safety level that results in the lowest expected cost. It is easy to determine the safety stock using this method. The objective is to minimize the sum of the cost of holding the safety stock and the cost of the stockouts. The danger of stockout occurs only during the lead time. There are \( RQ \) lead times of opportunity for a stockout to occur is obtained by taking the derivative of the expected annual cost of safety stock with respect to the reorder point and setting it equal to zero.

**Annual cost of safety stock** = (holding cost) + (stockout cost)
\[ \text{TC}_s = SH + GR Q \left( \int_B^\infty f(M)dM \right) \quad (\text{eqn A.6}) \]

\[ = H(B - \bar{M}) + GR P(M > B) Q \]

\( \text{TC}_s \) = expected annual cost of safety stock
\( B = S + \bar{M} \) = reorder point in units
\( S \) = safety stock in units
\( H \) = holding or carrying cost per unit per year
\( G \) = backordering cost per outage
\( R \) = average annual demand in units
\( Q \) = lot size or order quantity in units
\( f(M) \) = probability density function of lead time demand
\( \bar{M} \) = average lead time demand units

If the derivative of the expected annual cost of safety stock with respect to the reorder point is taken and set equal to zero, the following relationship is obtained;

\[ f(B) = HQ GR \quad (\text{eqn A.7}) \]

However, for normal distribution, the optimum reorder point is not obtained from above equation, since the ordinate \( f(B) \) undergoes a change of scale when it is transformed to the standard normal distribution. Thus, if we are to find \( f(B) \), we must find here the standard normal distribution has an ordinate of \( f(B) \times \sigma \). The standard normal deviate \( Z \) for the optimum stockout probability can be obtained directly from the standard normal table.

The optimizing safety stock can be obtained from normal database and the following formulas;

\[ Z = (B - \bar{M}) \times (\sigma_D) \times \sqrt{\bar{M}} = S(\sigma \times \sqrt{\bar{M}}) \quad (\text{eqn A.8}) \]

\[ S = Z \sigma = Z \sigma_D \sqrt{\bar{M}} \quad (\text{eqn A.9}) \]

where

\( Z \) = standard normal deviate
\( B = D \times L + S \) = reorder point in units
\( \bar{M} = D \times L \) = average lead time demand
\[ \sigma = \text{standard deviation of lead time demand} \]
\[ L = \text{lead time} \]
\[ S = \text{safety stock in units} \]
\[ \sigma_D = \text{standard deviation of demand for a time period other than the lead time.} \]

**Service level is given instead of stockout cost per outage**

Most of the time in the ROK Army stockout costs are very difficult to obtain because of its accounting system. Under these circumstances, using set service level is a more rational way of determining safety stock. A service level indicates a level of ability to meet customer demand from stock.

The establishment of a service level is a subjective management judgment that is based on convenience rather than scientific justification. The choice by management of service level implies a cost attributed intuitively or indirectly to stockout.

**Example:** If the annual demand for an item is normally distributed with a mean of 8000 units and standard deviation of 1000 units, what should the safety stock and reorder point be if the lead time is 1.2 month? (Assume management has decided it is willing to be out of stock in 5% of the order cycles)

From the normal distribution table, \( Z = 1.64 \) corresponding to a 5% one tail risk of stocking out.

\[ D = R \times L = 12 \times 667 \]

\[ \sigma_D = \sigma \sqrt{(L+12)} = 288 \]

\[ B = M_a = (D \times L) + 1.64 \sigma_D \sqrt{L} = 669 \]

\[ S = M_a - D \times L = 335 \]

Where:

- \( R = \text{Average annual demand} \)
- \( M_a = B = \text{lead time demand at acceptable service level in units} \)

The safety stock is 335 units, and the reorder point is 669 units.
APPENDIX B
DATA STRUCTURE

1. OVERVIEW

There are seven interrelated files in the proposed system. The PROPERTY and ASL file, PROPERTY.DBF and ASL.DBF, are used in the inventory control management. They contain for each item its stock number, its attributes and its inventory control criteria.

The main files for transactions and report and statistics generation are the BATCH MASTER, and STOCKOUT files - BATCH.DBF, MASTER.DBF, STOCKOUT.DBF. The batch file is a depository of all the transactions arriving in one day. At the end of the day, transactions are processed and the demand is satisfied based on the transaction priority, priorities are determined by the unit's Standard Operating Procedure(SOP.) Transactions are moved to either the master or the stockout files.

These three files contain transaction specific data. As detail files, they define only those elements defining a transaction and do not include global or unchanging elements, such as the customer's description or stock item's nomenclature. Global elements are referenced by a unique identifier.

2. DATA FILES

a. Master

The elements in the master file are SN, CI, VN, REQNO, QTY, TYPE, UNITCOST, MISC, DATE, and POSTED. The SN field identifies the stock number of each item on the transaction and is the key to the items file. CI contains the customer number and is the key that link a transaction to the customer file which contains data on customer and supplier. SN and CI link the records to the other source files. VN contains the voucher number of the transaction, and REQNO is the requisition number.

The remaining fields -- QTY, UNITCOST, DATE, contain the quantity, unit price, and date of the transaction. The TYPE field contains the transaction type which is shown in chapter II, Fig 2.2 transaction procedure. Table 4 in this appendix includes a detail description of the contents of the various elements.
### TABLE 3
**DATA STRUCTURE**

<table>
<thead>
<tr>
<th>Database</th>
<th>Field name</th>
<th>Primary</th>
<th>Keys</th>
<th>Records</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Master</td>
<td>(SN, CI, VN, REQNO, QTY, TYPE, UNITCOST, MISC, DATA, POSTED)</td>
<td>SN</td>
<td>CI, TYPE</td>
<td>Record on all transactions.</td>
<td></td>
</tr>
<tr>
<td>Property</td>
<td>(SN, SERIALNO, NN, UNIT, ONHAND, UNITCOST, TVALE, ESSENCE, CLASS, DATE)</td>
<td>SN</td>
<td></td>
<td>Data on characteristics of stock item.</td>
<td></td>
</tr>
<tr>
<td>ASL</td>
<td>(SN, ROP, SL, ROOST, RCN, DATE)</td>
<td>SN</td>
<td></td>
<td>Data on transaction control measures.</td>
<td></td>
</tr>
<tr>
<td>Customer</td>
<td>(CI, CDESC, ADDRESS, ZIPCODE, FUND, EXPEND, PRIORITY, DATE)</td>
<td>CI</td>
<td></td>
<td>Data on customer's record.</td>
<td></td>
</tr>
<tr>
<td>Stockout</td>
<td>(SN, CI, VN, REQNO, TYPE, QTY, UNITCOST, DATE, MISC, POSTED)</td>
<td>SN</td>
<td>CI, TYPE</td>
<td>Records on all stockout transaction.</td>
<td></td>
</tr>
<tr>
<td>Batch</td>
<td>(SN, CI, VN, REQNO, TYPE, QTY, UNITCOST, DATE, MISC, POSTED)</td>
<td>SN</td>
<td>CI, VN, REQNO</td>
<td>Intermediate records on transactions.</td>
<td></td>
</tr>
<tr>
<td>Normal</td>
<td>(TZ, POFB, POFM, EOFZ)</td>
<td>no key</td>
<td></td>
<td>Table of normal distribution.</td>
<td></td>
</tr>
</tbody>
</table>

**b. Property file**

The property file contains all the information pertinent to each stock number. The first field, SN, is the key for accessing this file. The ONHAND and UNITCOST are quantity on hand and the unit cost respectively. For other fields refer to the Table 3.

**c. ASL file**

The ASL file contains all the information on the control measures for each item. The first field, SN, is the key for accessing this file. ROP field is the requisition objective defined in chapter two. ROP is the reorder point, whenever the ONHAND on property file reaches to this point, the system issues a request for issue to the logistics support command automatically. SL is safety level used in calculating the appropriate requisition objective.
d. Customer file

The customer file, CUSTOMER.DBF, whose structure is shown in Table 3, contains all the information on both customers and suppliers. The customer ID (CI) is the primary key for the customer file, allowing quick access to a particular customer record. The field CDESC contains the description of the CI. The two fields FUND and EXPEND, although not used by the system, are included in this file. Those fields provide the flexibility to extend the system and add fund accounting modules.

e. Stockout file

The stock-out file, STOCKOUT.DBF, contains the Stock Out items. The fields SN and CI link the file to the property file and customer file. The VN field defines the transaction.

f. Batch file

The batch file holds all transactions for the day. At the end of the day, transactions are processed and fulfilled based on their priority. As a result of the processing transactions are transferred to either the master or the Stock Out files. The field are the same as master file.

g. Normal file

The normal file, NORMAL.DBF, contains the values of the normal distribution, which provides the Z value corresponding to sigma. The field EOFZ contains the expected number of stockout with the probability of stockout.

3. INDEX FILES

The proposed system maintains a number of index files, each database file (except normal.dbf) uses at least one index as a primary index for data retrievals during execution of the program. In addition, some files have additional indexes used by programs in printing reports or internal operations.
<table>
<thead>
<tr>
<th>Field name</th>
<th>Type</th>
<th>Width</th>
<th>Dec</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>Address</td>
<td>Character</td>
<td>60</td>
<td>0</td>
<td>Customer</td>
</tr>
<tr>
<td>Cdesc</td>
<td>Character</td>
<td>30</td>
<td>0</td>
<td>Customer</td>
</tr>
<tr>
<td>Class</td>
<td>Character</td>
<td>4</td>
<td>0</td>
<td>Material</td>
</tr>
<tr>
<td>Date</td>
<td>Date</td>
<td>8</td>
<td>0</td>
<td>All databases</td>
</tr>
<tr>
<td>Essence</td>
<td>Logic</td>
<td>1</td>
<td>0</td>
<td>Property</td>
</tr>
<tr>
<td>Expend</td>
<td>Numeric</td>
<td>10</td>
<td>2</td>
<td>Customer</td>
</tr>
<tr>
<td>Fund</td>
<td>Numeric</td>
<td>10</td>
<td>2</td>
<td>Customer</td>
</tr>
<tr>
<td>Fundoh</td>
<td>Numeric</td>
<td>10</td>
<td>2</td>
<td>Customer</td>
</tr>
<tr>
<td>Misc</td>
<td>Logic</td>
<td>1</td>
<td>0</td>
<td>Master, Batch</td>
</tr>
<tr>
<td>NM</td>
<td>Character</td>
<td>16</td>
<td>0</td>
<td>Property</td>
</tr>
<tr>
<td>Onhand</td>
<td>Numeric</td>
<td>5</td>
<td>0</td>
<td>Property</td>
</tr>
<tr>
<td>Ost</td>
<td>Numeric</td>
<td>3</td>
<td>0</td>
<td>ASL</td>
</tr>
<tr>
<td>Outdate</td>
<td>Character</td>
<td>4</td>
<td>0</td>
<td>Stockout</td>
</tr>
<tr>
<td>Priority</td>
<td>Character</td>
<td>1</td>
<td>0</td>
<td>Property</td>
</tr>
<tr>
<td>Qty</td>
<td>Numeric</td>
<td>5</td>
<td>0</td>
<td>Master, Batch, Stockout</td>
</tr>
<tr>
<td>RCN</td>
<td>Numeric</td>
<td>4</td>
<td>0</td>
<td>ASL</td>
</tr>
<tr>
<td>Refilldate</td>
<td>Date</td>
<td>8</td>
<td>0</td>
<td>Stockout</td>
</tr>
<tr>
<td>ReqNO</td>
<td>Character</td>
<td>16</td>
<td>0</td>
<td>Batch, Stockout, Master</td>
</tr>
<tr>
<td>RO</td>
<td>Numeric</td>
<td>5</td>
<td>0</td>
<td>ASL</td>
</tr>
<tr>
<td>ROP</td>
<td>Numeric</td>
<td>4</td>
<td>0</td>
<td>ASL</td>
</tr>
<tr>
<td>Serialno</td>
<td>Character</td>
<td>10</td>
<td>0</td>
<td>Property</td>
</tr>
<tr>
<td>SL</td>
<td>Numeric</td>
<td>4</td>
<td>0</td>
<td>ASL</td>
</tr>
<tr>
<td>SN</td>
<td>Character</td>
<td>16</td>
<td>0</td>
<td>Master, ASL, Property, Stockout, Batch</td>
</tr>
<tr>
<td>Tvalue</td>
<td>Numeric</td>
<td>10</td>
<td>2</td>
<td>Property</td>
</tr>
<tr>
<td>Type</td>
<td>Character</td>
<td>2</td>
<td>0</td>
<td>Master, Stockout, Batch, Transaction type</td>
</tr>
</tbody>
</table>

* Fundoh numeric means the onhand amount of $.*
### TABLE 4
#### DATA DICTIONARY (CONT'D.)

<table>
<thead>
<tr>
<th>Unit</th>
<th>Character</th>
<th>4</th>
<th>0</th>
<th>Property</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Measureing unit of the item.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>(Ex. Cm, Kg, Each, Drum.....)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

| Unitcost | Numeric | 8 | 2 | Master, Property, |
|----------|---------|---|---| Stockout, Batch. |
|          | Unit price of each item. |
|          | (Ex. $20.00 for hammer) |

| VN       | Character | 16| 0 | Master, Batch |
|----------|-----------|---|---| Voucher number: |
|          | (Ex. 1500QM-0012-8309) |

| Zipcode  | Character | 6 | 0 | Customer |
|----------|-----------|---|---| Zipcode of a customer's address. |

### TABLE 5
#### INDEX FILES

<table>
<thead>
<tr>
<th>File</th>
<th>Index key</th>
<th>Index name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Property</td>
<td>SN</td>
<td>Snproper</td>
<td>Index by stock number</td>
</tr>
<tr>
<td>Customer</td>
<td>CI</td>
<td>Cicust</td>
<td>Index by customer's ID</td>
</tr>
<tr>
<td>ASL</td>
<td>SN</td>
<td>Inasl</td>
<td>Index by stock number</td>
</tr>
<tr>
<td>Batch</td>
<td>SN+CI</td>
<td>Sncibat</td>
<td>Index by stock number and Customer's ID (Used in transaction issue list)</td>
</tr>
<tr>
<td>Stockout</td>
<td>SN+CI</td>
<td>Stockout</td>
<td>Index by stock number and customer's ID (used in retrieve stockout item which is caused by customer)</td>
</tr>
<tr>
<td>Master</td>
<td>SN+CI</td>
<td>Sctmast</td>
<td>Index by stock number and Customer's ID and transaction type.</td>
</tr>
<tr>
<td></td>
<td>+TYPE</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>CI+SN</td>
<td>Csmast</td>
<td>Index by customer ID and stock number</td>
</tr>
</tbody>
</table>

53
APPENDIX C
SYSTEM STRUCTURE

1. OVERVIEW

The proposed system consists of four modules MANAGFL, TRANSACT, REPORTS and ANALYSIS performing the following four functions: file management, transaction processing, report generating, and analysis. There are a number of subroutines and performing functions common to one or more modules. The system structure is shown in Figure C.1. The menu system designed for this application corresponds to the gray portion on the figure. Appendix D (User's Manual) contains detailed instructions on how to use the system. This appendix will concentrate on describing the modules.

2. PMMAIN

PMMAIN is the normal starting point of the program and it displays the main menu and stores the last request and voucher numbers issued by the division. The menu is used to select one of the four main modules as shown in Figure C.2.

3. MANAGFL

This module is designed for file management and contains ten submodules and several screen formats. During the initial system loading, the user needs to enter data into Property, ASL, and Customer files using the ADDPROP, ADDASL, and ADDPROP submodules. During routine operations, the user may correct entered records in Property, ASL, and Customer files by using EDITPROP, EDITASL, and EDITCLST respectively. The user may correct transactions using EDITMAST and EDITBAT.

The flow chart in Figure C.3 illustrates the procedure of ADDPROP.prg. Since the file management procedures are closely related, only one flow chart is presented as an illustration.

The Manageq provides queries on files. The user may consult the present list of stock numbers by stock number, material class, or all stock items. An example is shown on the Figure C.4.
Figure C.1 Detailed system structure
### Inventory Management System

**1: Transaction**
- Routine transaction: Request for issue
- Receive item
- Query on transaction

**2: Manage file**
- Management of file:
  - Add
  - Edit (delete, change)
  - Query on files

**3: Report**
- Report Generation:
  - Transaction status report
  - Stockout report
  - Essential item stockout
  - OST report, Fund status

**4: Analysis**
- Analysis of:
  - Reorder point
  - Lead time
  - Requisition Objective
  - Safety Level

**Enter Selection (1 - 4, 5 to change date, or 0 to exit: )**

Today is 1/18/87 (2ULU:7322)

---

**Figure C.2** Pmmain screen.

### 4. TRANSACT

This module includes ten submodules corresponding to the type of transaction, and query on transaction. The overall data flow is shown in the Figure 4.3. This module is the main procedure and contains the analysis and report programs. The main menu for this module is shown in the Figure C.5.

**a. StockRD**

This submodule is designed for transaction type RD - request for issue to the division from the organizational unit. It stores input transactions, e.g. request for issue and turn-in, in a batch file. The submodule checks the validity of both the stock number and the Customer ID. If either is invalid, or the number is not included in the appropriate parent file, i.e. Property and Customer files, the program will prompt the user to update the parent files prior to accepting the entry. In other words no transactions is accepted without a valid SN and CI. The Figure C.6 illustrates the procedure.

**b. TRbatch and TRissue**

These submodules are the batch file processing procedures. They check whether the ONHAND quantity satisfies the total quantity requested in that day. If the ONHAND is less, transactions are satisfied based on their priority. The submodules generate a request for material when the ONHAND quantity reaches the reorder point.
Figure C.3 The flow chart of ADDPROP module
Query on Stock number

Options

1. Stock number
2. Class Enter Class Number: 7
3. All [0] for class 10
4. Return to main menu

Enter Option: 2

---

Page No. 1
11/16/87

Stock number Listing

<table>
<thead>
<tr>
<th>Stock number</th>
<th>Nomenclature</th>
<th>Price</th>
<th>Onhand</th>
<th>Requisition</th>
<th>Reorder</th>
<th>Safety</th>
<th>Objective</th>
<th>Point</th>
<th>Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>7520-00-281-5911</td>
<td>Basket, Waste</td>
<td>10.00</td>
<td>180</td>
<td>180</td>
<td>50</td>
<td>20</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7510-00-984-5787</td>
<td>Binder</td>
<td>12.00</td>
<td>93</td>
<td>190</td>
<td>10</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7920-00-141-5452</td>
<td>Handle, Wood</td>
<td>112.00</td>
<td>438</td>
<td>10</td>
<td>3</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7520-00-904-1268</td>
<td>Marker</td>
<td>2.00</td>
<td>230</td>
<td>290</td>
<td>100</td>
<td>30</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7510-00-022-8926</td>
<td>Pencil</td>
<td>1.00</td>
<td>520</td>
<td>7000</td>
<td>100</td>
<td>80</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7510-00-543-6792</td>
<td>Refill</td>
<td>0.20</td>
<td>1120</td>
<td>1800</td>
<td>900</td>
<td>45</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7930-00-141-5888</td>
<td>Wax, Floor</td>
<td>10.00</td>
<td>30</td>
<td>500</td>
<td>25</td>
<td>15</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7940-00-168-3366</td>
<td>Widget</td>
<td>120.00</td>
<td>23</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7777-77-777-7777</td>
<td>Test item</td>
<td>26.00</td>
<td>5085</td>
<td>500</td>
<td>280</td>
<td>140</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---

Figure C.4  Example of MANAGEQ module.

Finally they are responsible for aggregating all requests from a single customer into a one issue list. Figure C.7 shows their flowchart.

c. TRreqlst

Module TRREQLST is used for special events, that is, the manager may be requested to prepare for special events in the division such as training or unplanned construction. This module issues a warning to reflect that the input transaction is not a regular one.

d. TRstock

This module designed for type 'IL' transactions, ie issue from logistics support command. This module validates the customer or supplier ID and the item SN as one of the authorized supplier, initial supplies, and ASL item. A second check determines
whether the item received is a stock out item. If it is, the delayed issue list is processed, and all related recording is posted (STOCKOUT.DBF, MASTER.DBF). This process is shown in Figure C.8.

e. TURTOLC, TRTURNIN and TRTUNLST

These submodules accept items returned from units. Turn-in flow chart is shown in Figure C.9.

f. Query

This module generates transactions' reports. Queries such as how many transactions have occurred in given period for specified customer, stock number, material class, or all stock number, may be answered. Figure C.10 shows the screen menu for this submodule.
Figure C.6 TRSTOKRD (Request to division) module flow chart
Figure C.7 TRBATCH (batch process) module flow chart
Figure C.8 TRSTOCK (Issue from LSC) module flow chart
Figure C.9 TURNTULC (Turn-in to LSC) module flow chart
### Sales Tracking

**Options**

1. Detail Sales
2. Summary Sales
3. Detail Customer Sales
4. Summary Customer Sales
5. Return to main menu

**Enter Option:** 1

**Enter period for sales tracking (01/01/87 - 01/01/88)**

---

**Page No:** 1  
**11/16/87**

**Sales Transaction Tracking**

**Detail Sales**

<table>
<thead>
<tr>
<th>Customer ID</th>
<th>Quantity</th>
<th>Price</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Stock number 1111-11-111-111</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1500</td>
<td>7</td>
<td>10.00</td>
<td>70.00</td>
</tr>
<tr>
<td>1500</td>
<td>88</td>
<td>10.00</td>
<td>880.00</td>
</tr>
<tr>
<td>1509</td>
<td>1</td>
<td>10.00</td>
<td>10.00</td>
</tr>
<tr>
<td>1509</td>
<td>40</td>
<td>10.00</td>
<td>400.00</td>
</tr>
<tr>
<td>3333</td>
<td>95</td>
<td>10.00</td>
<td>950.00</td>
</tr>
<tr>
<td>5090</td>
<td>100</td>
<td>10.00</td>
<td>1000.00</td>
</tr>
<tr>
<td>5093</td>
<td>100</td>
<td>10.00</td>
<td>1000.00</td>
</tr>
<tr>
<td><strong>Subtotal</strong></td>
<td></td>
<td></td>
<td><strong>651</strong></td>
</tr>
<tr>
<td><strong>Subtotal</strong></td>
<td></td>
<td></td>
<td><strong>6510.00</strong></td>
</tr>
</tbody>
</table>

| **Stock number 2222-22-222-222** |          |       |       |
| 1500        | 97       | 10.00 | 970.00|
| 1500        | 97       | 10.00 | 970.00|
| 1504        | 100      | 2.00  | 200.00|
| 1507        | 10       | 2.00  | 20.00 |
| 1508        | 200      | 2.00  | 400.00|
| **Subtotal** |          |       | **504** |
| **Subtotal** |          |       | **2560.00** |

| **Stock number 3333-33-333-333** |          |       |       |
| 1500        | 25       | 0.00  | 0.00  |
| 1509        | 2        | 102.00| 204.00|
| 1509        | 2        | 102.00| 204.00|
| 5012        | 2        | 0.00  | 0.00  |
| 5014        | 1        | 102.00| 102.00|
| 5032        | 2        | 102.00| 204.00|
| **Subtotal** |          |       | **34** |
| **Subtotal** |          |       | **714.00** |

---

**Figure C.10** Example of Query on transaction module.
5. REPORTS

This module consists of RPTSR, RPOST, RPEIS, RPSOR for transaction status, order shipping time, essential item stockout, and stock out reports respectively. Each module creates temporary files and erases them at the end of session. The selection screen in this module is shown on the Figure C.11 followed by examples from the various reports.

<table>
<thead>
<tr>
<th>Type of Reports</th>
<th>Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>1: Transaction Status Report (DA form 3183)</td>
<td>Last TSR Report Date 11/12/87</td>
</tr>
<tr>
<td>2: Stockout Report (DA form 3184)</td>
<td>Today....11/16/87(7320)</td>
</tr>
<tr>
<td>3: OST Report (DA form 3185)</td>
<td>Biginning Date - Ending Date</td>
</tr>
<tr>
<td>4: Combat Essential Item Stockout Report (DA form 3186)</td>
<td>( / / - / / )</td>
</tr>
</tbody>
</table>

(Enter Selection from (1 - 4, or 0 to return) : :)

Figure C.11 Report module menu screen.

Refer to the Figure C.12, Figure C.13 and Figure C.14.
Hello
Figure C.13 RPOST (Order shipping time report) module flow chart
Figure C.14 RPSOR (Stockout report) module flow chart
6. ANALYSIS

This module consists of three procedures, ANLEADT, ANEOQ, and ANPROC and is used to calculate the average lead time of any stock item. The Average Lead Time is used in calculating safety level, reorder point, and finally economic order quantity. The module ANPROC has many procedures which contains zulu date, average, standard deviation, and variance. These procedures are called from the ANALYSIS module whenever needed. The data flow in this module is shown in chapter three limited analysis section. The detailed process is in the flow chart in Figure 4.5 The first screen of this module is shown on the Figure C.15

The first screen of this module is shown on the Figure C.15

<table>
<thead>
<tr>
<th>ANALYSIS of TRANSACTION</th>
<th>Today : 11/16/87</th>
</tr>
</thead>
<tbody>
<tr>
<td>THIS SYSTEM USES THE FIXED ORDER SIZE SYSTEM WITH PROBABILISTIC MODEL. THE FIXED ORDER SIZE SYSTEM IS COMPLETELY DEFINED BY THE ORDER QUANTITY [Q] AND REORDER POINT [B]. THE RISK OF STOCKOUT OCCURS AFTER REORDER POINT. TO GET [Q],[B], YOU HAVE TO DECIDE SERVICE LEVEL, OR STOCKOUT COST PER UNIT. THIS SYSTEM ASSUMED BACKORDER CASE WITH SERVICE LEVEL INSTEAD OF LOSTSALES CASE.</td>
<td></td>
</tr>
</tbody>
</table>

Enter stock number : -----

Enter time period you want to test( 01/01/87 - 12/31/87 )

Begin   End date

Leave BLANK any space to Exit

Figure C.15  Analysis module screen.
APPENDIX D
USER'S MANUAL

1. INTRODUCTION

The proposed system is for the manager who works at the ROK Army division or lower level logistics management. It is written in dBASEIII Plus programming language. Routine transaction handling, transaction tracking, reports generation, and limited analysis on the transaction records can be accomplished with this system.

The proposed system consists of three diskettes. One contains the software code developed in this thesis, the other two are dBASEIII plus system diskettes.

a. Requirements

The proposed system runs on the IBM PC family of computers, including the XT and AT, along with all true IBM compatibles. DBASEIII PLUS requires the following:

1. The dBASEIII PLUS program disks and manual.
2. An IBM PC, IBM XT, COMPAQ, or other 100 percent IBM PC-compatible computer with a monochrome or color monitor.
3. MS-DOS or PC-DOS
4. At least 256k of memory; 512k or more is suggested.
5. Two 360k floppy disk drives or one 360k drive and a hard disk drive. A hard disk is suggested.
6. A printer with at least 80-column capability is suggested.

[Ref. 8: p. 12]

b. Organization

The remainder of this manual is divided into two sections. Getting Started, describes the contents of the proposed system and how to install it. Working with the Proposed System, describes how to operate the system.

2. GETTING STARTED

The code of proposed system occupies one diskette only. Since the proposed system is not compiled, it needs the dBASEIII plus system diskettes. The proposed system assumes that dBASEIII plus is available in a subdirectory DBASEIII. So install Dbase III Plus in a subdirectory called DBASEIII, see Dbase III manual for instructions.
If your disk is missing any of the previous files, it will not work.

After confirming the existence of all the files, you are ready to install the system in your computer.

It is recommended that you install in a subdirectory called CONTROL in the DBASE III subdirectory. To create the subdirectory, ensure that you are in the DBASE III directory and type:

```
MD CONTROL
```

Change directory to 'CONTROL' by typing `cd dBASEIII CONTROL` and insert the proposed system diskette in drive A, and copy all files into C: using the following commands:

```
CD dBASEIII CONTROL
COPY A:*.* C:
CD
```

Now you are ready to start.
a. Configuration

Before installing the system, ensure that the config.sys file in your system contains the following commands:

```
BUFFERS = 15
FILES = 20
```

If your system does not contain a config.sys file, you may copy it from the Dbase III system diskette.

You also need a config.db file in your dBASEIII subdirectory with command 

```
TYPEHEAD = 20
```

If you do not have this file, copy it from the Dbase III system diskette. If you have config.db add the command and erase config.db file from the proposed system.

You can create either or both files using the DOS editor or the following commands:

```
Copy con config.db
......
TYPEHEAD = 20
......
```

You may want to get into the system directly from the bootup. If this is the case, add the following commands to your autoexec.bat file.

```
PATH = C:\DOS; dBASEIII
CD dBASEIII
DBASE
CD
```

Now you are ready to install the proposed system in your computer.

b. Summary of the proposed system disk.

The thesis system disk contains the following programs and files:

```
<table>
<thead>
<tr>
<th>Extention</th>
<th>DBF</th>
<th>PRG</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ASL</td>
<td>ANALYSIS</td>
</tr>
<tr>
<td></td>
<td>CUSTOMER</td>
<td>ANLEADT</td>
</tr>
<tr>
<td></td>
<td>NORMAL</td>
<td>TRREQLST</td>
</tr>
<tr>
<td></td>
<td>MASTER</td>
<td>RPTSR</td>
</tr>
<tr>
<td></td>
<td>STOCKOUT</td>
<td>ADDPROP</td>
</tr>
<tr>
<td></td>
<td>BATCH</td>
<td>EDITBAT</td>
</tr>
<tr>
<td></td>
<td>PROPERTY</td>
<td>MANAGEQ</td>
</tr>
</tbody>
</table>
```

```
|           | CONTINUE | ANPROC |
|           | PHMENU | PMAIN |
|           | TRSTOCK | TRANSACT |
|           | SETUP | TRACKING |
|           | ADDCUST | EDITPROP |
```
3. WORKING WITH THE PROPOSED SYSTEM

At the DBASE III prompt, type "DO PMAIN" to get the main menu in Figure D.1, the starting point of the system. At the main menu, you may execute one of five actions. Options 1 thru 4 will start one of the four program modules. The fifth, initiated by typing 5, allows you to change the system date.

![Inventory Management System](image)

Figure D.1 Main menu(PMAIN).

a. Routine transaction

You have reached this point by choosing option 1 from the main menu. This submodule allows to enter transactions and follows the same procedures as the division supply support action among logistics support command and organizational units.

There are nine options on the menu screen, see Figure D.2

1. Request for issue from a customer

This is the procedure to accept a request for issue from your customer and you reach this point by choosing option 1 from the Routine Transaction Menu. This screen is shown in Figure D.3

The program will check the validity of the customer and stock numbers. If either is invalid the program will not accept it and it will prompt you to check its validity or add it to the appropriate file.

Today is 11/18/87 (ZULU:7322)
Enter the customer and stock numbers and quantity required. The menu will prompt you for additional items, and the program will generate a request number.

For additional customers change the customer's code and follow the same procedure.

2. Issue to the customer

This program will generate an issue list for each customer. You have reached it by choosing option 2 from the transaction routines. This procedure will be executed once every day. The system will integrate all of the request for the day to get the allowance, and consult the availability of the item in stock. If an item is short to meet the requests, it will follow the priority to issue, then report the request for issue to the logistics support command.

You may get a print out if you so desire. Examples of the reports are shown in Figure D.4

3. Request for turn-in

This procedure handles requests for turn-in from a customer. You have reached this option by entering 3 from the Routine Transaction Menu. The screen is shown in Figure D.5, Enter the required elements and, when prompted, prepare the printer. An example of the output is presented below the screen.
Is this record correct? X

Figure D.3 Request for issue from customer screen.

4. Receive stock item from LSC
Select option 4 from the Routine Transaction Menu to reach this point. Enter the elements shown on the screen in Figure D.6. The possible output is below the figure.

This procedure is complex and it validates the customer and stock number, checks due-ins and due-outs, and delay issue of the item.

5. Direct request to LSC
Option 5, 6 will work for special request for issue to LSC. This procedure is not for routine transactions. A warning will be issued when you access these options.

6. Cancellation of request
Options 7, 8 will execute user's cancellation requests. Upon selection of one of these options you will receive the screen shown in Figure D.7 After you enter the data, you will be prompted to confirm the record the transaction will be cancel.
b. File management

It is important to enter the Property, ASL and Customer files data first. This will limit the interruptions due to missing CI and SN. To save the necessary data in advance or during the routine transaction job, choose option 2 from the main menu. It will display the screen shown in Figure D.8. From this menu, you can add customer and stock numbers to the Customer, Property, and ASL files. You can also edit records in these files as well as the Master and Batch Files.

There are nine options on this screen, these are

1. Addition

   (1) Customer. These programs add customer, supplier and item to the appropriate files. On selecting the number 1, the stock number shell will be blocked, and customer shell will await your entry. You must know the customer's code, usually the four digit common name of the unit. After entering the code of the customer, the system validates the record in customer file to prevent duplication. Once confirmed, the next screen which is named 'customer file' in Figure D.9 will appear. Enter the data and confirm that the record is correct. Your confirmation adds the record to the file.
**Request for turn-in**

<table>
<thead>
<tr>
<th>From</th>
<th>To</th>
<th>Type of transaction</th>
<th>Request No</th>
<th>Stock number</th>
<th>Description</th>
<th>Unit</th>
<th>Reusable?</th>
<th>Price</th>
<th>Quantity</th>
</tr>
</thead>
<tbody>
<tr>
<td>5010 Regiment 2 Bn</td>
<td>The 150 Infantry Division</td>
<td>(TD) Request for turn-in</td>
<td>5012S4-0011-7320</td>
<td>1111-11-111-1111</td>
<td>Test item1</td>
<td>Ea</td>
<td>Y</td>
<td>10.00</td>
<td>20</td>
</tr>
</tbody>
</table>

Is this record correct? X

Figure D.5 Request for turn-in from customer.

**Receive new stock from LSC**

<table>
<thead>
<tr>
<th>Receive from</th>
<th>Send to</th>
<th>Type of transaction</th>
<th>Voucher No</th>
<th>Request Number</th>
<th>Stock Number</th>
<th>Description</th>
<th>Unit</th>
<th>Quantity</th>
<th>Reusable?</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>The 3333 Logistics support Cmd</td>
<td>The 150 Infantry Division</td>
<td>Receive new stock from LSC</td>
<td>1500QM-0027-7320</td>
<td>- -</td>
<td>1111-11-111-1111</td>
<td>Test item1</td>
<td>EA</td>
<td>95</td>
<td>T</td>
<td>10.00</td>
</tr>
</tbody>
</table>

Is this record correct? X

Figure D.6 Receive stock item from LSC.

To add a property record in your property file, select option two. On selecting the option the customer shell will be blocked and stock number shell will be highlighted. After you have entered the stock number the system will validate your...
Cancel Request for issue item

Today ..11/16/87(7320)

Stock number : 2222-22-222-2222
Customer Code: 1500
Request number: 1500QM-0009-7305

Quantity : 97  Price : 10.00  Date :(11/0)/87

Is this record what you want to cancel?(Y/N) N

Figure D.7 Cancellation of request screen.

Management of files

Edit files
4. To change Customer (CI)
5. To change Property (SN)
6. To change ASL (SN)
7. To change Master (SN+REQNO)
8. To change Batch (SN+REQNO)
9. Query on files

Addition
1. To add Customer (CI)
2. To add Property (SN)
3. To add ASL (SN)

Information
Today is .....11/19/87
Stock number : XXXX-XX-XXX-XXXX
(Or/And)
Customer code :

[Enter Selection (1 - 9, or 0 to go to main menu): 1:]

Figure D.8 File management screen.

entry. If accurate, you'll see the next screen, in Figure D.10 Steps similar to customer addition are followed. However, since the ASL list is closely related to the property list.
is this record correct? y

figure d.9 customer addition screen.

the program will ask you whether the item is also an ASL item and will added to the ASL file if necessary.

is this record correct? x

figure d.10 property list addition screen.
Another method to add an ASL is to add it directly to the ASL file by choosing option 3. It is suggested that you use this method in the case of a newly allowed ASL item. The screen for this option is shown in Figure D.11

```
Authorized storage list file

<table>
<thead>
<tr>
<th>Stock number</th>
<th>1111-00-111-0001</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reorder Point</td>
<td>100</td>
</tr>
<tr>
<td>Safety Level</td>
<td>20</td>
</tr>
<tr>
<td>Requisition Objective</td>
<td>5000</td>
</tr>
<tr>
<td>Order Shipping Time</td>
<td>12 days</td>
</tr>
<tr>
<td>Resource control number</td>
<td>1122</td>
</tr>
</tbody>
</table>
```

Is this record correct? : x

Figure D.11 ASL list addition screen.

2. Editing or Deletion of an entry

There are five options in this process. Options 4, 5, 6 are equivalent to option 1, 2, 3 respectively. Option 7 allows you to change type TL records, other records' types are automatically saved through the transaction process. Option 8 allows edits of any type of transactions so use it carefully.

(1) Customer. From the screen 'management of file' in Figure D.8, select option 4. You will be asked to enter the customer's code that you want to change or delete. On entering the code the program retrieves the customer record from the customer file and displays the next screen in Figure D.12 Edit the screen and exit by pressing the CONTROL and END keys at the same time.

(2) Property and ASL. This procedure is almost same as the option 4. Take option 5 for property and 6 for ASL. The edit screen for the ASL list is shown in Figure D.13.
Figure D.12 Customer record editing screen.

Figure D.13 ASL list record editing screen.
(3) *Master file records.* This procedure edits the master file records. It requires you to enter the stock number, the customer's ID, and the date of the transaction. Its screen is shown in Figure D.14

```
Edit/Change Master file

Today........ 11/29/87

Stock Number: 3333-33-333-3333
Customer code: 1504  Action Type: RD
Request Number: 1500QM-0032-7333
                        1504S4-0001-7330  Customer's
Quantity: 10
Price: 102.5
Reusable?: T
Date: 11/29/87

(Insert mode): Ins
(Delete) Character: Del
Field: Y Record: U
(Record) Next: PgDn
Previous: PgUp
(Done/Save): End Abandon: Esc
```

Figure D.14 Master file record editing screen.

(4) *BATCH file records.* Sometimes you need to change records already entered in the ASL file through transaction routine. This procedure is more generous than option 7. You may enter the information you have, when you get the screen in Figure D.15, you will find out appropriate record by using the function keys which are given below the screen.
**Edit (change or delete) Batch file**

<table>
<thead>
<tr>
<th>Stock number</th>
<th>3333-33-333-3333</th>
</tr>
</thead>
<tbody>
<tr>
<td>Voucher number</td>
<td>1504S4-0001-7330</td>
</tr>
<tr>
<td>Customer Code</td>
<td>1504</td>
</tr>
<tr>
<td>Type of action</td>
<td>RD</td>
</tr>
<tr>
<td>Quantity</td>
<td>10</td>
</tr>
<tr>
<td>Date</td>
<td>11/29/87</td>
</tr>
<tr>
<td>Unitcost</td>
<td>$102.00</td>
</tr>
</tbody>
</table>

Figure D.15 Batch file record editing screen.

### c. Reports generation

Four types of reports are available in the system. Each report, other than combat essential item stock-out report, may be generated by stock number, by material class, and all. You can get both printed and screen reports. The menu screen is presented in Figure D.16.

1. **Transaction status report**

To get the transaction status report, select option 1 and enter appropriate time period. Select any option you want from the screen in Figure D.17. For example, if you want to get the transaction status reports on material class 2, select option 2. You will be asked to enter the material class, and the type of report, i.e., screen or print out. It is recommended that you postpone requesting the print out till you check the output on the screen.

Examples of each option are presented in Figure D.18.

2. **Other Reports**

Other reports can be obtained using the same procedure as the transaction status report. Select 2 for stock-out, 3 for order shipping time, 4 for combat essential item stock-out reports.
**RE**

**PORTS**

<table>
<thead>
<tr>
<th>Type of Reports</th>
<th>Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 : Transaction Status Report</td>
<td>Last TSR Report Date</td>
</tr>
<tr>
<td>(DA form 3183)</td>
<td>11/12/87</td>
</tr>
<tr>
<td>2 : Stockout Report</td>
<td>Today....11/16/87(7320)</td>
</tr>
<tr>
<td>(DA form 3184)</td>
<td></td>
</tr>
<tr>
<td>3 : OST Report</td>
<td></td>
</tr>
<tr>
<td>(DA form 3185)</td>
<td></td>
</tr>
<tr>
<td>4 : Combat Essential Item</td>
<td></td>
</tr>
<tr>
<td>Stockout Report</td>
<td></td>
</tr>
<tr>
<td>(DA form 3186)</td>
<td></td>
</tr>
</tbody>
</table>

*Enter Selection from (1 - 4, or 0 to return) : *

**Figure D.16**  Reports generation menu screen.

**Sales Tracking**

**Options**

1. Detail Sales
2. Summary Sales
3. Detail Customer Sales
4. Summary Customer Sales
5. Return to main menu

**Figure D.17**  Transaction status report option menu.

d. Analysis

From the main menu screen select option 4. When you get the screen in Figure D.19 enter the stock number you want to analyze and the time period.
### Transaction Status Report

**From:** The 150 Infantry Division  
**To:** The 3333 Logistics Support Command  
**Date:** 11/06/87

<table>
<thead>
<tr>
<th>No.</th>
<th>Stock number</th>
<th>Description</th>
<th>D/I beg</th>
<th>Req'st</th>
<th>Cancel</th>
<th>Rece'v</th>
<th>Turnin</th>
<th>D/Inow</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1111-11-111-1111</td>
<td>Test item1</td>
<td>0</td>
<td>950</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>950</td>
</tr>
<tr>
<td></td>
<td>Subtotal</td>
<td></td>
<td>0</td>
<td>950</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>950</td>
</tr>
<tr>
<td>1</td>
<td>2222-22-222-2222</td>
<td>Test item2</td>
<td>0</td>
<td>970</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>970</td>
</tr>
<tr>
<td></td>
<td>Subtotal</td>
<td></td>
<td>0</td>
<td>970</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>970</td>
</tr>
<tr>
<td>1</td>
<td>3333-33-333-3333</td>
<td>Test item3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>Subtotal</td>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>8888-88-888-8888</td>
<td>Test item4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>264</td>
<td>0</td>
<td>-264</td>
</tr>
<tr>
<td></td>
<td>Subtotal</td>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>264</td>
<td>0</td>
<td>-264</td>
</tr>
<tr>
<td>1</td>
<td>9999-99-999-9999</td>
<td>Test item5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>24</td>
<td>0</td>
<td>-24</td>
</tr>
<tr>
<td></td>
<td>Subtotal</td>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>24</td>
<td>0</td>
<td>-24</td>
</tr>
</tbody>
</table>

**Material Management NCO:**  
**Date:** ____/____/____  
**Material Management Officer:**  
**Date:** ____/____/____

### Stockout Report

**From:** The 3333 Logistics Support Command  
**To:** The 150 Infantry Division  
**Date:** 11/06/87

<table>
<thead>
<tr>
<th>No.</th>
<th>Stock number</th>
<th>Total 30days</th>
<th>30days item</th>
<th>60days item</th>
<th>90days item</th>
<th>120days item</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1111-11-111-11</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>Subtotal</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>2222-22-222-2222</td>
<td>370</td>
<td>740</td>
<td>370</td>
<td>740</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>Subtotal</td>
<td>370</td>
<td>740</td>
<td>370</td>
<td>740</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>5555-55-555-5555</td>
<td>1000</td>
<td>1000</td>
<td>1000</td>
<td>1000</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>Subtotal</td>
<td>1000</td>
<td>1000</td>
<td>1000</td>
<td>1000</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>8888-88-888-8888</td>
<td>100</td>
<td>2200</td>
<td>100</td>
<td>2200</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>Subtotal</td>
<td>100</td>
<td>2200</td>
<td>100</td>
<td>2200</td>
<td>0</td>
</tr>
</tbody>
</table>

**Material Management NCO:**  
**Date:** ____/____/____  
**Material Management Officer:**  
**Date:** ____/____/____

Figure D.18 Examples of Transaction status reports.
The time period will effect the output results. Choose longer period to increase the confidence in the results.

**ANALYSIS OF TRANSACTION**

Today: 11/16/87

This system uses the Fixed Order Size System with probabilistic model. The Fixed Order Size System is completely defined by the order quantity \( Q \) and reorder point \( R \). The risk of stockout occurs after reorder point. To get \( Q \), \( R \), you have to decide service level, or stockout cost per unit. This system assumed backorder case with service level instead of lost sales case.

Enter stock number: --

Enter time period you want to test: (01/01/87 - 12/31/87)

Begin    End date

Leave BLANK any space to Exit

Figure D.19 Analysis start-up screen.

You may try as many times as you want with different possible data. However you should select one method only. If you do not enter an element, default data will be assumed (i.e., service level 85% and holding cost rate 15% of the annual inventory). After entering data into the screen in Figure D.20, the program will display the results and will instruct you on how to save them. The result screen is shown in Figure D.20.
You selected stock number: 8883-88-888-8888
(Test item?, Unit:Es, Class:8)

Purchasing price(?P): 22 S/Es
Annual Demand(?R): 100 Es/year
Lead time(OST) in Month: 0 Month(s)

Ordering cost(?C): 0.00 S/order
Holding cost unit per year: 0.00 %
(Select one of these)
Stockout cost(If Known): 0.00 S/unit Select? N
Service Level in year: 0.0000 % Select? Y

Is this record right?(Y/N): X

Figure D.20 Variable entering for analysis.

e. Query

Two types of queries are available. One is transaction tracking, and the other queries on a customer or present status of a stock number.

For the management purposes the transaction records on a stock item or customer's records may be monitored. You may ask for detailed or summarized transaction record. Select option 9 from transaction menu screen. After the screen on Figure D.21 comes up, select any option.

For example:

If you want a customer's transaction detailed record, select option 1. You will be prompted for type of output, i.e., screen or print out.

If you need to know the present inventory status on a stock number, material class, or all items, select option 9 from the file management menu on Figure D.8. The next procedure is the same as the transaction tracking. The options are presented on Figure D.22
Sales Tracking

Options
1. Detail Sales
2. Summary Sales
3. Detail Customer Sales
4. Summary Customer Sales
5. Return to main menu

Enter Option: 1

Enter period for sales tracking (01/01/87 - 01/01/88)

Figure D.21  Transaction tracking.

Query on Stock number

Options
1. Stock number
2. Class  Enter Class Number: 7
3. All  [0] for class 10
4. Return to main menu

Enter Option: 2

Figure D.22  Query on stock number.
APPENDIX E
PROGRAM LISTING

MAIN MENU
a. PMAIN

***************************************************************************
***************************************************************************
***************************************************************************
* Module name....: PMAIN.prg
* Author........: Park, Taeyong
* Date........,: Aug 10, 1987
* Purpose.......: Main menu of Inventory Management System
* for Republic of Korea Army Infantry Division
* Called by.....: Modules called : PMENU.prg,MANAGEFL.prg,TRANSACTION.prg
*.................................................. REPORTS.prg,ANALYSIS.prg
* Variales used.: Public.: MCI holds customer identification code
*............................... STOCKN holds stock number which identifies item
*............................... TODAY holds current system date
*............................... CHECK holds condition
*............................... ZULU holds zulu date which is converted
*............................... REQNO1 holds request number which issued the last
*............................... VNI holds voucher number issued the last
*............................... MTYPE holds type of transaction
*............................... VAR holds the value of variance
*............................... STD holds the value of standard deviation
*............................... MLEADT holds the value of leadtime of any item
*............................... ZiDAY, Z2DAY, Z3DAY
***************************************************************************
***************************************************************************
***************************************************************************
* CLEAR ALL
*..................Close all open files and clear all variables.
CLOSE ALL
*..................Set working environment.
SET TALK OFF
SET BELL OFF
SET HEADING OFF
SET HELP OFF
SET MENU OFF
SET SAFETY OFF
SET STATUS OFF
*..........................This sets up the CRASH.TXT file which records all actions so that if the system crashes, the database can be recreated. This file will be deleted if the system terminate normally.
*..........................
*..........................SET ALT ON
*..........................Define the public variables which can be used any module.
Public MCI,STOCKN,CHECK,ZULU,REQNO1,VNI,MTYPE,VAR,STD,MLEADT
*..........................
*..........................Restore memorized variables from memory file.
*..........................Restore from MYFILE.mem
Store space(4) to MCI,ZULU,MTYPE
STOCKN = space(16)
CHECK = SPACE(1)
*..........................Create memory variable for today's date.
TODAY = date()
*..........................Convert current date into zulu date.
Set century on

89
ZDAY = DTOC(TODAY)
Store "1/1/"+substr(zday,7,4) to Z1DAY
Z2day = TODAY - ctod(Z1day)+1
Store substr(Z2DAY,10)+str(Z2DAY,7,3) to Z3DAY
Store substr(Z3DAY,1,4) to ZULU
Set century off
DO WHILE .T.
  CLEAR
  DO PIMENU
  DO WHILE .T.
    i=0
    DO WHILE i=0
      i=INKEY()
      @ 22,64 SAY ""
      IF UPPER(CHR(i))$"012345"
        EXIT
      ENDDO
      i=0
    ENDDO
    @ 22,64 SAY UPPER(CHR(i))
    IF .NOT. CHR(i)$"5"
      EXIT
    ENDDO
    @ 24,38 GET today
    READ
    Set century on
    ZDAY = DTOC(TODAY)
    Store "1/1/"+substr(zday,7,4) to Z1DAY
    Z2day = TODAY - ctod(Z1day)+1
    Store substr(Z2DAY,10)+str(Z2DAY,7,3) to Z3DAY
    Store substr(Z3DAY,1,4) to ZULU
    Set century off
    @ 24,38 SAY today
    @ 24,53 SAY ZULU
    ENDDO
  DO CASE
    Case CHR(I) = '1'
      DO TRANSACT
    Case CHR(I) = '2'
      DO MANAGEFL
    Case CHR(I) = '3'
      DO REPORTS
    Case CHR(I) = '4'
      DO ANALYSIS
    Case CHR(I) = '0'
      Release All like M*
      Release TODAY,1,CHECK,STOCKN
      Save to MYFILE.mem
      SET TALK ON
      SET BELL ON
      SET HEADING ON
      SET HELP ON
      SET MENU ON
      SET SAFETY ON
      SET STATUS ON
      CLEAR ALL
      EXIT
    OTHERWISE
      chr(7)
  ENDCASE
  ENDDO
  SET ALTE OFF
  CLOSE ALTE
  ERASE CRASH.txt
  CLOSE DATABASES
  CLEAR

90
When done, exit from the system

RETURN

---

b. PMMENU

*****************************************************************************
** Module name: PMMENU.prg
** Author: Park, Taeyong
** Date: AUG 10, 1987
** Purpose: Provide Menu screen for the PMAIN program
** Called by: PMAIN.prg
** Module called: None
** Variable used: Today: holds current system date
*****************************************************************************

@ 1,9 TO 3,69
@ 4,1 TO 23,77 DOUBLE
@ 6,3 TO 12,37
@ 6,4 TO 7,20 DOUBLE
@ 6,5 SAY $SPACE(15)
@ 6,61 TO 12,75
@ 5,4 TO 7,59 DOUBLE
@ 6,43 SAY $SPACE(16)
@ 14,3 TO 21,37
@ 13,4 TO 15,20 DOUBLE
@ 14,5 SAY $SPACE(15)
@ 14,41 TO 21,75
@ 13,42 TO 15,59 DOUBLE
@ 14,43 SAY $SPACE(16)
@ 5,2 SAY CHR(176)+CHR(176)
@ 6,2 SAY CHR(176)
@ 7,2 SAY CHR(176)
@ 8,2 SAY CHR(176)
@ 9,2 SAY CHR(176)
@ 10,2 SAY CHR(176)
@ 11,2 SAY CHR(176)
@ 12,2 SAY CHR(176)
@ 13,2 SAY CHR(176)+CHR(176)
@ 14,2 SAY CHR(176)
@ 15,2 SAY CHR(176)
@ 16,2 SAY CHR(176)
@ 17,2 SAY CHR(176)
@ 18,2 SAY CHR(176)
@ 19,2 SAY CHR(176)
@ 20,2 SAY CHR(176)
@ 21,2 SAY CHR(176)
@ 22,2 SAY REPLICATE(CHR(176),75)
@ 21,76 SAY CHR(176)
@ 20,76 SAY CHR(176)
@ 19,76 SAY CHR(176)
@ 18,76 SAY CHR(176)
@ 17,76 SAY CHR(176)
@ 16,76 SAY CHR(176)
@ 15,76 SAY CHR(176)
@ 14,76 SAY CHR(176)
@ 13,60 SAY REPLICATE(CHR(176),17)
@ 12,75 SAY CHR(176)
@ 11,76 SAY CHR(176)
@ 10,76 SAY CHR(176)
@ 9,76 SAY CHR(176)
@ 8,76 SAY CHR(176)
@ 7,76 SAY CHR(176)
@ 6,76 SAY CHR(176)
@ 5,60 SAY REPLICATE(CHR(176),17)
@ 5,21 SAY REPLICATE(CHR(176),21)
@ 6,38 SAY REPLICATE(CHR(176),3)
@ 7,38 SAY REPLICATE(CHR(176),3)
@8.38 SAY REPLICATE(CHR(176),3)
@9.38 SAY REPLICATE(CHR(176),3)
@10.38 SAY REPLICATE(CHR(176),3)
@11.38 SAY REPLICATE(CHR(176),3)
@12.38 SAY REPLICATE(CHR(176),3)
@13.21 SAY REPLICATE(CHR(176),21)
@14.38 SAY REPLICATE(CHR(176),3)
@15.38 SAY REPLICATE(CHR(176),3)
@16.38 SAY REPLICATE(CHR(176),3)
@17.38 SAY REPLICATE(CHR(176),3)
@18.38 SAY REPLICATE(CHR(176),3)
@19.38 SAY REPLICATE(CHR(176),3)
@20.38 SAY REPLICATE(CHR(176),3)
@21.38 SAY REPLICATE(CHR(176),3)
@2.12 SAY "Inventory Management System"
@6.6 SAY "1: Transaction"
@6.44 SAY "2: Manage file"
@14.6 SAY "3: Report"
@14.44 SAY "4: Analysis"
@9.44 SAY "Management of file:"
@10.45 SAY "Add"
@10.45 SAY "Edit (delete, change)"
@11.45 SAY "Query on files"
@8.6 SAY "Routine transaction:"
@9.7 SAY "Request for issue"
@10.7 SAY "Receive item"
@11.7 SAY "Query on transaction"
@16.6 SAY "Report Generation:"
@17.7 SAY "Transaction status report"
@18.7 SAY "Stockout report"
@19.7 SAY "Essential item stockout"
@20.7 SAY "OST report, Fund status"
@16.44 SAY "Analysis of"
@17.45 SAY "Reorder point"
@18.45 SAY "Lead time"
@19.45 SAY "Requisition Objective"
@20.45 SAY "Safety Level"
@22.8 SAY "Enter Selection (1 - 4, 5 to change date, or 0 to exit: |"
@24.29 SAY "Today is "+"+(ZULU: )""
@24.38 SAY TODAY
@24.53 SAY ZULU
STORE "" TO SELECT
@22.63 GET SELECT PICT "9"
RETURN

*---------------------------------* Eof: PMMENU.prg *---------------------------------*

92
2. ROUTINE TRANSACTION

a. TRANSACTION MENU

******************************************************************************

* Module name.....: TRANSACT.prg
* Author...........: Park, Taeyong
* Date............: Aug 20, 1987
* Purpose.........: Record all kind of transaction into the Batch, Master file.
* Called by......: FMAIN.prg
* Modules called: TRBATCH.prg, TRSCANCEL.prg, TRSTOCK.prg
* ................: TRREQUEST.prg, TRTRUNKIN.prg
* ................: TRUNLST.prg, TRUNLC.prg
* Variables used.: Public.: STOCKNI, NTYPE, MCUST, MCI, MREQNO
* ................: Local.: MCI1, MC12, MHOST, MTYPEA, MTITLE, MISCM, MDESC1, * MCDESC2

******************************************************************************

DO WHILE .T.
  Mci1 = space(4)
  Mci2 = space(4)
  Mhost = space(50)
  Mcust = space(50)
  Mtype = space(2)
  Mtypea = space(26)
  Stockn = space(16)
  Mcdesc1 = space(30)
  Mcdesc2 = space(30)
  Mreqno = space(16)
  Miscm = "T" CLEAR
  @ 1.15 TO 3.55 double
  @ 6.3 TO 21.38
  @ 5.4 TO 7.28
  @ 6.5 SAY SPACE(23)
  @ 6.41 TO 12.75
  @ 5.42 TO 7.59
  @ 6.43 SAY SPACE(16)
  @ 14.41 TO 21.75
  @ 13.42 TO 15.59
  @ 14.43 SAY SPACE(16)
  @ 2.17 SAY "Transaction routine"
  @ 6.6 SAY "Type of transaction"
  @ 6.44 SAY "Source"
  @ 14.44 SAY "Information"
  @ 8.6 SAY "1: Request for issue to Div"
  @ 9.6 SAY "2: Issue to customer from Div"
  @ 11.6 SAY "3: Request for turn-in to Div"
  @ 12.6 SAY "4: Receive from LSC"
  @ 14.6 SAY "5: Request for issue to LSC"
  @ 15.6 SAY "6: Request for turn-in to LSC"
  @ 17.6 SAY "7: Cancel RL to LSC"
  @ 18.6 SAY "8: Cancel RD to Div"
  @ 20.6 SAY "9: Query on transaction"
  @ 8.42 SAY "Stock number : "
  @ 10.42 SAY "Customer : "
  SET COLOR TO N/W
  @ 8.56 SAY STOCKN PICT "9999-9999-9999"
  @ 10.56 SAY MCI PICT "9999"
  SET COLOR TO
  @ 16.47 SAY "Today is ....."
  @ 16.60 SAY today
  @ 17.51 SAY "(Zulu : "+zulu")"
  @ 19.47 SAY "Last VN : "+vnl
@ 20,44 SAY "Last Regno : " + reqnol
@ 22,12 SAY "Enter selection from (1 - 9, or 0 to return) : ":
STORE "" TO SEL
@ 22,59 SAY SEL
*.........The following lines are for select loop.................*
i=0
DO WHILE i=0
  i=INKEY()
  @ 22,59 SAY ""
  IF UPPER(CHR(i))$"0123456789"
    @ 22,59 SAY CHR(1)
    EXIT
  ENDIF
  i=0
ENDDO
ENDDO
DO CASE
  Case CHR(I) = '4'
    MTYPE = "II"
    Mtitle = "Receive newstock from LSC"
    Mhost = "Receive from :"
    Mcust = "Send to :"
    Mtypea = Mtitle
    Mci2 = "1500"
    @ 8, 56 GET Stockn pict "9999-9999-9999-9999"
    @ 10,56 SAY Mci2 pict "XXXX"
    Read
    @ 8, 56 SAY Stockn
    @ 10,56 SAY Mci2
    DO TRSTOCK
  Case CHR(I) = '3'
    MTYPE = "TD"
    Mtitle = "Request for turn-in"
    Mhost = "Request from :"
    Mcust = "Send to :"
    Mtypea = Mtitle
    Mci2 = "1500"
    @ 8, 56 GET Stockn pict "9999-9999-9999-9999"
    @ 10,56 GET Mci2 pict "XXXX"
    Read
    @ 8, 56 SAY Stockn
    @ 10,56 SAY Mci2
    DO TRTURNIN
  Case CHR(I) = '1'
    MTYPE = "RD"
    Mci2 = "1500"
    @ 8, 56 GET Stockn pict "9999-9999-9999-9999"
    @ 10,56 GET Mci2 pict "XXXX"
    Read
    @ 8, 56 SAY Stockn
    @ 10,56 SAY Mci2
    DO TRSTORD
  Case CHR(I) = '2'
    MTYPE = "ID"
    @ 8, 56 SAY "XXXX-XX-XXX-XXXX"
    @ 10,56 SAY "XXXX"
    DO TRBATCH
  Case CHR(I) = '5'
    MTYPE = "RL"
    Clear
    @ 10,10 to 17,65
    @ 12,15 SAY "Your selection (RL) must be resulted from other"
    @ 13,15 SAY "type of transaction. It means this module"
    @ 14,15 SAY "is conducted automatically. However you may"
    @ 15,15 SAY "continue this if you eagerly want to do"
    @ 17,15 SAY "Do you want to continue(Y/N)? ";
    GET CHECK pict ";
    Read

94
If CHECK = "Y"
  DO TRREQQLST
Endif
Loop
Case CHR(I) = '7'
  MTYPE = "CL"
  @ 8, 56 GET Stockn pict "9999-9999-9999-9999"
  @ 10, 56 GET Mcil pict "XXXX"
  Read
  @ 8, 56 SAY Stockn
  @ 10, 56 SAY Mcil
  Do TRCANCEL
Case CHR(I) = '6'
  MTYPE = "TL"
  @ 8, 56 GET Stockn pict "9999-9999-9999-9999"
  @ 10, 56 SAY Mcil pict "XXXX"
  Read
  @ 8, 56 SAY Stockn
  @ 10, 56 SAY Mcil
  Do TURNTOLC
Case CHR(I) = '8'
  MTYPE = "CD"
  NCase CHR(I) = '9'
  HTYPE = "QUERY"
  Do Tracking
CASE I = 27 .OR. CHR(I) = '0'
  ? Chr(7)
  CLEAR
  EXIT
ENDCASE
ENDDO
RETURN

*-----------------------------------------------------------------*
Eof: Transact.prg *-----------------------------------------------------------------*

b. TRSTOCKRD

******************************************************************************
******************************************************************************
******************************************************************************
******************************************************************************
Module name....: TRSTOCKRD.prg
******************************************************************************
******************************************************************************
******************************************************************************
******************************************************************************
* Module name....: TRSTOCKRD.prg
* Author.........: Park, Taeyong
* Date...........: Aug 28, 1987
* Purpose........: Record request for issue transaction from customer into batch file.
* Called by.......: TRANSACTION.prg
* Modules called :
* Variables used..:
  * Public.:
  * Local.:
******************************************************************************
******************************************************************************
******************************************************************************
******************************************************************************
Select A
Use BATCH index SNCIBAT, CIBAT
Select B
Use PROPERTY index SNPROPER
Select C
Use ASL index INASL
Select D
Use CUSTOMER index CICUST
Select B
Store "X" to batcheck, checks
Xcheck = .T.
Do while Xcheck
  Clear
@ 10.15 to 14.65
@ 11.17 Say " Customer Code : " Get Mcil Pict "9999"
@ 13.17 Say " Stock Number : " Get Stockn pict "9999-99-999-9999"
Read
If Mcil = " .OR. stockn=" 
Exit
Endif
Select D
Seek mcil
If .NOT. found()
  @ 11.16 clear to 13.64
  @ 11.17 Say "Not found " +Mcil+ " code customer "
  @ 13.17 Say "Do you want to add this as customer?(Y/N) ";
  Get checks pict !!
  Read
    If checks = "Y"
      Mci=Mcil
      Do addcust
      Store cdesc to Mcdescl
      Loop
    Else
      Loop
    Endif
  Else
    Store cdesc to Mcdescl
    Select B
    StockN = Upper(StockN)
    Seek StockN
    If .NOT. found()
      Clear
      Store "X" to checks
      @ 13.10 say stockn + " is Not found in your property book"
      @ 15.20 Say " Is this the INITIAL demand ? " get checks
      Read
      checks = upper(checks)
      If checks = "Y"
        Store 0 to Mcost
      Else
        Clear
        @ 15, 20 say "Check stock number and try again !"
        Wait
        Loop
      Else
        Endif
    Else
      mcost = unitcost
    Endif
  Endif
Endif
Select A
Append Blank
Replace CI with MCII, Type with Mtype, SN with StockN, Misc with .T.
Replace Unitcost with Mcost, Date with today, Reqno with Mreqno
Do while .T.
  Set format to batch
  Read
  If batchcheck = "Y"
    Set format to
    Replace REQNO with mreqno
    Stockn = space(16)
    Exit
  Endif
Enddo
Enddo
Close Databases
Release all
Return
*----------------------------------------* Eof TRSTOKRD.prg *----------------------------------------*
c. TRBATCH

*****************************************************
********************************** TRBATCH.PRG **************
********************************** *****************************************************
* Module name.....: TRBATCH.prg *
* Author...........: Park, Taeyong *
* Date............: Aug 28. 1987 *
* Purpose..........: This is the module for performing the *
* BATCH file which contains the records of routine transactions *
* Called by.......: TRANSACT.prg *
* Modules called : ISSUE.prg *
* Databases used.: MASTER, PROPERTY, ASL, BATCH, TEMPBAT *
* Variables used: *
* Public.: REQNO1, ZULU *
* Local..: MTY holds unit quantity of a given stock item. *
* MUNITC holds unit price of a given item. *
* STOCKN holds stock number *
* LNECTR holds the number of the line written so far *
* N holds how many iteration has happened *
*****************************************************

Select A
Use MASTER index SCTMAST
Select B
Use PROPERTY index SNPROPERTY
Select C
Use ASL index INASL
Select B
Set relation to SN into C
Select A
Set relation to SN into B
Select E
Use Batch index SNCIBAT, CIBAT
*.... Create a temporary file for stock number and type of transaction.*
*.... is same as given condition. ...........................................
Total on SN to TEMPBAT for TYPE="RD"
Use TEMPBAT
Go top
*.... If nothing has happened the day, give information and return.....*
If Eof()
  @ 10,14 Clear to 14,65
  @ 10,14 to 14,65
  @ 11,17 Say "Your Batch file is empty -> Go and take rest !"
  @ 13,19 Say "Press Any key to go main menu"
  Wait " "
  Release all
  Close databases
  Erase TEMPBAT.dbf
  Return
Endif
Index on SN to TEMPBAT
Use TEMPBAT index TEMPBAT
Set relation to SN into PROPERTY
Store "X" to mprint.checks
Store 1 to PAGECTR, LNECTR, N
Store 0 to TOTAL, MTOTAL, Z
Go top
To while .NOT. Eof()
  Clear
  @ 10,14 Clear to 14,65
  @ 10,14 to 14,65
  @ 11,27 say "Wait while it is cession "
  Store SN to STOCKN
  Store QTY to NQTY
Store B->nm to Hnm
Store B->unit to Munit
Store B->unitcost to MUNITC
Store B->ONHAND to MONHAND
Store MONHAND - NQTY to XONHAND
If XONHAND < C->ROP
  *.......... Check whether the master file already has an order
  *.......... of the item or not, if yes, check the amount so that
  *.......... can do action necessary.
  Select A
  Seek STOCKN + "1500RL"
  If found() .AND. .NOT. POSTED
    Store QTY + XONHAND to MONHAND
  Else
    Store XONHAND to MONHAND
  Endif
Select E
Endif
If MONHAND < C->ROP
  If mprint = "X"
    Do while .T.
      @ 10.14 clear to 14.65
      @ 10.14 to 14.65
      @ 11.27 say "Ready printer? " get MPRINT pict "!
      Read
      *.Set printer ready to print out, and console off so that
      *............... can be being conducted without disturbance.
      If mprint = "y"
        Set print on
        @ 13.20 say "Wait while printing............"
        Set console off
        Exit
      Endif
    Enddo
Endif
If LINCTR=40*(N-1)+ 1
If LINCTR = 1
  Store Val(Substr(REQNO1,8,4)) to MREQNO
  MREQNO = 100000+MREQNO+1
  REQNOL= "1500QM"+-++substr(str(MREQNO),7,4)+"-"+ZULU
Endif
?? 'REQUEST FOR ISSUE'
Page ' + str(pagectr,2)
?? 'Date : '+Dtoc(date())
?? 'To : The 3333 Logistics Support Command' + Mcdescl
?? 'From : The 150 Infantry Division ' Request No : '+Reqnol
?? 'No Stock number Description ' ?? 'Unit Quantity Price Total'
Store pagectr + 1 to pagectr
Endif
Store (C->ROP)-MONHAND to MQTY
Store MQTY * MUNITC to MTOtal
Store TOTAL + MTOtal to TOTAL
Store linectr to Z
?? Str(2,4)+"+sn+"+rtrim(mm)+space(15 - LEN(rtrim(mm)))+"+munit+"+str(mqty)+str(munitc)
?? str(munitc*mqty)
linectr = linectr + 1
*......... Record the action on the master file.
Select A
Append Blank
Replace SN with StockN,Ci with "1500"
Replace Reqno with reqnol,Qty with MQty,Type with "RL"
Replace Unitcost with munitc, Misc with .T., Date with today
Replace Posted with .F.

Select E
If linectr=40*N + 1
   ?
   ?
   ?
   ?;'Material Management NCO : ___________________'
   ?;'Date : __/____/____
   ?;'Material Management Officer : ________________'
   ?;'Date : __/____/____
Eject
N = N + 1
Endif
End Select E
Skip
Enddo

*.......................... Print only when heading printed
If linectr > 1
   ?
   ?
   ?
   ?
   ?;'Material Management NCO : ___________________'
   ?;'Date : __/____/____
   ?;'Material Management Officer : ________________'
   ?;'Date : __/____/____
Endif
End Select E
Skip

*.......................... Reset printer off and console on so that make it
   *possible to communicate with the user.
   *Erase the temporary file which is no more useable.
   Close databases
   Do TRISSUE
   Release all
   Close databases
   Return

*d. TRISSUE

*******************************************************************************
*                        TRISSE.UE.PRG                                      *
*******************************************************************************
* Module name....: TRISSE.UE.PRG                                             *
* Author..........: Park, Taeyong                                               *
* Date............: Sept 2, 1987                                               *
* Purpose.........: This is the issue module for generating                     *
*                  Receipt to the customer correspondinig                      *
*                  to request for issue.                                        *
* Called by......: Batch.prg                                                   *
* Modules called : none                                                        *
* Variales used...: Total,Mtotal,Z,Mxpend,Mfundoh                               *
* Databases used.: MASTER, PROPERTY, CUSTOMER, BATCH, STOCKOUT                *
*******************************************************************************

99
Select A
Use MASTER index SCTMAST
Select B
Use PROPERTY index SNPROPER
Select C
Use STOCKOUT index STOCKOUT
Select D
Use CUSTOMER index CICUST
Select A
Set relation to SN into B
Select E
Use BATCH index CIBAT,SNCIBAT
Set relation to SN into B
If Eof()
  Clear
  @ 15,25 SAY "Lucky today! Go and take a rest"
  ? Chr(7)
  X=0
  Do while x<35
    X=X+1
  Enddo
  Close databases
  Return
Endif
Store "X" to Mprint
Do while .T.
  @ 10,14 clear to 14,65
  @ 10,14 to 14,65
  @ 11,25 Say 'Ready Printer ? ' Get mprint pict ''!
  Read
  @ 11,42 Say mprint
  If Mprint='Y'
    Set print on
    @ 13,20 Say "Printing....."
    Set console off
    Store 0 to total,mtotal,Z,Mfundoh,Mexpend
    Store 1 to pagectr,linectr,N
    Exit
  Else
    Clear
    Set color to N/W
    @ 13,20 Say "You Pressed "+mprint
    Set color to
    @ 15,20 Say "Do you want to finish the job(Y/N)?" Get check
    Read
    If check = 'Y'
      close databases
      Return
    Else
      Store "X" to Mprint
    Loop
  Endif
Enddo
Go top
Do while .NOT. eof()
  If delete()
    Skip
  Endif
  Store ci to mci
  Select D
  Seek mci
  Store Expended to Mexpend
  Store Fundoh to Mfundoh
  Store cdesc to mcdesc1
  Select E
  Do while Type="RD" .AND. ci = mci

100
If delete()
  Skip
Loop
Endif
Store type to mtype
Store b->ununitcost to mcost
Store b->onhand to fonhand
Store qty to Egty
Store fonhand-Egty to fonhand
Store reqno to mreqno
Store sn to stockn
If linectr=40*(N-1)+1
  If linectr = 1
    Store Val(Substr(vnl,8,4)) to Mvn
    Mvn=100000+Mvn+1
    Vnl="1500QM"+-"Substr(str(Mvn),7,4)+"zulu
  Endif
  ??
  ??
  ??
  ??
  ISSUE LIST'
  ??
  ??
  Date : '+Dtoc(date())
  ??
  ??
  From : '+ 'The 150 Infantry Division'
  ??
  ??
  To : '+ mdescl
  ??
  ??
  Vocher No:'+vnl
  ??
  ??
  No Stock number Description
  ??
  Onhand Unit Quant y Price Total'
  store pagectr+ 1 to pagectr
Endif
If fonhand < 0
  ? chr(7)
  clear
  @ 15,20 Say "You have stockout item "+stockn
  @ 17,20 Say "This will be added to stockout file"
  num = 0
  Do while num<30
    num=num+1
  Enddo
  Store linectr to Z
  ? Str(Z,2)+' '+sn+' '+rtrim(b->nm)+space(15 -LEN(rtrim(b->nm)))+'
  ?? '+'SORRY! Delay delivery'
  Linectr=linectr+1
Select C
Append Blank
Replace sn with stockn,ci with mci,reqno with mreqno
Replace qty with eqty,unitcost with mcost
Replace outdate with today,posted with .F.,Type with mtype
Select E
Delete
Loop
Endif
If fonhand >=0
  Store linectr to Z
  ? Str(Z,2)+' '+sn+' '+rtrim(b->nm)+space(15 -LEN(rtrim(b->nm)))+
  ?? '+'+str(b->onhand)+' '+ b->unit
  ?? str(qty)+str(b->unitcost)+str(b->unitcost*qty)
  linectr = linectr + 1
  Store b->unitcost*qty to mtotal
  Store total+mtotal to total
Select A
Append blank
Replace sn with stockn,ci with mci,vn with vnl,
reqno with mreqno,type with mtype
Replace qty with eqty
Replace unitcost with mcost,Misc with .T.;
Date with today,Posted with .T.
Select B
Seek stockn
Replace onhand with fonhand, tvalue with fonhand*unitcost
Select E
Delete
Skip
Endif
If linectr=40*N+1
???? Grand Total ' + str(total)
???? 'Material Management NCO : 
???? 'Date : ______/____/
???? 'Material Management Officer : 
Eject
N = N + 1
Endif
Enddo
If linectr>1 .AND. .NOT. linectr=40*N+1
???? Grand Total :$' + str(total)
???? 'Material Management NCO : 
???? 'Date : ______/____/
???? 'Material Management Officer : 
Store 1 to pagectr, linectr, N
store 0 to total, mtotal
Eject
Endif
Select D
Seek mci
Replace Fundoh with Mfundoh-mtotal
Replace Expend with Mexpend+mtotal
Select E
Enddo
Select E
Pack
Store space(4) to mci
Release all
Set print off
Set console on
*--------------------------* Eof TRISSUE.prg *--------------------------*

c. TRREEQLST
Select A
  Use MASTER index SCTMAST
Select B
  Use PROPERTY index SNPROPER
Select D
  Use CUSTOMER index CICUST
Select A
  Store "X" to CHECK
  Store 0 to MQTY, TOTAL, MTOTAL
  Mreqno = 100000 + Mreqno + 1
  Rqno = "1500QM" + " + substr(str(Mreqno), 7, 4) + " + zulu
Do while .T.
  Clear
  @ 10, 15 to 14, 65
  @ 13, 17 Say "Customer Code : " Get Mci1 pict "9999"
  @ 13, 17 Say "Stock Number : " Get Stockn pict "9999-99-999-9999"
  @ 15, 25 Say "Leave blank any space to exit."
  Read
  Select D
  Seek Mci1
  Do Case
    Case Mci1 = " "
      Exit
    Case .NOT. found()
      @ 11, 16 clear to 13, 64
      @ 11, 17 Say "Not found " + Mci1 + " code customer " ;
      Get check pict "1"
      Read
      If check = "Y"
        Do addcust
          Store "X" to CHECK
          Store cdesc to Mcdescl
        Else
          Loop
        Endif
      Case found()
      Store cdesc to Mcdescl
    EndCase
  Select B
  StockN = Upper(StockN)
  Seek StockN
  Do case
    Case StockN = " "
      Exit
    Case .NOT. found()
      Clear
      @ 13, 10 say StockN + " is Not found in your property book"
      @ 15, 20 Say "Is this the INITIAL demand? " GET CHECK Pict "1"
      Read
      If check = "Y"
        Store 0 to Mcost
        Store "X" to CHECK
      Else
        Clear
        @ 15, 20 say "Check stock number and try again !"
        Wait
        Loop
      Endif
    Case found()
      Store UNITCOST to MCOST
      Store ONHAND to MONHAND
      Store NM to MMN
      Store UNIT to MUNIT
    Endcase
Select A
Append Blank
Replace C1 with MCI1, Type with Mtype, SN with StockN, Misc with .T.
Replace Unitcost with Mcost, Date with today, Reqno with Reqno1
Do while .T.
  Clear
  @ 2,  9 SAY "Request for Issue"
  @ 3, 43 SAY "Today is...
  @ 3, 56 SAY Today
  @ 5,  7 SAY "To: " + mcdescl
  @ 6,  7 SAY "From: The 1500 Infantry Division"
  @ 8,  6 SAY "Type of transaction:"
  @ 9, 54 SAY "Request Number:" 
  @ 9, 50 GET REQNO1 PICTURE "99991-9999-9999"
  @ 12, 19 SAY "Stock Number:" 
  @ 12, 32 GET MASTER->SN PICTURE "9999-9999-9999"
  @ 14, 19 SAY "Description :
  @ 14, 32 GET MNM
  @ 16, 34 SAY "Unit :
  @ 16, 41 GET MUNIT FUNCTION "!AAA" PICTURE "XXXX"
  @ 18, 19 SAY "Quantity :"
  @ 18, 32 GET M_QTY PICT "99999" 
  @ 18, 50 SAY "Reusable? :"
  @ 18, 62 GET MASTER->Misc
  @ 20, 19 SAY "Price :"
  @ 20, 32 GET M_COST PICT "9999.99"
  @ 23, 30 SAY "Is this record correct? " GET CHECK pict "!
  @ 4,  4 TO 21,  7
  @ 1,  6 TO 3, 40
  @ 7,  5 TO 7, 70
  @ 10,  5 TO 10, 70
  @ 8, 32 TO 9, 32

Read
If CHECK = "Y"
  Replace QTY with Mqty
  Store "X" to CHECK
  Exit
Endif
Enddo
@ 10,12 clear to 14,63
@ 10,12 to 14,63
@ 11,23 say "Ready printer? " GET CHECK pict "!
Read
If check = "Y"
  Set print on
  @ 13,20 SAY "Wait while printing........."
  Store "X" to CHECK
  Set console off
Endif
Store 1 to pagectr, linectr, N
Store 0 to total, mtotal, Z
If linectr<1
  ?
  ?
  ?
  ?
  ?
  To : ' + mcdescl
  From : The 150 Infantry Division'
  Request No: ' + Reqno1
  No Stock number Description 
  'Unit Quantity price Total'
Endif
Store (Mqty*Mcost) to Mtotal
Store linectr to Z
? Str(Z,4) + ' +sn+' + rtrim(Mnm)+ space(15 - LEN(rtrim(Mnm)))
?? ' + Munit+ ' +str(mqty) + str(Mcost)

104
Mcost * mty
Store Linectr+1 to linectr
Store Mcost * qty to mtotal
Store total + mtotal to total
@ 15, 10 clear to 17, 65
@ 16, 15 Say "Do you have more to request (Y/N)? ", Get Check Pict!
Read
If Check = "Y"
    Store "X" to CHECK
    Store 0 to Mgtys
    Store space (16) to Stockn
    Loop
Endif

?? Grand Total : $' + str(total)
??
?? 'Material Management NCO : ________________
?? 'Date : __/__/______
??
?? 'Material Management Officer : ________________
?? 'Date : __/__/______
Eject
Store space (16) to Stockn
Enddo
Clear
Set print off
Set console on
Close databases
Release all
Return

*-----------------------------* Eof TRREQ1ST.prg *-----------------------------*

TRSTOCK

* Module name.....: TRSTOCK.prg
* Author...........: Park, Taeyong
* Date............: AUG 18, 1987
* Purpose.........: Record item received from higher command
* on the master file. If needed add to the
* ASL or PROPERTY file
* Called by........: TRANSACT.prg
* Modules called : None
* Variables used ..:
*   Public: Vnl
*   Local: Docheck, check, fcheck, Dcheck, mqtys, smqtys...
* Databases used.: MASTER, PROPERTY, ASL, CUSTOMER
*-----------------------------* TRSTOCK.prg *-----------------------------*
Select A
Use Master INDEX SCTMAST
Select B
Use PROPERTY index SNPROPER
Select C
Use ASL index INASL
Select D
Use CUSTOMER index CICUST
store 1 to M
Store "X" to check, fcheck, dcheck
store 0 to mqtys, smqtys, mqty, mlqty, Tqty
Store Val (Substr (Vnl, 8, 4)) to mvnl
mvnl = 100000 + mvnl + 1

105
Vnl = "15000M-"+Substr(str(vnl),7,4)+"-"+Zulu
Docheck = .T.
Do while DOCHECK
  Clear
  *......Check Your customer, or add him
  Do while .T.
    @ 10,15 to 14,65
    @ 11,17 Say " Customer Code : " get mcil pict "9999"
    @ 13,17 Say " Stock Number : " get stockn pict;
    "9999-99-99-9999"
    @ 15,20 Say "Leave BLANK stock number to exit"
    Read
    Seek mcil
    If found()
      Mcdescl = cdesc
      Mcdesc2 = "The 150 Infantry Division"
      Exit
    Else
      @ 11,16 clear to 13,64
      @ 11,21 Say "Not found "+mcil+- code customer"
      @ 13,21 Say "Do you want to add the customer? " get;
      check pict "!
      Read
      If check = "Y"
        mcil = mcil
        Do addcust
        Store cdesc to mcdesc1
      Endif
    Endif
  Enddo
  *.....Check the new item initial or requested one.
  Select B
  StockN = Upper(StockN)
  Seek StockN
  Do case
    Case StockN = ". "
      Docheck = .F.
      Loop
    Case .NOT. found()
      @ 10,15 to 14,65
      @ 11,21 say " Not found in your property book"
      @ 13,21 Say " Is this the INITIAL supply? " get;
      check pict "!
      Read
      If check = "Y"
        Append blank
        Replace SN with StockN,DATE with Today
        Store 0 to unitcost,onhand
        Set format to PROP
        Read
        If fcheck = "Y"
          Set format to
          Store unitcost to mcost
          Store onhand to monhand
          Replace TVALUE with (mcost*monhand)
        Endif
      @ 10,15 clear to 14,65
      @ 10,15 to 14,65
      @ 12,17 Say "Is This "+stockn +;" ASL item? " get check pict "!
      Read
      If check = "Y"
        Select C
        Seek stockN
        If found()
          @ 11,14 clear to 13,64
          @ 11,17 say SN+" is Already exist!"
@ 13,17 Say "Do you want to change?" get check Pict !
Read
If check = "Y"
Replace SN with StockN
Replace Date with Date()
Do while .T.
Set format to ASL
Read
dcheck = upper(dcheck)
IF dcheck = "Y"
Close format
Exit
Endif
Enddo
Else
Append blank
Replace SN with StockN
Replace date with date()
Do while .T.
Set format to ASL
Read
IF dcheck = "Y"
Close format
Exit
Endif
Enddo
Endif
Else
@ 10,15 clear to 14,65
@ 10,15 to 14,65
@ 12,25 Say "You did good job. Now try again!"
@ 13,25 Say "Press Y to continue" get;
check Pict !
Read
If check = "Y"
Loop
Endif
Else
@ 10,15 clear to 12,65
@ 10,15 to 14,65
@ 11,25 Say "Sorry no way to go, Try again!"
@ 13,25 Say "Press Y to continue" get;
Check picle!.
Read
Loop
Endif
Case found()
mcost = unitcost
monhand = onhand
Endcase
Select A
Store stockn + "1500" + "RL" to mstockn
Sum QTY for SN=stockn .AND. type="RL" .AND. .NOT. posted to Tqty
Store "X" to zcheck
Do while .T.
Append blank
Replace CI with MCI, Type with Mtype
Replace SN with StockN,qty with Tqty
Replace Unitcost with Mcost
Replace Date with today,misc with .T.,posted with .T.
Replace vn with vnl
Set format to mast
Read
zcheck = upper(zcheck)
If zcheck = "Y"
Close format
Store QTY to M1QTY
Store M1qty to MQTY
Exit
Endif
Enddo
Go top
Seek mstockn
*........If requested item, post it
If found()
  Do while .NOT. Eof() .AND. type="RL" .AND. SN=STOCKN
    If posted
      skip
    loop
  Endif
  Store REQNO to MREQNO
  Store DATE to MDATE
  Store UNITCOST to M1COST
  Store QTY to CQTY
  Do case
    Case M1QTY-CQTY < 0
      Replace vn with vnl.posted with .T.
      Replace QTY with M1QTY
      Append Blank
      Replace sn with stockn
      Replace reqno with mreqno
      Replace date with mdate
      Replace unitcost with m1cost
      Replace qty with cqty-m1qty
      Replace posted with .F.
      Replace ci with "1500",type with "RL"
      Replace misc with .T.
      Exit
    Case M1QTY-CQTY>=0
      Replace vn with vnl.posted in .T.
      Store M1QTY-CQTY to H1QTY
      If H1QTY=Q
        Exit
      Endif
  Endcase
  Skip
Enddo
Select B
Seek STOCKN
Replace ONHAND with MONHAND+MQTY
Replace TVALUE with (MONHAND+MQTY)*MCOST
Endif
*. .Check If it a stockout refill item.
Select F
Use STOCKOUT index STOCKOUT
Set relation to sn into property
Seek stockn
If Eof() .OR. Bof()
  Stockn = space(16)
  Select D
  Loop
Endif
Store 'X' to mprint
Do while .T.
  @ 10,15 clear to 14,65
  @ 10,15 to 14,65
  @ 11,17 Say " You have a delayed delivery."
  @ 13,17 Say 'Ready Printer ?' get mprint pict "!"
  Read
  If mprint = 'Y'
    Set print on
  @ 13,17 Say "Printing............"
Endif
Enddo
Do while .NOT. Eof() .AND. SN=STOCKN
If POSTED
  skip
Endif
Store REQNO to MREQNO
Store CI to MCI
Store OUTDATE to MDATE
Select D
Seek MCI
Store cdesc to mdesc1
Select F
?
?
?
?
?
?
?
?

ISSUE LIST'
Page :1'
?
?' From : ' + 'The 150 Infantry Division'
?
?' Date : '+DTOC(date())
?
?' To : ' + mdesc1
?
?' Request Number : ' + mreqno
?
?' No Stock number Description ' 
?? ' Onhand Unit Quantity price ' 
?? ' Total' 
? ' '+sn' '+trim(b->nm)
?? space(15 - LEN(trim(b->nm)))
?? str(b->onhand)+ ' + b->unit
?? str(qty) + str(b->unitcost)+str(b->unitcost*qty)
Store b->unitcost*qty to total
?
?? ' -----------------------------' 
?
?? ' 
?? ' str(total)
?
?
?' Material Management NCO :'
?? ' Date :____/____/____
?
?' 'Material Management Officer :
?? ' Date :____/____/____
Eject
Store qty to mqtys
Replace posted with .T.
Replace refilldate with Today
Select D
Seek mci
Replace expend with expend-total
Store expend to mexpend
Replace fundoh with fundoh-mexpend
Select B
Replace onhand with monhand -mqtys
Replace tvalue with (monhand-mqtys)*mcost
Select A
Append blank
Replace SN with STOCKN, CI with MCI, QTY with MQTYS, TYPE with "RD"
Replace REQNO with MREQNO, UNITCOST with MCOST, POSTED with .T.
Replace DATE with MDATE, MISC with .T., VN with VN1
Select F
Skip

109
Enddo
Set print off
Set console on
Stockn = space(16)
Select D
Enddo
Close databases
Clear

*------------------------* Eof TRSTOCK.prg *------------------------*

g. TRTURN-IN

**********************************************************************
** TRTURNIN.PRG **
**********************************************************************
* Program..: Turn-in.prg
* Author....: PARK, Taeyong
* Date.....: Sept 2, 1987
* Notes.....: Record Request for turn-in from Customer into Master and property book.

Select B
Use PROPERTY index SNPROPER
Select C
Use ASL index INASL
Select D
Use CUSTOMER index CICUST
Select E
Use BATCH index SNCIBAT,CIBAT
Store "The 150 Infantry Division" to Mcdesc2
Store "X" to checks,fcheck,Zcheck
If mcil = "
  Close databases
  Return
Xcheck = .T.
Do while Xcheck
  Store mcil to custm
  Do while .T.
  Clear
    @ 10,15 clear to 14,65
    @ 10,15 to 14,65
    @ 11,17 Say " Customer Code : " Get mcil pict "9999"
    @ 13,17 Say " Stock Number : " Get stockn ;
    Pict "9999-99-999-9999"
    @ 15,20 Say "Leave blank Stock Number to Exit"
    Read
    Select D
    Seek mcil
    If .NOT. found()
      @ 16,17 Say "Not found " +mcil+" Code customer...."
      @ 17,17 Say " Please check the code, or add customer first"
      Store "X" to ANS
      @ 16,15 clear to 18,70
      @ 17,17 Say "Do you want Add the customer Now? " ;
      get ans pict "/!"
      Read
      If ANS = "Y"
        Store mcil to mci
        Do ADDCUST
      Endif
    Else
      Store cdesc to mcdesc1
    Exit
  Endif
Enddo
If mcil <> custm

110
@10,15 clear to 14,65
@10,15 to 14,65
@12,20 Say "This data will not be added ......"
?? Chr(7)
N = 0
Do while N < 35
N = N + 1
Enddo
Exit
Endif
Select B
StockN = Upper(StockN)
Seek StockN
Do case
Case stockN = " "
  Select E
  Pack
Xcheck = .F.
Case .NOT. found()
  Clear
@13,10 say stockn + " is Not found in your property book"
@15,20 Say "Please check STOCK NUMBER and try again"
Wait
Case found()
  mcost = unitcost
  monhand = onhand
  Select E
  Append Blank
  Replace CI with MCI1, Type with Mtype, SN with StockN,;
  Unitcost with Mcost, Date with Today,Reqno with Mreqno
  Replace POSTED with .F.
  Do while .T.
    Set format to turnin
    Read
    zcheck = upper(zcheck)
    If zcheck = "Y"
      mqty = qty
      Store Reqno to mreqno
    Set format to
    Exit
  Endif
Enddo
Clear
Stockn = space(16)
Endcase
Enddo
Do TRTUNLST
Close Databases
Release all
Return
*-----------------------------------------------* Eof turnin.prg *-----------------------------------------------*

h. TRTUNLST

************************************************************************ TRTUNLST.PRG **************************************************************************
************************************************************************ TRTUNLST.PRG **************************************************************************
* Program.: TRTUNLST.prg
* Author.: Park, Taeyong
* Date.: Sept 2, 1987
* Notes.: Report generate program, called from TRTURNIN.prg
************************************************************************ TRTUNLST.PRG **************************************************************************
Clear
Select E

111
Set relation to sn into property
Store 'X' to mprint
Store 0 to total, mtotal, Z
Do while .T.
  Clear
  @ 10, 15 to 15.65
  @ 12, 25 Say 'Ready Printer? ' get mprint pict "!"
  Read
  If mprint = 'Y'
  Set print on
  @ 14, 25 Say "Printing...."
  Set console off
  Exit
  Else
  @ 15, 15 Say "Be ready printer ! 
  Wait
Endif
Enddo
Store Val(substr(Vnl, 8, 4)) to mvn
Store ?ivn = lQ0000+mvn+1
Vnl = "15Q0QM-"+substr(str(mvn), 7, 4)+"-"+Zulu
Store 1 to pagectr, linectr, N
Go top
Do while .NOT. eof()
  If type = mtype .AND. ci=Custm
    If .NOT. posted
      If linectr = 40*(N - 1) + 1
        Mvn = Val(substr(Vnl, 8, 4))
        Mvn = 100000+mvn+1
        Vnl = "15Q0QM-"+substr(str(Mvn), 7, 4)+"-"+Zulu
      
      ?? ' (Turn-In) 
      ?? From : The 150 Infantry Division 
      ?? 
      ?? Date : '+'DOC(date()) 
      ?? To : '+'mcdescl 
      ?? 
      ?? Voucher No : ' +Vnl 
      ?? 
      ?? No Stock number Description 
      ?? Quantity Unit price Total Reusables 
      Store pagectr + 1 to pagectr
    Endif
    Store linectr to Z
    If misc
      ? Str(Z,4)+ ' +SN+ ' +rtrim(b->nm)+space(15-LEN(rtrim(b->nm)))
      ?? ' + str(qty)+' + b->unit 
      ?? str(b->unitcost)+ str(b->unitcost*qty)+' 
      ?? Misc
      Store b->unitcost*qty to mtotal
      Store total+mtotal to total
      Replace posted with .T.
      Store B->onhand - qty to monhand
      Select B
      Replace onhand with monhand
      Replace tvalue with monhand*unitcost
      Select E
    Else
      ? Str(Z,4)+ ' +SN+ ' +rtrim(b->nm)+
      space(15-LEN(rtrim(b->nm)))+
      ?? '+str(qty)+' +b->unit 
      ?? misc
      Replace posted with .T.
      Store total to total
    Endif
Endif
Replace vn with vn1
If linectr=40*N
Endif
Endif
Enddo

Module name...... TRCancel.prg
* Author......... Park, Taeyong
* Date........... Sept 20, 1987
* Purpose......... Cancel the request for issue item
* Called by...... Transact.prg
* Modules called : None

Use Master index SCTMAST
Store "N" to Checks
Store Stockn to Msn
Store "1500" to Mcust
Store Msn+Mcust+"RL" to Msncu
Seek Msncu
Do case
  Case STOCKN = ""
    More = .F.
    ? Chr(7)
    @ 11,15 clear to 15,65
@ 11,15 to 15,65
@ 13,17 Say "Please Enter Stock number you want to cancel !"
N = 0
Do While N < 35
   N = N + 1
Enddo
Case found()
   Set format to CANCMAST
   Read
   If checks = "y"
      Replace Vn with "CANCEL-0000-0000", posted with .T.
      Store unitcost to mcost
      Store regno to mregno
      Store qty to mqty
      Append Blank
      Replace sn with stockn, ci with "1500", Regno with mregno
      Replace Qty with Mqty, Type with "CL", unitcost with mcost
      Replace Misc with .T., posted with .T., Date with today
      Replace Vn with "Cancel-0000-0000"
      More = .F.
   Endif
   Set format to
Case .NOT. found()
   @ 11,15 clear to 15,65
   @ 13,18 Say "You never request "+stockn+" code item !"
   ?? chr(7)
   N = 1
   Do While N < 35
      N = N + 1
   Enddo
Endcase
Close Databases
Return
*-----------------------------* Eof Cancel.prg *-----------------------------*

j. TRACKING

******************************************************************************* TRACKING.PRG *******************************************************************************
******************************************************************************* TRACKING.PRG *******************************************************************************
* Module name....: Trancking.prg  *
* Author..........: Park, Tae Yong  *
* Date............: Sept 20. 1987  *
* Purpose.........: Tracking the record of transaction.  *
* Called by.......: Transact.prg  *
* Modules called : None  *
******************************************************************************* TRACKING.PRG *******************************************************************************
Set talk off
Set echo off
Use master
Index on Ci+SN to CSMAST
Use master index scmast, CSMAST
title = "Sales Tracking"
Mok = "Y"
Do while .T.
Clear
   @ 4.25 say title
   @ 8.15 Say "Options"
   @ 10.17 Say "1. Detail Sales"
   @ 11.17 Say "2. Summary Sales"
   @ 12.17 Say "3. Detail Customer Sales"
   @ 13.17 Say "4. Summary Customer Sales"
   @ 15.17 Say "5. Return to main menu"
   @ 17.15 Say "Enter Option: "

@ 2,1 to 23,75
Store "" to sel
@ 17,29 Get sel
i=0
Do while i=0
i=inkey()
If chr(i)="12345"
Exit
Endif
i=0
Enddo
@ 17,29 Say Chr(i)
If chr(i)="5"
Exit
Endif
Store space(8) to Enddate
Store space(8) to begdate
@ 21,9 Say "Enter period for sales tracking ( )"
@ 21,42 Get begdate pict "99/99/99"
@ 21,53 Get Enddate pict "99/99/99"
Read
Store Ctod(begdate) to begdate
Store Ctod(Enddate) to enddate
Set filter to Date>begdate .AND. Date<Enddate
Do case
Case chr(i)="1"
Set Order to 1
Do setup
Report form tranp1
Case chr(i)="2"
Set Order to 1
Do setup
Report form tranp2
Case chr(i)="3"
Set Order to 2
Do setup
Report form tranp3
Case chr(i)="4"
Set Order to 2
Do setup
Report form tranp4
Otherwise
Loop
Endcase
If Upper(Mok)="Y"
Set console on
Set print off
Set Order to 1
Else
@ 24,17 Say "Press any key to continue..."
Wait"
Endif
Clear
Enddo
Close databases
Erase cimast.ndx
clear
Return
*---------------------------* Eof TRacking.prg *---------------------------*
3. REPORTS
   a. REPORTS MENU

******************************************************************************
******************************************************************************
****************************************************************************** REPORTS.PRG
******************************************************************************
******************************************************************************
* Module name: Reports.prg *
* Author: Park, Taeyong *
* Date: Sept 15, 1987 *
* Purpose: To generate report to higher command or to use in division *
* * *
* Called by: Pmain.prg *
* Modules called: RPTSR.prg RPSOR.prg RPOST.prg RPEIS.prg *
* Variables used: *
  * Public: Reportd(last report date) *
  * Local: Bigdate, Enddate, Msel *
* *
******************************************************************************

DO WHILE .T.
  Store space(8) to begdate, enddate
  Store "X" to Msel
  CLEAR
  @ 1.15 TO 3.55 double
  @ 4.1 TO 23.77
  @ 6.3 TO 21.38
  @ 5.4 TO 7.28
  @ 6.5 SAY SPACE(23)
  @ 6.41 TO 12.75
  @ 5.42 TO 7.59
  @ 6.43 SAY SPACE(16)
  @ 14,41 TO 21.75
  @ 13.42 TO 15.59
  @ 14,43 SAY SPACE(16)
  @ 2.25 SAY "R E P O R T S"
  @ 6.6 SAY "Type of Report"
  @ 6.44 SAY "Information"
  @ 14,44 SAY "Time Period"
  @ 9.6 SAY "1 : Transaction Status Report"
  @ 10,10 SAY "(DA form 3183)"
  @ 12.6 SAY "2 : Stockout Report"
  @ 12.10 SAY "(DA form 3184)"
  @ 15.6 SAY "3 : OST Report"
  @ 16.10 SAY "(DA form 3185)"
  @ 18.6 SAY "4 : Combat Essential Item"
  @ 19.10 SAY "Stockout Report"
  @ 20.10 SAY "(DA form 3186)"
  @ 8.46 SAY "Last TSR Report Date"
  @ 9.50 SAY TSDATE
  @ 11.46 SAY "Today.... +DOC(TODAY)+"("+zulu+")"
  @ 17.45 SAY "Beginning Date - Ending Date"
  @ 19.50 SAY ""
  SET COLOR TO N/W
  @ 19.51 SAY Begdate Pict "99/99/99"
  @ 19.62 SAY Enddate Pict "99/99/99"
  SET COLOR TO @ 22.13 SAY "Enter Selection from (1 - 4, or 0 to return) : :|
  STORE ":" TO SEL
  @ 22.60 SAY SEL
  *........The following lines are for select loop
  i=0
  DO WHILE i=0
    i=INKEY()
    @ 22.60 SAY ""
    IF UPPER(CHR(I))$"01234"
  @ 22.60 SAY CHR(I)
    EXIT
  ENDF
i=0
ENDDO
IF CHR(I) = '0'.OR. I=27
  ? Chr(7)
  Clear
  Exit
ENDIF
DO CASE
  Case CHR(I) = '2'
    @ 8, 46 SAY "Last SOR Report Date"
    @ 9, 50 SAY SORDATE
  Case CHR(I) = '3'
    @ 8, 46 SAY "Last OST Report Date"
    @ 9, 50 SAY OSTDATE
  Case CHR(I) = '4'
    @ 8, 46 SAY "Last EIS Report Date"
    @ 9, 50 SAY EISDATE
ENDCASE
@ 19,51 GET Begdate Pict "99/99/99"
@ 19,62 GET Enddate Pict "99/99/99"
READ
IF BEGDATE = " " .OR. ENDDATE=" "
  ? CHR(7)
  @ 24,15 SAY "YOU HAVE TO ENTER THE PERIOD"
K=0
DO WHILE K<40
  K=K+1
ENDDO
LOOP
Store Ctod(begdate) to begdate
Store Ctod(endate) to endate
DO CASE
  Case CHR(I) = '1'
    DO RPTSR
      STORE TODAY TO TSRDATE
    Case CHR(I) = '2'
      DO RPSOR
        STORE TODAY TO SORDATE
      Case CHR(I) = '3'
        DO RPOST
          STORE TODAY TO OSTDATE
        Case CHR(I) = '4'
          DO RPEIS
            STORE TODAY to EISDATE
          OTHERWISE
            ? CHR(7)
    ENDCASE
ENDDO
RETURN

b. RPTSR

*******************************************************************************
*******************************************************************************
*******************************************************************************
*******************************************************************************
*******************************************************************************
*******************************************************************************
*******************************************************************************
*******************************************************************************
*******************************************************************************
*******************************************************************************
*******************************************************************************
*******************************************************************************
*******************************************************************************
*******************************************************************************
*******************************************************************************
*******************************************************************************
*******************************************************************************
*******************************************************************************
*******************************************************************************
*******************************************************************************
*******************************************************************************
Clear
Select B
Use PROPERTY index snproper
Do while .T.
  Store "" to Classn. Mprint
  Store "Transaction Status Reports" to Mtitle
  Do Select
    Store "(Date>=Begdate .AND. Date=Enddate)" to period
    Store "Type='RL'.OR.Type='IL'.OR.Type='CL'.OR.Type='TL'" to tcond
    Store &PERIOD" TO CN1
    Select A
    Use MASTER index sctmast
    Copy to M1mast for &tcond
    Do case
      Case upper(Chr(i))="1"
      Store "1" to Sel
      Select B
      Index on Class+Sn to CSPROPER
      Select A
      Use M1mast
      Copy to STMAST for &cn1
      Exit
      Case upper(Chr(i))="2"
      Store "2" to Sel
      If classn = "0"
        Store "10" to Classn
      Endif
      Select A
      Use M1mast
      Index on sn to MiMAST
      Use MiMAST INDEX M114Ast
      Set relation to sn into B
      Copy to STMAST for b->class='&classn'
      Use STMAST
      Erase xtmast.dbf
      Erase xtmast.ndx
      Exit
      Case upper(Chr(i))="3"
      Store "3" to Sel
      @ 16,30 Say "Enter stock number" get stockn;
      Read
      Select A
      Use M1mast
      Index on sn to M1mast
      Use M1MAST INDEX M1M1Ast
      Store "&period .AND. sn='&stockn'" to conditi
      Copy to STMAST for &conditi
      Exit
      Case upper(Chr(i))="0"
      Close databases
      Return
      Otherwise
      ? chr(7)
    Endcase
  Enddo
Enddo
Select A
Use STMAST
Index on SN+TYPE+CI to STMAST
Use STMAST index stmast
Erase M1MAST.dbf
Erase M1MAST.ndx
Do printer
Store 1 to linectr, pagectr, N, liner
Store 0 to mdi, reqqty, canqty, rec qty, turn qty, DINow, mqty
Store 0 to sreqqty, smdi, scanqty, srecqty, sturnqty, sdn now

If Eof() @ 10.18 Clear to 14.58
@ 10.18 to 14.58
@ 11.20 SAY "You Never Requested"
@ 13.20 Say "Press any key to return...."
Wait"

Close databases
Erase SIMAST.DBF
Erase SIMAST.NDX
Return

Endif
select A
Clear
Go top
*..........................Holds one of item from property file

Do while .NOT. Eof() .AND. sn=stockn
Store SN to stockn
select B
store class to classn
Store nn to mnm
Select A
Seek stockn
If found()
Sum qty*Unitcost for type="RL" .AND. date < begdate
Store qty*unitcost to RLqty
Sum qty*unitcost for type="IL" .AND. date < begdate
Store qty to ILqty
MDI=RLqty - ILqty
Seek stockn
Do while .NOT. Eof() .AND. sn=stockn
Store qty*unitcost to mqty
Do case
  Case type = "RL"
    Req qty = reqqty+mqty
  Case Type = "CL"
    Can qty = Can qty+mqty
  Case Type = "IL"
    Rec qty = Rec qty+mqty
  Case Type = "TL"
    T urn qty = Turn qty+mqty
Endcase
Skip
Enddo
Store MDI+req qty-Can qty-Rec qty to DINow
If liner=1
  ?
  ?
  ?
  ?
  ??
  ??
  ?
  ?
  ??
  ?
  ?
  ?
  ?
  ?
  ?
  ?
  ?
  ?
  ?
  ?
  ?? From : The 150 Infantry Division
  ?
  ?
  ?
  From To : The 3333 Logistics Support Command
  ?
  ?
  ?
  ?
  ?
  ?
  ?
  ?
  ? Dtoc(begdate)+'- '+Dtoc(enddate)
  ?
  ?'No Stock number Description '
  ?
  ?'D/Ibeg Reg st Cancel Rec ve Turnin D/Inow'
  Store pagectr + 1 to pagectr
Endif

Store linectr to Z
? Str(Z,4)4' '+stockn+' '+rtrim(mnm)+space(15 - LEN(rtrim(mnm)))
119
? ' +substr(str(MDI),5,6)+ ' + substr(str(reqqty),4,7)
? ' +substr(str(Canqty),4,7)+substr(str(Recqty),4,7)
? ' +substr(str(Turnqty),4,7)+substr(str(Dlnow),4,7)
Store Smdi+mdi to Smdi
Store Sreqqty+reqqty to Sreqqty
Store Scanqty+canqty to Scanqty
Store Srecqty+recqty to Srecqty
Store Sturnqty+turnqty to Sturnqty
Store Sdinow+dinow to Sdinow
Store 0 to mdi,reqqty,canqty,recqty,turnqty,Dlnow
Linectr=linectr+1
Liner=liner+1
If liner=40*N

? ' +substr(str(MDI),5,6)+ ' + substr(str(reqqty),4,7)
? ' +substr(str(Canqty),4,7)+substr(str(Recqty),4,7)
? ' +substr(str(Turnqty),4,7)+substr(str(Dlnow),4,7)
Endif
Enddo
If liner>1

? ' +substr(str(MDI),5,6)+ ' + substr(str(reqqty),4,7)
? ' +substr(str(Canqty),4,7)+substr(str(Recqty),4,7)
? ' +substr(str(Turnqty),4,7)+substr(str(Dlnow),4,7)
Endif
Enddo
If mprint="Y"
Eject
Set console on
Set print off
Endif
Set color to N/W
Wait
c. RPOST

******************************************************************************
** RPOST.PRG ******************************************************************************
******************************************************************************
* Module name:.... RPOST.PRG  *
* Author...........: Park, Tae-yong  *
* Date..........: Sept 25. 1987  *
* Purpose.........: Generate Order shipping time report  *
* Called by.....: Reports.prg  *
* Modules Called : None  *
******************************************************************************

Clear
Select B
Use PROPERTY index snproper
Do while .T.
    Store " " to Classn, Mprint
    Store "ORDER SHIPPING TIME REPORT" to mtitle
    Do select
        Store "(Date>=Begdate .AND. Date<=Enddate)" to period
        Do case
            Case upper(chr(i))="1"
                @ 9,15 to 15,64 double
                @ 11,25 Say "Finished report for TSR "
                @ 13,25 Say "Period "+Dtoc(begdate)+" - "+Dtoc(enddate)
            Endif
            Do while n<35
                N=N+1
            Enddo
            Clear
            Store Space(16) to stockn
            Release all
            Return
        Case upper(chr(i))="2"
            @ 14,35 Say "Enter Class number(1-10) " get Classn
            @ 15,35 Say " (0 for 10)"
            Read
            If classn = "0"
                Store "10" to Classn
            Endif
            Select A
            Use master INDEX stcmast
            Set relation to sn into B
            Store "type='RL' .AND. &period .AND. b->class='&classn'';
to condit1
Copy to RLMAST for &condit1
Store "type='IL' .AND. &period .AND. b->class='&classn'"
  to condit2
  Copy to ILMAST for &condit2
Exit

Case upper(chr(i))="3"
  Select A
  Use master INDEX sctmast
  Set relation to sn into B
  Store "type='RL' .AND. &period .AND. sn='&stockn'"
  to condit1
  Copy to RLMAST for &condit1
  Store "type='IL' .AND. &period .AND. sn='&stockn'"
  20 to condit2
  Copy to ILMAST for &condit2
Exit

Case upper(chr(i))="0"
Close databases
Return
Otherwise
Endcase
Enddo
Select A
Use RLMAST
Index on SN+VN to SRRL
Use RLMAST index SRRL
If Eof()
@ 10,18 Clear to 14,58
@ 10,18 to 14,58
@ 11,20 SAY "You Never Requested"
@ 13,20 Say "Press any key to return...."
Wait ""
Close databases
Erase RLMAST.DBF
Erase SRRL.NDX
Return
Endif
Select C
Use ILMAST
Index on SN+VN to SRIL
Use ILMAST index SRIL
If Eof()
@ 9,18 clear to 13,60
@ 9,18 to 13,60
@ 10,20 SAY "You Never Received the requested item"
@ 12,20 Say "Press any key to return...."
Wait ""
Close databases
Erase RLMAST.DBF
Erase SRRL.NDX
Erase ILMAST.DBF
Erase SRIL.NDX
Return
Endif
Store 0 to SN1,SN2,SN3,SN4,SN5,SM1QTY,SM2QTY,SM3QTY,SM4QTY,SM5QTY
Store 0 to N1,N2,N3,N4,N5,M1QTY,M2QTY,M3QTY,M4QTY,M5QTY
Store 0 to STOTN,TOTQTY,TOTN,STOTQTY,mdate
Store 1 to linectr,pagectr,N,liner
Go top
Do printer
Clear
Select A
Go top
*.................................Holds one of item from property file

Do while .NOT. Eof()
    Store SN to stockn
    select B
    store class to classn
    Select C

*.................................Find the stock number which is the same as
*.................................at the file which is sorted type "IL"

Seek stockn
If found()
    Do while .NOT. Eof() .AND. sn=stockn
        Store date to mdate
        Store VN to M1VN
        Store Stockn+M1VN to mstock
        *............................Find the stocknumber which is the same voucher
        *............................at the file which is sorted by type "RL"
        Select A
        Seek mstock
        If found()
            Store date to m2date
            Store qty to mqty
            Store mdate=m2date to mdate
            Do case
                Case mdate<7
                    N1=N1+1
                    M1qty=M1qty+mqty
                Case mdate=7<15
                    N2=N2+1
                    M2qty=M2qty+mqty
                Case (Mdate>=15 .and. Mdate<30)
                    N3=N3+1
                    M3qty=M3qty+mqty
                Case (Mdate>=30 .and. Mdate<60)
                    N4=N4+1
                    M4qty=M4qty+mqty
                Case (Mdate>=60 .and. Mdate<90)
                    N5=N5+1
                    M5qty=M5qty+mqty
            Endcase
        Endif
    Store N1+N2+N3+N4+N5 to TOTN
    Store M1QTY+M2QTY+M3QTY+M4QTY+M5QTY to TOTQTY
    Select C
    Skip
Enddo

If line1=1

?? ' Report  

?? ' Page ' + str(pagectr,2)
?? ' Date : ' + dtoc(date())
?? ' To : The 3333 Logistics Support Command'
?? ' From : The 150 Infantry Division'
?? ' From
?? ' To  

?? ' (' + dtoc(begdate) + ') - (' + dtoc(enddate) + ')
?? ' No Stock number

?? ' Total Qty
?? ' Total 7days 14days 30days 60days
?? ' Qty Freq Qty Freq Qty Freq Qty

store pagectr + 1 to pagectr
Endif
Store linectr to Z
? Str(Z.3)+"+stockn+substr(str(TOTN).5.6)
?? substr(str(TOTQTY).5.6)
?? substr(str(N1).5.6)+substr(str(M1QTY).5.6)
?? substr(str(N2).5.6)+substr(str(M2QTY).5.6)
?? substr(str(N3).5.6)+substr(str(M3QTY).5.6)
?? substr(str(N4).5.6)+substr(str(M4QTY).5.6)
Store STOTN+TOTN to STOTN
Store STOTQTY+TOTQTY to STOTQTY
Store SN1+NI to SN1
Store SH1QTY+H1QTY to SM1QTY
Store SN2+N2 to SN2
Store SH2QTY+H2QTY to SM2QTY
Store SN3+N3 to SN3
Store SM2QTY+M2QTY to SM2QTY
Store SN4+N4 to SN4
Store SM4QTY+M4QTY to SM4QTY
Linectr=linectr+1
liner=liner+1
If liner=40*N

? 'Material Management NCO : ________________________'
?? 'Date : __________'
?? 'Material Management Officer : ____________________'
?? 'Date : __________'
Eject
N = N + 1
Endif
Endif
If sel="1"
If linectr=1
Select B
Skip
Store class to classn
Loop
Select B
Skip
Endif
If (b->class<>classn) .OR. EOF()

? 'Class '+classn'
?? '-----------------

? 'Subtotal '
?? substr(str(STOTN).5.6)+substr(str(STOTQTY).5.6)
?? substr(str(SN1).5.6)+substr(str(SM1QTY).5.6)
?? substr(str(SN2).5.6)+substr(str(SM2QTY).5.6)
?? substr(str(SN3).5.6)+substr(str(SM3QTY).5.6)
?? substr(str(SN4).5.6)+substr(str(SM4QTY).5.6)
??
Store 0 to SN1,SN2,SN3,SN4,SN5,SH1QTY,SH2QTY,SM2QTY,SM3QTY,SM4QTY,SM5QTY
Store 0 to N1,N2,N3,N4,N5,NIQTY,M2QTY,M3QTY,M4QTY,M5QTY
Store 0 to STOTN,TOTQTY,TOTN,TOTQTY,mdate
Store 1 to linectr
Liner=liner+3
Endif
Enddo
If liner > 1

? 'Material Management NCO : ________________________'
?? 'Date : __________'
?? 'Material Management Officer : ____________________'
?? 'Date : __________'
Endif

124
If mprint="Y"
   Eject
   Set console on
   Set print off
Endif
Close databases
Erase Csproper.ndx
Erase RLMAST.DBF
Erase SRIL.NDX
Erase ILMAST.DBF
Erase SRIL.NDX
Set color to N/W
Wait
Set color to
@ 9.15 clear to 15,64
@ 9.15 to 15.64 double
@ 13.25 Say "Finished report for OST"
N=0
Do while n<35
   N=N+1
   Store Space(16) to stockn
   Release all
   Clear
Return
*--------------------------------* Eof OST.prg *--------------------------------*

### d. RPEIS

********************************************************************************************
* Module name.....: RPEIS.prg
* Author..........: Park, Taeyong
* Date............: Sept 24, 1987
* Purpose.........: Generate Combat essential item Stockout history report to higher command.
* Called by.......: Reports.prg
* Modules called :
* Variables used:
** Public: S, Sts, Mprint, Mqty, Mcost, **
********************************************************************************************

Select B
Use PROPERTY
Copy to TEMPPROP for ESSENCE
Use TEMPPROP
Index on Class+SN to CSproper
Use TEMPPROP index CSPROPER
Select A
Use STOCKOUT
Index on SN to SNSTOK
Use STOCKOUT index SNSTOK
Select B
Store 1 to linectr, pagectr, N, liner
Store 0 to sts30, sts60, stsh, stl20, stco30, stco60, stco90, stco120
Store 0 to S30, S60, s90, s120, sco30, sco60, sco90, sco120, mqty, mscost
Go to top
Store CLASS to CLASSN
Store " " to Mprint
Do printer
Clear
Do while .NOT. Eof()
   Store SN to STOCKN
   Store NM to MNM

125
Select A
Seek STKCN
If found()
    Do while .NOT. Eof().AND. sn=stockn
        Store QTY to MQTY
        Store QTY*UNIT COST to MCOST
        If Dtoc(REFILLDATE)=""
            Skip
        Endif
        Store REFILLDATE to FILLDATE
        Store OUTDATE to SOUTDATE
        Bal = FILLDATE-SOUTDATE
        Do case
            Case bal <= 30
                S30 = S30+mqty
                Sco30 = Sco30+mcost
            Case Bal <= 60 .AND. Bal>30
                S60 = S60+mqty
                Sco60 = Sco60+mcost
            Case Bal <= 90 .AND. Bal>60
                S90 = S90+mqty
                Sco90 = Sco90+mcost
            Case Bal > 90
                S120 = S120+mqty
                Sco120 = Sco120+mcost
        Endcase
        Skip
    Enddo
    Store S30+S60+S90+S120 to SQTY
    Store SCO30+SCO60+SCO90+SCO120 TO SCOST
If liner=1
    ?? 'Combat Rssential Item '
    ?? 'Stockout Report'
    ?? 'Page ' + Str(pagectr,2)
    ?? 'Date :'+Dtoc(dateo)
    ?? 'To : The 3333 Logistics Supprot Command'
    ?? 'From : The 150 Infantry Division'
    ?? From To'
    ?? ('+Dtoc(begdate)+') - ('+Dtoc(enddate)+')
    ?? 'No Stock number ''
    ?? 'Total 30days 60days 90days 120days'
    ?? 'Total Cost Item Cost Item Cost Item Cost Item Cost'
    ?? 'Item Cost'
    store pagectr + 1 to pagectr
Endif
Store linectr to Z
? Str(2,3)+ ' '+stockn+substr(str(sqty),5,6)
? substr(str(scost),5,6)
? substr(str(S30),5,6)+substr(str(sco30),5,6)
? substr(str(S60),5,6)+substr(str(sco60),5,6)
? substr(str(S90),5,6)+substr(str(sco90),5,6)
? substr(str(S120),5,6)+substr(str(sco120),5,6)
Store msgty+sqty to msgty
Store mscost+scost to mscost
Store Sts30+s30 to Sts30
Store Stco30+sc30 to Stco30
126
Store Sts60+s60 to Sts60
Store Stco60+sc60 to Stco60
Store Sts90+s90 to Sts90
Store Stco90+sc90 to Stco90
Store Sts120+s120 to Sts120
Store Stco120+sc120 to Stco120
Linectr=linectr+1
liner=liner+1
If liner=40*N
    Linectr=linectr+1
    liner=liner+1
Endif
Endif
If linectr=1
    Select B
    Skip
    Store Class to classn
    Loop
Endif
Select B
Skip
If class<>classn .OR. Eof()
    'Class '+classn+'
Enddo
Subtotal
    'Subtotal '+substr(str(msqty),5,6)+substr(str(mscost),5,6)
    'Subtotal '+substr(str(sts30),5,6)+substr(str(stco30),5,6)
    'Subtotal '+substr(str(sts60),5,6)+substr(str(stco60),5,6)
    'Subtotal '+substr(str(sts90),5,6)+substr(str(stco90),5,6)
    'Subtotal '+substr(str(sts120),5,6)+substr(str(stco120),5,6)
    Store 0 to sts30,sts60,sts90,sts120,stco30,stco60,stco90,stco120
    Store 0 to s30,s60,s90,s120,sc30,sc60,sc90,sc120,mscost,msqty
    Store 1 to linectr
    Liner=liner+4
Endif
Store class to classn
Enddo
If liner > 1
    'Material Management NCO : 
    'Material Management Officer : 
Endif
If mprint ="y"
    'Set print off
    'Set console on
Endif
Close all
Erase SStok.ndx
Erase tempprop.dbf
Erase cspnprop.ndx
Erase cspnproper.ndx
Clear to 15,64
Clear to 15,64
@ 9,15 Say "Finished report for Stockout"
@ 13,25 Say "Period "+Dtoc(begdate)+" - "+Dtoc(enddate)
N=0
Do while n<35
    N=N+1
Enddo
Release all
Clear
Return

*-----------------------------* Eof Eis.prg *-----------------------------*

e. RPSOR

********************************************************************************
*** RPSOR.PRG ********************************************************************************
Module name..... RPSOR.prg
Author........... Park, Taeyoung
Date............. Sept 24. 1987
Purpose........ Generate Stockout history report
* Called by....... Reports.prg
********************************************************************************

Select B
Use PROPERTY index snproper
Clear
Do while .T.
    Store "" to Classn, Mprint
    Store "STOCKOUT REPORT" to mtitle
    Do select
        Store "(OutDate>=Begdate .AND. outDate<=Enddate)" to period
        Do case
            Case upper(chr(i))="1"
                Sel=1''
                Select B
                Index on Class+Sn to CSPROPER
                Select A
                Use stockout
                Copy to TEMPOUT for &period
                Exit
            CASE upper(chr(i))="2"
                Sel=2''
                @ 14.30 Say "Enter Class number(1-10) " get Classn
                @ 15.30 Say "(0 for 10)"
                Read
                If classn = "0"
                Store "10" to Classn
                Endif
                Select A
                Use stockout INDEX stockout
                Set relation to sn into B
                Store "&period .AND. b->class='&classn'" to condit1
                Copy to TEMPOUT for &condit1
                Exit
            Case upper(chr(i))="3"
                Sel=3''
                @ 16.30 Say "Enter stock number" get stockn;
                pict '9999-99-999-9999'
                Read
                Select A
                Use stockout INDEX stockout
                Set relation to sn into B
                Store "&period .AND. sn='&stockn'" to condit1
                Copy to TEMPOUT for &condit1
                Exit
            Case upper(chr(i))="0"
                Close databases
                Return
            Otherwise
                ? chr(7)
        Endcase
    Endselect
Enddo

Endcase
Enddo
Select A
Use TEMPOUT
Index on SN to SNTEMP
Use TEMPOUT Index SNTEMP
If Eof()
  @ 10,18 Clear to 15,58
  @ 10,18 to 15,58
  @ 13,20 SAY " Never Stockout"
  @ 13,20 Say " Press any key to return...."
  Wait ""
  Close databases
  Erase TEMPOUT.DBF
  Erase SNTEMP.NDX
  Return
Endif
Do printer
Clear
Store 1 to linectr,pagectr,1,liner
Store 0 to s30,s60,s90,s120,sto30,sto60,sto90,sto120
Store 0 to S3O,S60,S90,s120,sto3O,sto60,sto9O,sto12O,msqty,mscost
Go top
Do while .NOT. Eof()
  Store sn to stockn
  Select B
  Store nm to mnm
  Store class to classn
  Select A
  Seek stockn
  If found()
    Do while .NOT. Eof().AND. SN=STOCKN
      Store qty to mqty
      Store qty*unitcost to mcost
      If Dtoc(refilldate)=" "
        Skip
      Loop
    Store refilldate to filldate
    Store outdate to soutdate
    Bal = (filldate)-(soutdate)
    Do case
      Case bal <= 30
        S3O = S3O+mqty
        Sco3O = Sco3O+mcost
      Case Bal <= 60 .AND. Bal>30
        S6O = S60+mqty
        Sco6O = Sco6O+mcost
      Case Bal <= 90 .AND. Bal>60
        S90 = S90+mqty
        Sco90 = Sco90+mcost
      Case Bal > 90
        S12O = S12O+mqty
        Sco12O = Sco12O+mcost
    Endcase
    Skip
  Enddo
Store S30+S60+S9O+S12O to SQTY
Store SC03O+SC06O+SC09O+SC012O TO SCOST
If liner=1
  ?? ' Stockout Report'
  ?? ' Page ' + str(pagectr,2)
  ?? ' Date : ' + Dtoc(date())
129
To: The 3333 Logistics Support Command

From: The 150 Infantry Division

No Stock number

Total 30days 60days 90days 120days

Total Cost Item Cost Item Cost Item Cost

store pagectr + 1 to pagectr

Endif

Store linectr to Z

Str(Z,3)+' '++stockn+substr(str(sqty),5,6)

substr(str(scost),5,6)

substr(str(s30),5,6)+substr(str(sco30),5,6)

substr(str(s60),5,6)+substr(str(sco60),5,6)

substr(str(s90),5,6)+substr(str(sco90),5,6)

substr(str(s120),5,6)+substr(str(sco120),5,6)

Store msqty+sqty to msqty

Store mcost+scost to mcost

Store Sts30+s30 to Sts30

Store Stco30+sco30 to Stco30

Store Sts60+s60 to Sts60

Store Stco60+sco60 to Stco60

Store Sts90+s90 to Sts90

Store Stco90+sco90 to Stco90

Store Sts120+s120 to Sts120

Store Stco120+sco120 to Stco120

Linectr=linectr+1

liner=liner+1

If liner=40*N

'Material Management NCO: ________________________

'Material Management Officer: ________________________

Eject

N=N + 1

Endif

Endif

If sel="1"

If linectr=1

Select B

Skip

Store class to classn

Loop

Endif

Select B

skip

Endif

If b->class<>classn .OR. Eof()

'Class '++classn' ---------'

'---------------------------------'

'Subtotal '

substr(str(msqty),5,6)+substr(str(mcost),5,6)

substr(str(sts30),5,6)+substr(str(stco30),5,6)

substr(str(sts60),5,6)+substr(str(stco60),5,6)

substr(str(sts90),5,6)+substr(str(stco90),5,6)

substr(str(sts120),5,6)+substr(str(stco120),5,6)

Store 0 to sts30,sts60,sts90,sts120,stco30,stco60,stco90,stco120

Store 0 to s30,s60,s90,s120,sco30,sco60,sco90,sco120,mscost,msqty

Store 1 to linectr

Liner=liner+3

Endif

130
Enddo
If liner > 1
? 
? 'Material Management NCO : ___________________________
? ? ' Date : ______/____/____ 
? 
? 'Material Management Officer : _________________________
? ? ' Date : ______/____/____ 
Endif
Close all
If mprint = "y"
Eject
Set console on
Set print off
Endif
If sel="1"
Erase csproper.ndx
Endif
Erase tempout.dbf
Erase SNtemp.ndx
Set color to N/W
Wait
Set color to
@ 9,15 clear to 15,64
@ 9,15 to 15,64 double
@ 11,25 Say "Finished report for Stockout ":
@ 13,25 Say "Period "+Dtoc(begdate)+" - "+Dtoc(enddate)
N=0
Do while n<35
N=N+1
Enddo
Clear
Store space(16) to stockn
Return
*------------------------------------------------- Eof SOR.prg *--------------------------*
4. ANALYSIS

4.1 ANALYSIS STARTUP

****************************************************************************** ANALYSIS.PRG ******************************************************************************
******************************************************************************
* Module name ....: ANALYSIS.prg
* Author..........: Park, Taeyong
* Date ...........: Sept 28, 1987
* Purpose ........: Analyze the reorder point, safety level
* EOQ by using probabilistic model
* (Fixed Order Size System)
* Called by .......: PMAIN.prg
* Modules called : ANEOQ.prg, ANLEADT.prg
* Variables used : Public.: TODAY, STD, STOCKN
* Local.: BEGDATE, ENDDATE, CTITLE, MLEADT, MQTY, ORDERC,
* SERVL, FRATE, RSTD, LSTD, N, M, CSC, CSL.
******************************************************************************

DO WHILE .T.
  Store space(8) to BEGDATE, ENDDATE
  Store space(50) to CTITLE
  Store "N" to CSL, CSC
  Store space(16) to STOCKN, MREQNO
  Store 0 to MLEADT, MQTY, RDEM, ORDERC, SCOST, SERVL, FRATE, RSTD, LSTD
  Store 0 to DATEBAL, LEADDEM, TOTDATE, MLEADT
  Store 1 to N, M
  CLEAR

Text

THIS SYSTEM USES THE FIXED ORDER SIZE SYSTEM WITH PROBABILISTIC MODEL. THE FIXED ORDER SIZE SYSTEM IS COMPLETELY DEFINED BY THE ORDER QUANTITY Q, AND REORDER POINT B. THE RISK OF STOCKOUT OCCURS AFTER REORDER POINT. TO GET Q, B, YOU HAVE TO DECIDE SERVICE LEVEL, OR STOCKOUT COST PER UNIT. THIS SYSTEM ASSUMED BACKORDER CASE WITH SERVICE LEVEL INSTEAD OF LOSTSALES CASE.

Endtext

@19.17 SAY "Enter stock number: "
@21.12 SAY "Enter time period you want to;"
  test("
  SET COLOR TO N/W
  @21.48 SAY "01/01/87"
  @21.59 SAY "12/31/87"
  SET COLOR TO
  @22.50 SAY "Begin End date"
  @24.20 SAY "Leave BLANK any space to Exit"
  @1.10 TO 3.60 DOUBLE
  @4.1 TO 23.77 DOUBLE
  @2.15 SAY "A N A L Y S I S of T R A N S A C T I O N"
  @3.62 SAY "Today: " +DTOC(DATE())
  @19.38 GET STOCKN PICT "9999-99-9999-9999"
  READ
  IF STOCKN= 'EXIT
    ENDIF
  @21.48 GET BEGDATE PICT "99/99/99"
  @21.59 GET ENDDATE PICT "99/99/99"
  Read
  If Begdate= "" .OR. Enddate= "" .OR. Stockn= "" Exit
Endif
Store Ctod(Begdate) to Begdate
Store Ctod(Enddate) to Enddate

If BEGDATE>=ENDDATE
  Clear
  @ 15, 25 SAY "The End date MUST be later than the begin date"
  @ 17, 30 SAY "Press ENTER to redo....."
  Wait " "
  Loop
Endif
Do ANLEADT
  If Rdemc=0
    Clear
    Loop
  Endif
  Do ANEOQ
Enddo
Close DATABASES
RELEASE ALL
STOCKN=SPACE(16)
RETURN
*--------* Eof: Analysis.prg *--------*

b. ANLEADT

******************************************************************************
******************************************************************************
******************************************************************************
** Module name.......: Leadt.Prg
** Author...........: Park, Tae Yong
** Date............: Sept 30, 1987
** Purpose.........: Calculate Lead time to Supply for given Item
** Called by.......: Analysis.Prg
** Modules called.: Proc.prg
** Variables used.: Public.: Mleadt,Mqty,Mzulu
** Local.:          
******************************************************************************

Set procedure to ANPROC
SET CONSOLE OFF
Clear
  @ 10,15 to 14,65
  @ 11,25 Say " Please do NOT touch"
  @ 13,25 Say " Working....... "
Select B
Use PROPERTY index snproper
Select A
Use master index sctmast
Copy to RLMAST for type="RL" .AND. (Date>=Begdate .AND. Date<=Enddate);
  .AND. sn=stockn
Copy to ILMAST for type="IL" .AND. (Date>=Begdate .AND. Date<=Enddate);
  .AND. sn=stockn
Copy structure to DEM fields qty,type,date
Select E
Use DEM
Select C
Use ILMAST
Index on REQNO to SRIL
Use ILMAST index SRIL
Select D
Use RLMAST
Index on REQNO to SRRL
Use RLMAST index SRRL
Select C
If Eof() ? chr(7)
@ 15, 25 SAY "You receive nothing from LSC between"
@ 16, 30 SAY Dtoc(Begdate) + " - " + Dtoc(Enddate)
@ 18, 25 SAY "Press ENTER to return...."

Wait "11

Close databases
Erase RLMAST.DBF
Erase SRRL.NDX
Erase ILMAST.DBF
Erase SRIL.NDX
Erase DEM.DBF
Release all
Set console on
Return

Endif

*.............To calculate the lead time of an item

Go top

Do while .NOT. Eof()
Store reqno to MREQNO
Store date to ILDATE
Select D
Seek Mreqno
If Eof() OR. Bof()
Select C
Skip
Loop
Else
Store Date to RLDATE
Do while reqno=mreqno .AND. .NOT. Eof()
N=N+1
Datebal=(c->Date) - (Date)
Totdate=Todate+Datebal
If rldate>date
Store Date to Rldate
Endif
Skip
Enddo

Endif

Select A
Sum qty for sn=stockn .AND. type="RD" .AND. ;
(date)=RLDATE .AND. date<=ILDATE) to leaddem
Select E
Append blank
Replace qty with leaddem
Replace type with "LD"
Replace date with Rldate
Store 0 to leaddem
Select C
Skip
Enddo

Store Todate/N to Hleadtl
store 1Mleadtl/30 to mleadt

*......................

To get Annual average demand (1 year moving average)

Select A
Store "date=begdate .AND. date<=enddate" to Condx
Sum qty for sn=stockn .AND. type="RD" .AND. &condx to Rdem
Store enddate-begdate to totd
Store Rdem*365/totd to Rdem
Store Round(totd/30,0) to Nmax

*......................

To get standard deviation of annual demand from monthly demand

Go top

Store begdate to mdate
Store "(sn=stockn .AND. type='RD')" to cond1
Store "(date>=mdate+30*N .AND. date<=mdate+30*(N+1))" to cond2
Do while N < Nmax
Sum qty for &cond1 .AND. &cond2 to mqty
Select E
Append blank
Replace qty with mqty
Replace date with (mdate+30*n)
Replace type with "DM"
N= N+1
Select A
Store 0 to mqty
Enddo
Select E
Do Std with "qty","Type='DM'"
Close databases
Erase RLNAST.DBF
Erase SRRL.NDX
Erase ILNAST.DBF
Erase SRIL.NDX
Erase DEM.DBF
SET CONSOLE ON
SET PROCEDURE TO RETURN
------------------* Eof ANLEADT.prg *-------------------

c. ANEOQ

***************************************************************************************
***************************************************************************************
***************************************************************************************
***************************************************************************************
***************************************************************************************
***************************************************************************************
***************************************************************************************
***************************************************************************************
***************************************************************************************
***************************************************************************************
***************************************************************************************
***************************************************************************************
***************************************************************************************
***************************************************************************************
***************************************************************************************

Set decimals to 5
Select C
Use ASL index INASL
Select B
Use NORMAL
Select A
Use PROPERTY index snproper
Set relation to SN into C
Do while .T.
    Seek stockn
    Store "PRESENT" to title
    Clear
    Store "Is this record right?(Y/N):" to ctitle
    Store Ltrim(str(ASL->ROP)) to Mrop
    Store Ltrim(str(ASL->SL)) to Msl
    Store Ltrim(str(ASL->RO)) to Mro
    Store Ltrim(str(ASL->OST)) to Most
    Do while .T.
        Set format to EOQ
        Read
        If check = "Y"
            Set format to
            Exit
        Endif
    Enddo

    ............Holding cost
    If Frate=0
        Store 15.0 to Frate
    Endif

135
Store \((\text{Frate}/100)\ast \text{Property->unitcost}\) to \(\text{Holdc}\)

*......ordering cost

If \(\text{Orderc} = 0\)

Store 20.0 to \(\text{Orderc}\)

Endif

Do case

Case (\(\text{CSL} = "Y" \).AND. \(\text{CSC} = "Y"\)).OR.(\(\text{CSL} = "N" \).AND. \(\text{CSC} = "N"\))

\n?

@ 10. 15 clear to 16.65
@ 10.15 to 14.65
@ 12.20 SAY "You have to select ONE of Choices"
@ 15.20 Say "Press ENTER to redo......"
Store 0 to Frate,Orderc,Scost,Serv1
Store "N" to CSC,CSN
Store "X" to check
Wait " "
Loop

*........................Holding cost rate

Case CSL="N".AND. CSC="Y"

\(Q_0 = \sqrt{2\ast \text{Rdem}\ast \text{Orderc}/\text{Holdc}}\)
\(\text{Sigfofb} = \text{Holdc}\ast Q_0/(\text{Scost}\ast \text{Rdem})\)
Select B
Locate for sigfofb <=msigfofb
Store \(T_z\) to Mtz
Store Eofz to meofz

Case CSL="Y".AND. CSC="N"

If \(\text{Servl} = 0\)

Store 85.000 to serv1

Endif

Store serv1/100 to mServ1
\(Q_0 = \sqrt{2\ast \text{Rdem}\ast \text{Orderc}/\text{Holdc}}\)
\(\text{Scost}=\text{Holdc}\ast Q_0/(mServ1\ast \text{Rdem})\)
Store l-Servl to mpofs
Select B
Locate For pofs<=mpofs
Store \(T_z\) to Mtz
Store Eofz to meofz

Endcase

Store 0 to EOQ,MOST,MROP,MSL,MRO
Store \(\sqrt{2\ast \text{Rdem}\ast (\text{Orderc}+\text{Scost}\ast \text{Meofz})}/\text{Holdc}\) to EOQ
Store Mleadt1 to Most
Store \(\text{Rdem}\ast Mleadt1/12 + \text{Mtz}\ast \text{Std}\ast \sqrt{\text{Mleadt}}\) to Mrop
Store \(\text{Mrop} = (\text{Rdem}\ast Mleadt1/12)\) to Msl
Store EOQ\ast Mrop to Mro
Store "Calculated" to title
Store "Do you want to change the old records with this?:" to ctitle
Select C
Store Ltrim(str(MROP)) to Mrop
Store Ltrim(str(MSL)) to Msl
Store Ltrim(str(MRO)) to Mro
Store Ltrim(str(MOST)) to Most
Seek stockn
Set format to EOQ

Read 0

If check = "Y"

Set format to
Replace RO with Val(Mro)
Replace Rop with Val(Mrop)
Replace SL with Val(Msl)
Replace OST with Val(Mhost)
Replace date with today
Exit

Else

Set format to
Store 0 to Frate,Orderc,Scost,Serv1
Store "N" to CSC,CSN
Store "X" to check

Endif

136
Enddo
Close databases
Set decimals to
Return

*-------------------------* Eof EOQ.prg *-------------------------*

J. ANPROC

********************************************************************************
PROC.PRG  ********************************************************************************
********************************************************************************
Module name....: Proc.prg
Author..........: Park, Taeyong
Date............: Sept 30, 1987
Purpose.........: Procedures for calaulating statistics
Variables used.:*
Public..: FieldName
Local...: Max, Min, Std,Var
********************************************************************************

PROCEDURE Max
PARAMETERS fieldname,Condition
SET FILTER TO &Condition
GO TOP
Max = -99999
DO While .NOT. EOF()
   IF &FieldName > Max
      Max = &FieldName
   Endif
   Skip
ENDDO
SET FILTER TO RETURN

PROCEDURE Min
PARAMETERS FieldName,Condition
SET FILTER TO &Condition
GO TOP
Min = 99999
DO While .NOT. EOF()
   IF &FieldName < Min
      Min = &FieldName
   Endif
   Skip
ENDDO
SET FILTER TO RETURN

PROCEDURE Var
PARAMETERS FieldName,Condition
SET FILTER TO &Condition
GO TOP
COUNT TO N
SUM(&FieldName) TO TOT
SUN(&FieldName 2) TO TOTsq
Correction = TOT 2/N
Var = (TOTsq - Correction)/(n-1)
SET FILTER TO RETURN

PROCEDURE Std
PARAMETERS FieldName,Condition
SET FILTER TO &Condition
GO TOP
COUNT TO N
SUM(&FieldName) TO TOT
SUM(&FieldName 2) TO TOTsq
Correction = TOT 2/N
Variance = (TOTsq - Correction)/(n-1)
Std = SQRT(Variance)
SET FILTER TO
RETURN
PROCEDURE Mzulu
PARAMETER XDATE
STORE "01/01/"+substr("&XDATE",7,2) to Z1day
STORE (CTOD("&XDATE")-CTOD(Z1day)+1) TO Z2DAY
STORE SUBSTR("&XDATE",8)+SUBSTR(STR(1000000+Z2day),8,3) to MZULU
RETURN
*-----------------------------* EOF ANPROC.Procedure*-----------------------------*
5. MANAGEFL

a. MANAGEFL MENU

******************************************************************************
* Program: Managefl.PRG *
* Author...: PARK, TAEYONG *
* Date.....: AUG 21, 1987 *
* Notice...: Copyright 1987 *
* Notes....: Menu file for file management system, called by pmain.prg *
******************************************************************************

Do while .T.
check = "X"
CLEAR
@ 1,15 TO 3,55 double
@ 4,1 TO 23,77
@ 6,3 TO 7,20
@ 6,5 SAY SPACE(15)
@ 6,41 TO 12,75
@ 6,42 TO 7,59
@ 6,43 SAY SPACE(16)
@ 14,41 TO 21,75
@ 13,42 TO 15,59
@ 14,43 SAY SPACE(16)
@ 2,17 SAY "Management of files"
@ 6,6 SAY "Add files"
@ 6,44 SAY "Addition"
@ 14,44 SAY "Information"
@ 8,47 SAY "1.To add Customer (CI)"
@ 9,47 SAY "2.To add Property (SN)"
@ 10,47 SAY "3.To add ASL (SN)"
@ 9,57 SAY "4.To change Customer (CI)"
@ 11,57 SAY "5.To change Property (SN)"
@ 13,57 SAY "6.To change ASL (SN)"
@ 15,57 SAY "7.To change Master (SN+REQNO)"
@ 17,57 SAY "8.To change Batch (SN+REQNO)"
@ 20,57 SAY "9.Query on files"
@ 16,47 SAY "Today is ....."
@ 15,48 SAY today
@ 18,43 SAY "Stock number :
@ 19,55 SAY "(Or/And)"
@ 20,43 SAY "Customer code :
SET COLOR TO N/W
@ 18,59 SAY Stockn pict "9999-9999-9999-9999"
@ 20,59 SAY mci pict "9999"
SET COLOR TO
STORE "" TO SEL
@ 22,58 SAY SEL
@ 22,8 SAY "Enter Selection (1 - 9, or 0 to go to mainmenu) : ":
* .........The following lines are for select loop
i=0
DO WHILE i=0
   i=INKEY()
   @ 22,58 SAY ""
   IF UPPER(CHR(i))$"0123456789"
      @ 22,58 SAY CHR(i)
   EXIT
ENDIF
i=0
ENDDO
Do case
Case CHR(I) = '1'
   @ 18,59 SAY "XXXX-XX-XXX-XXXX"
   @ 20,59 get mci pict "9999"
READ
DO ADDCUST
  Case CHR(I) = '2'
    @ 18, 59 get stockn pict "9999-99-999-9999"
    @ 20, 59 say "XXXX"
    READ
    DO ADDPROP
  Case CHR(I) = '3'
    @ 18, 59 get stockn pict "9999-99-999-9999"
    @ 20, 59 say "XXXX"
    READ
    DO ADDASL
  Case CHR(I) = '4'
    @ 18, 59 say "XXXX-XX-XXX-XXXX"
    @ 20, 59 get mci pict "9999"
    READ
    DO EDITCUST
  Case CHR(I) = '5'
    @ 18, 59 get stockn pict "9999-99-999-9999"
    @ 20, 59 say "XXXX"
    READ
    DO EDITPROP
  Case CHR(I) = '6'
    @ 18, 59 get stockn pict "9999-99-999-9999"
    @ 20, 59 say "XXXX"
    READ
    DO EDITASL
  Case CHR(I) = '7'
    @ 18, 42 clear to 20, 65
    @ 18, 43 say "Stock number: " get stockn pict;
    "9999-99-999-9999"
    @ 19, 43 say "Customer code: " get mci pict "9999"
    @ 20, 43 say "Transaction type: " get mtype pict "!!"
    READ
    DO EDITMAST
  Case CHR(I) = '8'
    @ 18, 42 clear to 20, 65
    @ 18, 43 say "Stock number: " get stockn pict;
    "9999-99-999-9999"
    @ 19, 43 say "Customer code: " get mci pict "9999"
    @ 20, 43 say "Transaction type: " get mtype pict "!!"
    READ
    DO EDITBAT
  Case CHR(I) = '9'
    Do Manageg
    CHECK = "I"
    CASE I=27 .OR. CHR(I) = '0'
    ? Chr(7)
    RETURN
    OTHERWISE
    ? CHR(7)
  ENDCASE
ENDDO
CLOSE DATABASES
RETURN

*-----------------------------* Eof: Managefl.prg *-----------------------------*

b. ADDASL

*******************************************************************************
****************** ADDASL.PRG ******************
*******************************************************************************
* Module name.....: ADDASL.prg
* Author..........: Park, Taeyong
* Date...........: Aug 25, 1987
* Purpose.........: Add Authorized Storage List item (ASL) into
* ASL file by manual
* Called by......: MANAGEFL.prg

140
* Modules called : None
* Variables used..:
* Public.: CHECK.

If stockn = ""
    Return
Endif
Select A
Use ASL INDEX INasl
Select B
Use property INDEX Snproper
Select A
*-------- Set up loop for adding new ASL.
Store "X" to dcheck, fcheck
Dloop = .T.
Do while Dloop
    Clear
    *------- check to see if stock number already exists.
    Seek StockN
    Do case
        Case found()
            @15,10 say SN + " is Already exist in ASL file!"
            @17,20 say " Do you want to change ? " get check
            Read
            Check = upper (check)
            If check = "Y"
                Clear
                Set format to Editasl
                Read
            Endif
        Case .not. found()
            Clear
            Append blank
            Replace SN with StockN
            Replace date with date()
            Do while .T.
                Set format to ASL
                Read
                If upper(dcheck) = "Y"
                    Set format to Exit
                Endif
            Enddo
        Endcase
    Store "X" to tcheck
    Clear
    @17,20 say " Do you want to check property file?" get tcheck
    Read
    Tcheck = upper(tcheck)
    If Tcheck = "Y"
        Select B
        Clear
        Seek StockN
        Do case
            Case found()
                @15,15 say StockN + " is already exist in property file "
                @17,20 say " Do you want to change ? " get tcheck
                Read
                Tcheck = upper(tcheck)
                If tcheck = "Y"
                    Clear
                    Set format to Editprop
                    Edit recno()
                    Set format to
                Endif
            Dloop = .F.
        Case .NOT. found()
Clear
Append blank
Replace SN with StockN
Replace date with date()
Do while .T.
  Set format to prop
  Read
  fcheck = upper(fcheck)
  If fcheck = "Y"
    Replace tvalue with unitcost * onhand
    Set format to
    Dloop = .F.
  Exit
Endif
Enddo
Endif
Dloop = .F.
Enddo
Close all
Store space(16) to stockn
Release all
Return
********************************************************************
********************************************************************
EOF Addasl.prg
********************************************************************
********************************************************************
c. ADDPROP

********************************************************************
********************************************************************
********************************************************************
********************************************************************
ADDPREP.PRG
********************************************************************
********************************************************************
********************************************************************
********************************************************************
* Module name....: ADDPROP.prg *
* Author..........: Park, Taeyong *
* Date...........: Aug 20. 1987 *
* Purpose.........: Add new properties into property file *
* Called by.......: MANAGEFL.prg *
* Modules called : None *
* Variables used.: *
* Public.: TODAY *
********************************************************************
********************************************************************
********************************************************************
********************************************************************
If StockN = " "
  Return
Endif
Clear
Select A
Use ASL INDEX INasl
Select B
Use property INDEX Snproper
Today = date()
check = "X"
stock = .T.
Do while Stock
  clear
  Seek StockN
  Store "X" to fcheck
  Do case
    Case found()
      clear
      Store "X" to tcheck
      @ 15,10 say StockN + " is already exist in property file"
      @ 17, 20 Say "Do you want to change it NOW ?" get tcheck
      Read
      Tcheck = upper(tcheck)
      If tcheck = "N"
        clear
        Stock = .F.
        Exit
  Endcase
  Endif
  Dloop = .F.
Enddo
Home
Endif
Set format to Editprop
Edit recno()
Replace tvalue with unitcost * onhand
Set format to
Case .NOT. found()
Append blank
Replace SN with StockN
Replace date with date()
Do while .T.
  Set format to prop
  Read
  fcheck = upper(fcheck)
  If fcheck = "Y"
    Replace tvalue with unitcost * onhand
    Set format to
    Stock = .F.
    Exit
Endif
Enddo
Endcase
Enddo
Cl ear
Store "X" to acheck
@ 15, 20 SAY "Is this " + stockN + " ASL item?" get acheck
Read
Acheck = upper (acheck)
If acheck = "Y"
  Select A
  Seek stockN
  Store "Y" to dcheck
  Do case
    Case found()
      @ 20,10 say SN + " is Already exist !,;
      Do you want to change ?" get check
      Read
      Check = upper (check)
      If check = "Y"
        Set format to EDITASL
        Edit recno()
        Set format to
        Exit
      Endif
    Case .NOT. found()
      Append blank
      Replace SN with StockN
      Replace date with date()
      Do while .T.
        Set format to ASL
        Read
        Dcheck = upper(dcheck)
        If dcheck = "Y"
          Set format to
          Exit
        Endif
      Enddo
      Enddo
Endcase
Endif
Enddo
Close all
Store space(16) to stockn
Release all
Return
*-----------------------* Eof Addprop.prg *-----------------------*

143
d. ADDCUST

******************************************************************************
******************************************************************************
******************************************************************************
******************************************************************************
******************************************************************************

* Module name.....: ADDCUST.prg
* Author..........: Park, Taeyong
* Date...........: Aug 25. 1987
* Purpose........: Add customer into customer file
* Called by.......: MANAGEFL.prg
* Modules called : None
******************************************************************************
******************************************************************************
******************************************************************************
******************************************************************************
******************************************************************************

Use Customer index Ccust
Cust= "T",
Store "Y" to Dcheck
Do while Cust
   *------ check proposed customer code
   Seek Mci
   Do case
      Case Mci = " "
         Clear
         Cust = .F.
      Case found()
         @ 10,15 clear to 15,65
         @ 10,15 to 15,65
         @ 12,20 say "Customer code " + Mci + " Already exists !"
         ? chr(7)
         Mci = space(4)
         Wait " "
      Case .not. found()
         Append blank
         Replace CI with mcI
         Replace date with date() 
         Do while .T.
            Set format to cust
            Read
            If dcheck = "Y"
               Set format to
               Replace fundoh with fund - expend
               McI = space(4)
               Exit
            Else
               Chr(7)
            Endif
         Enddo
      Endcase
   Enddo
Store mtype to mtypel
If mtypel <> " "
   Return
Endif
Close all
Release all
Return
******************************************************************************
******************************************************************************
******************************************************************************
******************************************************************************
******************************************************************************


e. EDITASL

******************************************************************************
******************************************************************************
******************************************************************************
******************************************************************************
******************************************************************************

* Module name.....: EDITASL.prg
* Author..........: Park, Taeyong
* Date...........: Aug 28. 1987
* Purpose........: Edit(delete or change) contents of entry
******************************************************************************
******************************************************************************
******************************************************************************
******************************************************************************
******************************************************************************
Use ASL index INASL
Clear
More = .T.
Do while More
Seek STOCKN
Do case
  Case STOCKN = " ":
   More = .F.
  Case found():
   Set format to EDITASL
   Edit recno()
   Set format to
   Store "Y" to check
   @ 15, 20 Say "Pack marked records NOW ? (Y/N)" get check pict "!
   Read
   If check = "Y"
   Set talk on
   Pack
   Set talk off
   Stockn = space(16)
   more = .F.
   Endif
  Case .NOT. found():
   @ 15,30 Say "No such STOCK NUMBER code in the file !"
   More = .F.
   Stockn = space(16)
   Wait
  Endcase
Enddo (while more)
Close all
Release all
Return
*---------------------* Eof EDITPROP.prg *---------------------*
Return
Case stockn # " .AND. Mvnc = " 
  Search = stockn
Case stockn # " .AND. Mvnc # " 
  Search = stockn + upper(mvnc)
Case stockn = " .AND. Mvnc # " 
  Search = upper(Mvnc)
Endcase
Seek search
If found()
  Set format to EDITBAT
  Edit recno()
  Set format to
  Store "Y" to check
  @ 15, 20 Say "Pack marked records NOW ? (Y/N)" get check pict "!
  Read
  If check = "Y"
    Set talk on
    Pack
    Set talk off
    Stockn = space(16)
    more = .F.
  Endif
Else
  @ 15, 10 Say "No such STOCK NUMBER or Voucher Number in the file !"
  More = .F.
  Stockn = space(16)
  Wait
Endif
Enddo
Close databases
Release all
Return
*------------------------* Eof Editbat.prg *------------------------*

g. EDITCUST

******************************************************************************
**************************** EDITCUST.PRG *****************************
******************************************************************************
* Module name .....: EDITCUST.prg
* Author ...........: Park Taeyong
* Date ............: Aug 28, 1987
* Purpose ..........: Edit Customer record from customer file
* Called by .......: MANAGEFL.prg
* Modules called : EDITCUST.fmt
* Variables used ..:
* Local ...: MCI
******************************************************************************

Use CUSTOMER index CICUST
Clear
More = .T.
Do while More
  Seek MCI
  Do case
    Case mci = " 
      More = .F.
  Case found()
    Set format to EDITCUST
    Edit recno()
    Set format to
    Store "Y" to check
    @ 15, 20 Say "Pack marked records NOW ? (Y/N)" get check pict "!
******************************************************************************

146
Read
If check = "Y"
Set talk on
Pack
Set talk off
Mci = space(4)
more = .F.
Endif
Case .NOT. found()
@ 15,35 Say "No such customer code in file !"
More = .F.
Mci = space(4)
Wait
Endcase
Enddo (while more)
Close all
Return
*-----------------------* Eof EDITcust.prg *-----------------------*

h. EDITMAST

******************************************************************************
******************************************************************************
******************************************************************************
* Module name..... EDITMAST.prg *
* Author........... Park, Taeyong *
* Date............ Aug 28. 1987 *
* Purpose......... Edit master file *
* Called by....... MANAGEFL.prg *
* Modules called : EDITMAST_fmt *
* Variables used... *
* Local...: STOCKN, MCI *
******************************************************************************
Use MASTER index SCTMAST
Clear
@ 10,10 to 15,62
@ 11,30 Say "Warning!"
@ 13,15 Say "This process is not allowed to every person."
@ 14,15 Say "Enter password to continue." get mpass pict "!!!"
Read
If pass <> mpass
Close all
Return
Endif
Set filter to SN=stockn .AND. ci=mci
Clear
@ 10,10 Say "Enter the transaction type : " get mtype Pict "!!"
Read
More = .T.
Do while More
Seek Mtype
Do case
   Case STOCKN = ""
   More = .F.
   Case found()
   Set format to EDITMAST
   Edit recno()
   Set format to
   Store "Y" to check
   @ 15, 20 Say "Pack marked records NOW ? (Y/N)" get check pict "!
   Read
   If check = "Y"
   Set talk on
Pack
Set talk off
Mci = space(4)
Stockn = space(16)
more = .F.
Endif
Case .NOT. found()
@ 15,35 Say "No such STOCK NUMBER code in file !"
More = .F.
Stockn = space(16)
Wait
Endcase
Enddo (while more)
Set filter to
Close all
Release all
Return
*---------------------* Eof EDITMAST.prg *---------------------*

i. EDITPROP

******************************************************************************
** EDITPROP.PRG ******************************************************************************
******************************************************************************
* Module name.....: EDITPROP.prg *
* Author..........: Park, Taeyong *
* Date...........: Aug 28. 1987 *
* Purpose.........: Edit property file *
* Called by.......: MANAGEFL.prg *
* Modules called : EDITPROP.fmt *
* Variables used... *
* Public.: *
* Local...: STOCKN,MCI *
******************************************************************************
Use PROPERTY index SNPROPER
Clear
More = .T.
Do while More
Seek STOCKN
Do case
Case STOCKN = " "
More = .F.
Case found()
Set format to EDITPROP
Edit recno()
Set format to
Store "Y" to check
@ 15, 20 Say "Pack marked records NOW ? (Y/N)" get check pict !
Read
If check = "y"
Set talk on
Pack
Set talk off
Mci = space(4)
STOCKN=SPACE(16)
more = .F.
Endif
Case .NOT. found()
@ 15,35 Say "No such STOCK NUMBER code in file !"
More = .F.
Mci = space(4)
Wait
Endcase
MANAGEQ

Close databases
Clear
Set talk off
set echo off
Use property
Index on Class to Clssprop
Use Property index snproper,Clssprop
Select B
Use ASL index inasl
Select A
Set relation to SN into B

Title = "Query on Stock number"
Mok="Y"
Store space(16) to stockn
Store "" to Mclass,Mconf
Do while .T.
Clear
@ 4,25 say title
@ 8,15 Say "Options"
@ 11,17 Say "1. Stock number"
@ 12,17 Say "2. Class"
@ 13,17 Say "3. All"
@ 15,17 Say "4. Return to main menu"
@ 17,15 Say "Enter Option: "
@ 2,1 to 23,75
Store "" to sel
@ 17,29 Get sel
i=0
Do while i=0
i=inkey()
If chr(i)$"1234"
 Exit
 Endif
i=0
Enddo
@ 17,29 Say Chr(i)
If chr(i)="4"
 Exit
 Endif
Do case
 Case chr(i)="1"
 @ 11,36 Say "Enter stock number ",
 Get Stockn pict "9999-999-9999-9999"
 Read
 If Stockn=" "
 ? Chr(7)

Loop
Endif
Set filter to $n='&Stockn'
Set Order to 1
Do setup
Report form stockrpl
Set filter to
Case chr(i)="2"
@ 12,36 Say "Enter Class Number:" Get Mclass pict "9"
@ 13,38 Say "01 for class 10"
Read
If Mclass=""
  ? Chr(7)
  Loop
Endif
Set filter to Class='&Mclass'
Set Order to 2
Do setup
Report form stockrpl
Set filter to
Case chr(i)="3"
@ 13,26 Say "< Is this your select(Y/N):" Get Mconf pict "!"
Read
If Mconf="!!"
  ? Chr(7)
  Loop
Endif
Set Order to 2
Do setup
Report form stockrpl
Otherwise
Loop
Endcase
If Upper(Mok)="Y"
  Set console on
  Set print off
  Set Order to 1
Else
  @ 24,17 Say "Press any key to continue..."
  Wait"
Endif
Clear
Enddo
Close databases
Release all
Erase classprop.ndx
Store space(15) to stockn
Clear
Return
*---------------------------------* Eof MANAGEQ.prg *---------------------------------*
APPENDIX F
SCREEN FORMAT

1. TRANSACT
   a. BATCH

   ************************************************************
   ************************************************************
   Module name ....: Batch.fmt ************************************************************
   ************************************************************
   @ 2, 11 SAY "Request for issue"
   @ 3, 47 SAY "Today ......"
   @ 3, 61 SAY today
   @ 5, 8 SAY "Requested From :"
   @ 5, 25 SAY CUSTOMER->CDESC
   @ 6, 8 SAY "Send To : The 150 Infantry Division"
   @ 8, 8 SAY "Type of transaction : (RD) Request No :"
   @ 8, 53 GET HREQNO PICTURE "9999!!-9999-9999"
   @ 9, 10 SAY "Request for Issue to Div (Customer's)"
   @ 12, 18 SAY "Stock Number :"
   @ 12, 32 GET BATCH->SN PICTURE "9999-99-999-9999"
   @ 14, 13 SAY "Description :
   @ 14, 32 GET PROPERTY->NM
   @ 16, 8 SAY "Send To The 150 Infantry Division"
   @ 16, 39 SAY "Price :
   @ 16, 47 GET PROPERTY->UNITCOST
   @ 18, 56 SAY "$"
   @ 18, 38 TO 20, 70
   *---------------------------------------* Eof.batch.fmt *---------------------------------------*

   b. MAST

   ************************************************************
   ************************************************************
   Module name ....: Mast.fmt ************************************************************
   ************************************************************
   @ 2, 9 SAY mtitle
   @ 3, 43 SAY "Today is...
   @ 3, 56 SAY Today
   @ 5, 7 SAY mhost + mcdesl
   @ 6, 7 SAY "$ "+mcust + mdesc2
   @ 8, 6 SAY "Type of transaction : Voucher No :"
   @ 8, 48 GET VNI PICTURE "9999!!-9999-9999"
   @ 9, 6 SAY Mtypea
   @ 9, 34 SAY "Request Number :"
   @ 9, 50 GET MREQNO PICTURE "9999!!-9999-9999"
   @ 12, 19 SAY "Stock Number :"
   @ 12, 32 GET MASTER->SN PICTURE "9999-99-999-9999"
   @ 14, 19 SAY "Description :
   @ 14, 32 GET PROPERTY->NM
   @ 16, 8 SAY "Send To The 150 Infantry Division"
   @ 16, 39 SAY "Price :
   @ 16, 47 GET PROPERTY->UNIT FUNCTION "/AAA" PICTURE "XXXX"
   @ 18, 38 TO 9, 38
   *---------------------------------------* Eof.mast.fmt *---------------------------------------*
**c. TURNIN**

```
@ 2, 11 SAY "Request for turn-in"
@ 3, 46 SAY "Today......"
@ 5, 6 SAY "From :"
@ 6, 8 SAY "To :"
@ 6, 13 SAY MCDESC1
@ 8, 6 SAY MCDESC2
@ 9, 10 SAY "Type of transaction : (ID) "
@ 9, 52 GET BATCH->REQNO PICTURE "9999!!-9999-9999"  
@ 11, 21 SAY "Stock number :"
@ 11, 37 GET BATCH->SN PICTURE "9999-99-9999"
@ 13, 21 SAY "Description :"
@ 13, 37 GET PROPERTY->NM
@ 15, 16 SAY "Unit :"
@ 15, 25 GET PROPERTY->UNIT PICTURE "!XXX"
@ 15, 37 SAY "Reuseable? :"
@ 17, 15 SAY "Price :"
@ 17, 25 GET PROPERTY->UNITCOST
@ 17, 37 SAY "Quantity :"
@ 20, 34 SAY "Is this record correct ?" Get Zcheck pict "!
@ 1, 4 TO 3, 40
@ 4, 4 TO 19, 70
@ 7, 5 TO 7, 69
@ 4, 36 TO 4, 36
@ 4, 37 TO 4, 37
@ 10, 5 TO 10, 69
@ 8, 37 TO 9, 37
```

---

**d. CANCMAST**

```
@ 2, 9 SAY "Cancel Request for issue item"
@ 3, 44 SAY "Today :"+Dtoc(date())+"("+Zulu+")"
@ 6, 25 SAY "Stock number :"
@ 6, 40 SAY MASTER->SN
@ 8, 25 SAY "Customer Code:"  
@ 8, 40 SAY MASTER->CI
@ 10, 24 SAY "Request number:"  
@ 10, 40 SAY MASTER->REQNO
@ 13, 13 SAY "Quantity :"
@ 13, 24 SAY MASTER->QTY
@ 13, 33 SAY "Price :"
@ 13, 41 SAY MASTER->UNITCOST
```

---

152
@ 13, 51 SAY "Date : (" 
@ 13, 58 SAY MASTER->DATE 
@ 13, 62 SAY ")" 
@ 17, 14 SAY "Is this record what you want to cancel?(Y/N)" ; 
Get checks pict ")" 
@ 1, 6 TO 3, 40 
@ 4, 5 TO 15, 70 
@ 16, 5 TO 18, 70 
*-----------------------------------------* Eof CANCMAST.fmt *-----------------------------------------*

2. MANAGEFL 

a. ASL 

******************************************************************************* 
******************************************************************************* Module name....: Asl.fmt******************************************************************************* 
******************************************************************************* 
@ 2, 8 SAY "Authorized storage list file" 
@ 3, 45 SAY "Today is ......." get today 
@ 3, 4 SAY "Stock number :" 
@ 6, 44 GET Stockn PICTURE "9999-99-999-9999" 
@ 9, 20 SAY "Reorder Point:" 
@ 9, 44 GET ASL->ROP 
@ 11, 20 SAY "Safety Level:" 
@ 11, 44 GET ASL->SL 
@ 13, 20 SAY "Requisition Objective:" 
@ 13, 44 GET ASL->RO 
@ 16, 20 SAY "Order Shipping Time:" 
@ 16, 44 GET ASL->OST 
@ 16, 50 SAY "days:" 
@ 18, 18 SAY "Resource control number :" 
@ 18, 44 GET ASL->RCN PICTURE "9999" 
@ 23, 35 SAY " Is this record correct? : " get dcheck pict ")" 
@ 1, 5 TO 3, 40 
@ 4, 4 TO 20, 70 
*-----------------------------------------* Eof.As1.fmt *-----------------------------------------*

b. CUST 

******************************************************************************* 
***** Cust.fmt (CUSTOMER file screen format called by ADDCUST)******************************************************************************* 
******************************************************************************* 
@ 2, 13 SAY "Customer File" 
@ 3, 43 SAY "Today is ...........
@ 3, 62 SAY CUSTOMER->DATE 
@ 6, 13 SAY "Customer Code" 
@ 6, 28 SAY CUSTOMER->CI 
@ 6, 47 SAY "Priority" 
@ 6, 57 GET CUSTOMER->PRIORITY 
@ 8, 22 SAY "Name:" 
@ 8, 28 GET CUSTOMER->CDESC 
@ 10, 19 SAY "Address" 
@ 10, 28 GET CUSTOMER->ADDRESS FUNCTION "S30": 
PICTURE "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX" 
@ 12, 23 GET CUSTOMER->ZIPCODE 
@ 12, 37 SAY "Zip code:" 
@ 15, 18 SAY "Fund : 
@ 15, 28 GET CUSTOMER->FUND PICTURE "99999999.99" 
@ 16, 40 SAY "S Allowed:" 
@ 17, 28 GET CUSTOMER->EXPEND PICTURE "99999999.99" 
@ 17, 40 SAY "S Expediture" 
@ 18, 22 SAY "------------------------"
c. PROP

************************************************************************** Module name: Prop.fmt**************************************************************************
@s 2, 12 SAY "Property Book file"
@s 3, 44 SAY "Today............"
@s 3, 62 SAY Today
@s 6, 22 SAY "Stock number ;"
@s 6, 38 GET Stockno PICTURE "9999-99-999-9999"
@s 7, 22 SAY "Serial NO ;"
@s 7, 38 GET PROPERTY->SERIALNO
@s 8, 38 GET PROPERTY->NM
@s 8, 22 SAY "Description ;"
@s 10, 25 SAY "Unit ;"
@s 10, 38 GET PROPERTY->UNIT
@s 10, 46 SAY "Class ;"
@s 10, 54 GET PROPERTY->CLASS PICTURE "99"
@s 13, 25 SAY "On hand ;"
@s 13, 38 GET PROPERTY->ONHAND
@s 13, 25 SAY "Price ;"
@s 13, 38 GET PROPERTY->UNITCOST
@s 15, 47 SAY "S"
@s 16, 25 SAY "-----------------------"
@s 18, 20 SAY "Is this combat essential item ? ;"
@s 23, 35 SAY "Is this record correct ? ;" get fcheck pict "!"
@s 1, 4 TO 3, 40
*s-----------------------------*

j. EDITBAT

************************************************************************** Editbat.fmt (BATCH file screen format called by EDITBAT)**************************************************************************
@s 2, 7 SAY "Edit(change or delete) Batch file"
@s 3, 46 SAY "Today............"
@s 3, 61 GET BATCH->DATE
@s 5, 23 SAY " Stock number ;"
@s 5, 40 GET BATCH->SN
@s 7, 23 SAY "Voucher number ;"
@s 7, 40 GET BATCH->REQNO
@s 9, 23 SAY " Customer Code ;"
@s 9, 40 GET BATCH->CI
@s 11, 14 SAY "Type of action ;"
@s 11, 35 GET BATCH->TYPE
@s 11, 50 SAY " Posted ;"
@s 11, 60 GET BATCH->POSTED
@s 13, 14 SAY " Quantity ;"
@s 13, 35 GET BATCH->QTY
@s 13, 50 SAY "Unitcost ;"
@s 13, 60 GET BATCH->UNITCOST
@s 13, 69 SAY "S"
@s 14, 35 SAY "Date : +Dtoc(BATCH->DATE)
@s 17, 7 SAY "(Insert mode) : Ins (Record) Next : PgDn"
@s 17, 7 SAY "(Delete) Character :Del Previous : PgUp"
@s 19, 13 SAY "Field: ctrl+Y Record:ctrl+ U (Done/Save) ;
End Abandon :Esc"
@s 1, 5 TO 3, 40
c. EDITCUST

******************************************************************************
***** Editcust.fmt (MASTER file screen format called by EDITMAST)******
******************************************************************************
@ 2, 10 SAY "Edit/Change Master file"
@ 3, 47 SAY "Today..........."
@ 3, 61 SAY DTOC(DATE())
@ 5, 20 SAY "Stock Number :"
@ 5, 36 GET MASTER->SN PICTURE "9999-99999-99999"
@ 7, 14 SAY "Customer code :"
@ 7, 31 GET MASTER->CI
@ 7, 43 SAY "Action Type :"
@ 7, 58 GET MASTER->TYPE
@ 9, 13 SAY "Request Number :"
@ 9, 31 GET MASTER->RENO PICTURE "99999999999999999"
@ 10, 50 SAY "Customer's :"
@ 12, 12 SAY "Quantity :"
@ 12, 24 GET MASTER->QTY
@ 12, 31 GET MASTER->RE NO PICTURE "99999999999999999"
@ 14, 58 GET MASTER->MISC
@ 14, 59 SAY "Date :"
@ 14, 36 GET MASTER->DATE
@ 17, 7 SAY "(Insert mode) : Ins (Record) Next : PgDn"
@ 18, 7 SAY "(Delete) Character :Del Previous : PgUp"
@ 19, 13 SAY "Field:ctrl+Y Record: ctrl+U (Done/Save) ;
ctrl+End Abandon :Esc"
@ 1, 5 TO 3, 40
@ 4, 5 TO 15, 70
@ 17, 35 TO 19, 35
******************************************************************************

f. EDITASL

******************************************************************************
***** Editasl.fmt (ASL screen format called by EDITASL)******
******************************************************************************
@ 2, 6 SAY "Edit Authorized Storage List file"
@ 3, 47 SAY "Today..........."
@ 3, 60 SAY DTOC(DATE())
@ 5, 20 SAY "Stock number :"
@ 5, 36 GET ASL->SN
@ 8, 10 SAY "ReOrder Point :"
@ 8, 26 GET ASL->ROP
@ 8, 38 SAY "Order shipping time :"
@ 8, 60 GET ASL->OST
@ 8, 64 SAY "days"
@ 10, 11 SAY "Safety Level :"
@ 10, 26 GET ASL->SL
@ 10, 38 SAY "Date :"
@ 10, 60 GET ASL->DATE
@ 12, 10 SAY "Requisition Objective :"
@ 12, 34 GET ASL->RO
@ 14, 23 SAY "Resource control number :"
@ 14, 50 GET ASL->RCN
@ 17, 7 SAY "(Insert mode) : Ins (Record) Next : PgDn"
g. EDITPROP

*********************************************
*** Editprop.fmt (PROPERTY file screen format called by EDITPROP)********
*********************************************
@ 2, 12 SAY "Edit property file"
@ 3, 46 SAY "Today ......."
@ 3, 61 SAY DTOD(TODAY)
@ 3, 7 SAY "Stock Number :"
@ 5, 23 GET PROPERTY->SN
@ 5, 45 SAY "Serial No :"
@ 5, 57 GET PROPERTY->SERIALNO
@ 7, 7 SAY "Nomenclature :"
@ 7, 23 GET PROPERTY->NM
@ 9, 15 SAY "Unit :"
@ 9, 23 GET PROPERTY->UNIT
@ 9, 30 SAY "Essential item? :"
@ 9, 43 GET PROPERTY->ESSENCE
@ 9, 54 SAY "Class :"
@ 9, 62 GET PROPERTY->CLASS
@ 12, 14 SAY "Onhand :"
@ 12, 23 GET PROPERTY->ONHAND
@ 12, 35 SAY "Price :"
@ 12, 49 GET PROPERTY->UNITCOST
@ 12, 59 SAY "$"
@ 14, 30 SAY "Total Value :"
@ 14, 49 GET PROPERTY->TVALUE
@ 14, 59 SAY "$"
@ 17, 7 SAY "(Insert mode) : Ins (Record) Next : PgDn"
@ 18, 7 SAY "(Delete) : Del Previous : PgUp"
@ 19, 13 SAY "Field: ctrl+Y Record: ctrl+U (Done/Save) : ;
ctrl+End Abandon : Esc"
@ 1, 5 TO 3, 40
@ 4, 5 TO 15, 70
@ 16, 5 TO 20, 70
@ 17, 35 TO 19, 35
*--------------------------- Eof EDITPROP.fmt *-------------------------*

h. EDITCUST

*********************************************
*** Editcust.fmt (CUSTOMER file screen called by EDITCUST)********
*********************************************
@ 2, 9 SAY "Edit or Delete CUSTOMER file"
@ 3, 46 SAY "Today .......
@ 3, 60 SAY TODAY
@ 3, 9 SAY "Customer :
@ 6, 20 GET CUSTOMER->CI
@ 5, 25 SAY "code Priority :
@ 5, 48 GET CUSTOMER->PRIORITY
@ 7, 20 GET CUSTOMER->CDESC
@ 7, 51 SAY "Description"
@ CUSTOMER->ADDRESS FUNCTION "S30"
PICTURE "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
@ 8, 51 SAY "Address"
@ 9, 20 GET CUSTOMER->ZIPCODE
@ 9, 51 SAY "Zipcode"
3. ANALYSIS

a. EOQ

@ 11, 9 SAY "Fund allowed;"
@ 12, 23 GET CUSTOMER->FUND
@ 12, 36 SAY "Expend"
@ 13, 23 GET CUSTOMER->FUNDOH
@ 13, 36 SAY "On hand"
@ 14, 33 SAY "Last edit date;"
@ 15, 51 GET CUSTOMER->DATE
@ 17, 7 SAY "(Insert mode): Ins (Record) Next : PgDn"
@ 18, 7 SAY "(Delete): Character : Del Previous : PgUp"
@ 19, 13 SAY "Field: ctrl+Y Record: ctrl+U  (Done/Save) ;

@ 2, 17 SAY "UNIT MAT
I0N for A N A L Y S I S"
@ 5, 40 SAY PROPERTY->SN
@ 6, 17 SAY "+Rtrim(PROPERTY->NM)+"; "+"Unit;");
+Rtrim(PROPERTY->UNIT)+" Class:"+PROPERTY->CLASS+");
"$/"+Ltrim(PROPERTY->UNIT)
"$/"+Ltrim(str(property->unitcost));
@ 10, 21 SAY "Annual Demand(R) : "+Ltrim(str(Rdem)){" ";
+Rtrim(property->unit)="/"/year"
@ 12, 14 SAY "Lead time(OST) in Month : "+Ltrim(str(mLeadt));
"/Month(s)"
@ 14, 21 SAY "Ordering cost(C) : $/order"
@ 15, 40 GET Orderc pict "9999.99"
@ 15, 40 SAY "Holding cost unit per year ; 
"%
@ 16, 5 SAY "Select one of these"
@ 17, 14 SAY "Stockout cost(If Known) ; 
$/unit"
@ 17, 40 GET Scost pict "9999.99"
@ 17, 55 SAY "Select?" get CSC pict "!"
@ 18, 16 SAY "Service Level in Year ; 
"%
@ 18, 40 GET Servl pict "99,9999"
@ 18, 5 SAY "Select?" get CSL pict "!"
@ 21, 5 SAY "+title+" Reorder point ; +Mrop
@ 21, 45 SAY "Safety Level ; + Msl
@ 22, 15 SAY "Requisition objective ; + Mro
@ 22, 45 SAY "Lead Time ; "+Host
@ 24, 5 SAY space(50-Len(Ctitle))+Ctitle get check pict "!
@ 1, 13 TO 3, 51 DOUBLE
@ 4, 1 TO 19, 74 DOUBLE
@ 20, 1 TO 23, 74 DOUBLE

*------* Eof EDITCUST.fmt *--------------------------*

*------* Eof EOQ.fmt *--------------------------*
LIST OF REFERENCES


## INITIAL DISTRIBUTION LIST

<table>
<thead>
<tr>
<th>No.</th>
<th>Copies</th>
<th>Name and Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>2</td>
<td>Defense Technical Information Center</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Cameron Station</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Alexandria, VA 22304-6145</td>
</tr>
<tr>
<td>2.</td>
<td>2</td>
<td>Library, Code 0142</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Naval Postgraduate School</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Monterey, CA 93943-5002</td>
</tr>
<tr>
<td>3.</td>
<td>1</td>
<td>Library, P.O.Box 77</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Gong Neung Dong, Dobong Goo</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Seoul 130-09, Republic of Korea</td>
</tr>
<tr>
<td>4.</td>
<td>1</td>
<td>Adjunct instructor Y.K. Mortagy</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Code 54M</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Department of Administrative Sciences</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Naval Postgraduate School</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Monterey, CA 93943</td>
</tr>
<tr>
<td>5.</td>
<td>1</td>
<td>Professor Thomas Moore</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Code 54 Mo</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Department of Administrative Sciences</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Naval Postgraduate School</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Monterey, CA 93943</td>
</tr>
<tr>
<td>6.</td>
<td>1</td>
<td>Kim, Nak-Heung</td>
</tr>
<tr>
<td></td>
<td></td>
<td>176-100 Pookahyun-Dong Sudaemoon-Koo</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Seoul, 120-00, Republic of Korea</td>
</tr>
<tr>
<td>7.</td>
<td>1</td>
<td>Kim, Jae Soo</td>
</tr>
<tr>
<td></td>
<td></td>
<td>#4-2, 375 Hwanghak-Dong Jung-Koo</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Seoul, 100-00, Republic of Korea</td>
</tr>
<tr>
<td>8.</td>
<td>1</td>
<td>Kim, Changwhan</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SMC 2889, Naval Postgraduate School</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Monterey, CA 93943</td>
</tr>
<tr>
<td>9.</td>
<td>1</td>
<td>Erdal Ozturk</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Gazipasa Mah. Zeytinlik Cad. No.5-15</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Trabzon Turkey</td>
</tr>
</tbody>
</table>
10. Park, Taeyong  
Kyungbuk kumrung-gun Umo-myun Okgye-dong 821  
Republic of Korea

11. Son, Gun ha  
Unpyung-gu Susaek-dong Kukdae-won (NDPS) Apt Ka-dong 306  
Seoul 120-99  
Republic of Korea
END
DATE
FILMED
6-88
DTIC