

NAVAL POSTGRADUATE SCHOOL Monterey, California



THESIS

B808776

DISTRIBUTION OF DATABASES ON CD-ROM

by

DOUGLAS EDWARD BRINKLEY

and

PAUL GORDON SCYOC

September, 1990

Thesis Advisor:

Barry A. Frew

Approved for public release; distribution is unlimited.

T251824

20100915185

REPORT DOCUMENTATION PAGE			
1a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED		1b. RESTRICTIVE MARKINGS	
2a. SECURITY CLASSIFICATION AUTHORITY		3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution is unlimited.	
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE			
4. PERFORMING ORGANIZATION REPORT NUMBER(S)		5. MONITORING ORGANIZATION REPORT NUMBER(S)	
6a. NAME OF PERFORMING ORGANIZATION Naval Postgraduate School	6b. OFFICE SYMBOL (If applicable) 37	7a. NAME OF MONITORING ORGANIZATION Naval Postgraduate School	
6c. ADDRESS (City, State, and ZIP Code) Monterey, CA 93943-5000		7b. ADDRESS (City, State, and ZIP Code) Monterey, CA 93943-5000	
8a. NAME OF FUNDING/SPONSORING ORGANIZATION	8b. OFFICE SYMBOL (If applicable)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER	
8c. ADDRESS (City, State, and ZIP Code)		10. SOURCE OF FUNDING NUMBERS	
		Program Element No	Project No
		Task No	Work Unit Accession Number
11. TITLE (Include Security Classification) Distribution of Databases on CD-ROM			
12. PERSONAL AUTHOR(S) Brinkley, Douglas Edward and Scyoc, Paul Gordon			
13a. TYPE OF REPORT Master's Thesis	13b. TIME COVERED From To	14. DATE OF REPORT (year, month, day) September 1990	15. PAGE COUNT 136
16. SUPPLEMENTARY NOTATION The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.			
17. COSATI CODES		18. SUBJECT TERMS (continue on reverse if necessary and identify by block number)	
FIELD	GROUP	SUBGROUP	
		CD-ROM; Database Design; Database Distribution; Optical Publishing; Pre-master.	
19. ABSTRACT (continue on reverse if necessary and identify by block number) <p>The vast majority of information distributed throughout the Navy today is done so using various paper products in the form of manuals, printouts, books and other publications. Paper publications are inherently expensive to produce, store and distribute on a large scale. They are also difficult to use when extracting detailed information for ad hoc reports. Compact Disc, Read Only Memory (CD-ROM) technology provides an alternative means to publish and distribute information to large numbers of users. This thesis examines the capabilities, advantages and disadvantages of CD-ROM technology to determine whether its use would be a preferred strategy for fulfilling information distribution requirements within the Navy. The research included the development of a CD-ROM pre-master file which was used to test the response of a CD-ROM application compared to manually extracting the same information from a printed manual.</p>			
20. DISTRIBUTION/AVAILABILITY OF ABSTRACT <input checked="" type="checkbox"/> UNCLASSIFIED/UNLIMITED <input type="checkbox"/> SAME AS REPORT <input type="checkbox"/> DTIC USERS		21. ABSTRACT SECURITY CLASSIFICATION	
22a. NAME OF RESPONSIBLE INDIVIDUAL Barry A. Frew		22b. TELEPHONE (Include Area code) (408) 646-2392	22c. OFFICE SYMBOL Code 05

Approved for public release; distribution is unlimited.

Distribution of Databases on CD-ROM

by

Douglas E. Brinkley
Lieutenant, Supply Corps, United States Navy
B.S., State University of New York at Albany, 1980

and

Paul G. Scyoc
Lieutenant, Supply Corps, United States Navy
B.B.A., University of North Florida, 1979

Submitted in partial fulfillment
of the requirements for the degree of


MASTER OF SCIENCE IN INFORMATION SYSTEMS

from the

NAVAL POSTGRADUATE SCHOOL

September 1990

Authors:

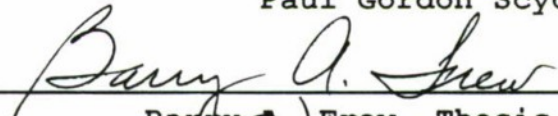


Douglas Edward Brinkley

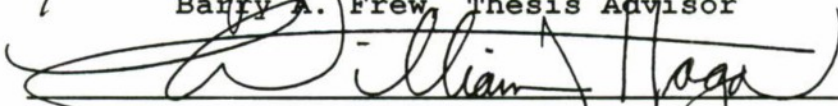


Paul Gordon Scyoc

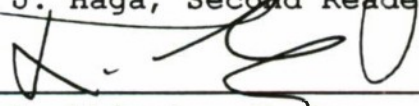
Approved by:



Barry A. Frew, Thesis Advisor



William J. Haga, Second Reader



David R. Whipple, Chairman
Department of Administrative Sciences

ABSTRACT

The vast majority of information distributed throughout the Navy today is done so using various paper products in the form of manuals, printouts, books and other publications. Paper publications are inherently expensive to produce, store and distribute on a large scale. They are also difficult to use when extracting detailed information for ad hoc reports.

Compact Disc, Read Only Memory (CD-ROM) technology provides an alternative means to publish and distribute information to large numbers of users. This thesis examines the capabilities, advantages and disadvantages of CD-ROM technology to determine whether its use would be a preferred strategy for fulfilling information distribution requirements within the Navy. The research included the development of a CD-ROM pre-master file which was used to test the response of a CD-ROM application compared to manually extracting the same information from a printed manual.

1 No 515
B808776
C. 1

TABLE OF CONTENTS

I. INTRODUCTION 1

 A. GENERAL 1

 B. OBJECTIVE 1

 C. RESEARCH METHODOLOGY 2

II. WHAT IS CD-ROM? 3

 A. BACKGROUND 3

III. ADVANTAGES AND DISADVANTAGES OF CD-ROM 7

 A. ADVANTAGES OF CD-ROM AS A STORAGE MEDIUM 7

 1. Storage Capacity 7

 2. Low Cost to Mass Produce 7

 3. Industry Standard Access to Data 8

 4. Relatively Low Cost Equipment Needed to Read
 Disks 8

 5. Data are Non-volatile 8

 6. Durable Medium 9

 7. Removable and Portable 9

 8. Low Transportation Costs 10

 9. Simplifies Security of Classified Material 10

 10. User Familiarity 10

B.	DISADVANTAGES OF CD-ROM AS A STORAGE MEDIUM . . .	10
1.	Slow Access Speeds	10
2.	Limited Number of CD-ROM Readers in Place Today	12
3.	Read Only Media	12
4.	Expensive Process When Dealing With Small Numbers	12
IV.	COST COMPARISON OF CD-ROM VERSUS PAPER	13
A.	SELECTION OF A SUITABLE SOURCE FILE	13
B.	HOW THE INFORMATION IS DISTRIBUTED TODAY	14
C.	CURRENT PRINTING AND SHIPPING COSTS	14
D.	COST TO PRODUCE AND SHIP ON CD-ROM	14
V.	DESIGN AND IMPLEMENTATION OF THE DATABASE APPLICATION	20
A.	DEFINITION PHASE	20
1.	Background of Problem Area	20
2.	Definition of the Problem	20
3.	Project Scope	21
B.	REQUIREMENTS PHASE	21
1.	Data Requirements	21
a.	User Environment	21
b.	Object Descriptions	22
c.	Methodology	24
2.	Application Requirements	25

a.	Data Flows/Processes	25
b.	Object Update/Display Mechanisms	25
C.	DESIGN PHASE	27
1.	Logical Database Design	27
2.	Application Design	28
a.	Control Mechanism	28
b.	Menu Hierarchy	29
c.	Materialization	32
D.	IMPLEMENTATION PHASE	36
1.	Database Implementation	36
2.	Data Preparation	37
VI.	PROS AND CONS OF THE CD-ROM APPLICATION AND THE PRINTED MANUAL	39
A.	CD-ROM APPLICATION PROS	39
1.	Data Protection	39
2.	Access Speed	39
3.	Accuracy	41
4.	Availability of Reports	41
5.	Data Manipulation/Extraction	41
6.	Standard Off-the-Shelf Software	42
B.	CD-ROM APPLICATION CONS	42
1.	Information Retrieval Performance	42
2.	Hardware Requirements	42
3.	Read Only	42
C.	PROS OF THE PRINTED LINEAL LIST MANUAL	43

1. Data Modification	43
2. Portability	43
D. CONS OF THE PRINTED LINEAL LIST MANUAL	43
1. Durability	43
2. Accuracy	43
3. Data Format	43
4. Information Retrieval Performance	44
VII. SUMMARY	45
A. CONCLUSION	45
B. RECOMMENDATIONS	46
Appendix A - THE PRE-MASTERING PROCESS	48
Appendix B - CD-ROM vs PAPER MANUAL TEST RESULTS	58
Appendix C - OBJECT DEFINITIONS FOR CODIS	60
Appendix D - UPDATE, DISPLAY AND CONTROL MECHANISMS	64
Appendix E - CODIS RELATION DEFINITION	66
Appendix F - dBASE IV FILE STRUCTURES	68
Appendix G - dBASE IV PROGRAM CODE	71
LIST OF REFERENCES	125
BIBLIOGRAPHY	126
INITIAL DISTRIBUTION LIST	127

I. INTRODUCTION

A. GENERAL

Many of the databases used by the Navy on a day to day basis are distributed on paper products in the form of manuals, catalogs, and printouts. Therefore, extracting desired information requires a tedious manual look-up procedure. Distributing the information on paper also prevents the user from using automated tools to manipulate or retrieve specific data for ad hoc queries and/or reports. Only recently has technology provided an alternative means to publish and distribute data to large numbers of users.

B. OBJECTIVE

The objective of this thesis is to evaluate the feasibility and methodology of using CD-ROM as a medium for publishing and distributing large databases. This research will include the development of a CD-ROM application and the associated 9-track pre-master tape. Secondary objectives are to document; 1) what is required to convert data from an existing digital media and format to a CD-ROM pre-mastering environment, 2) what are the advantages and disadvantages of using CD-ROM for distributing information, 3) what costs are associated with the production and distribution of CD-ROMs, and 4) how does the cost of distributing information on CD-ROM

compare with the printing and shipping costs of paper products.

C. RESEARCH METHODOLOGY

The methodology involved in this research began with a literature review of magazines, periodicals, and books to gain an understanding of the technology. The equipment required to perform the pre-mastering process was recently acquired by the Naval Postgraduate School. The system used was a CD Publisher model 88-600 from Meridian Data, Inc. (MDI) connected to a Compaq Deskpro 386 microcomputer. The data used to develop the CD-ROM application are extracted from the Navy Officer Master File and the Navy Standard Address File. Both sets of data were provided by the Defense Manpower Data Center (DMDC). During the course of the research both authors attended a CD-ROM conference in San Francisco to gain insight from industry experts.

Evaluation of the CD-ROM application was accomplished by timing manual searches of data from the Register of Commissioned and Warrant Officer of the U.S. Navy on Active Duty (commonly referred to as the Officer Lineal List) and comparing them with automated CD-ROM searches.

II. WHAT IS CD-ROM?

A. BACKGROUND

Compact Disc - Read Only Memory (CD-ROM) has the capacity to hold at least 680 megabytes of data. These data can include text files, computer programs, music, color images and animation. Disc readers are available today for most types of personal and business computers.

CD-ROM evolved from CD audio technology and it uses the same basic drive mechanisms and disc manufacturing processes. CD-ROM has developed from technological advances and economies of scale associated with the CD audio industry.

Another factor which expedited the development of CD-ROM was the adoption of industry standards, both physical and logical. All of the uses of Compact Disc, including CD-ROM and CD-I (Compact Disc Interactive), are based on CD audio specifications. These specifications are referred to by the colors of the binders in which they are published. They are:

CD Audio (including CD Graphics) - "The Red Book"

CD-ROM - "The Yellow Book"

CD-I - "The Green Book"

The physical standard refers to the physical characteristics and specifications for CD-ROMs. It also

describes how data are written to a disc. A disc is 120 mm (4.72 inches) in diameter with a 15 mm hole in the center, and is 1.2 mm thick. The disc is made from a clear plastic called polycarbonate. Information is stored on a disc in the form of pits and flat places between pits, called lands. The pits and lands are laid out on a spiral track which is read outward from the center of the disc. The total length of the track on a CD-ROM is almost 3 miles. The total number of pits is almost 2 billion [Ref. 1]. The pits are molded into the plastic which is then coated with a reflective metallic layer and a protective lacquer label. In order to read information from a CD, a laser beam is focused on the spiral track of pits and the amount of light reflected back into the objective lens is measured. Light striking one of the pits is diffracted (scattered) through such a wide angle that very little finds its way back into the lens. When the light is focused on the flat land between pits, most of it is reflected back into the lens. Figure 2-1 illustrates the difference in reflectivity between the pits and lands. It is the modulated signal produced by the combination of reflected and diffracted light that represents the information stored on the disc.

The logical standard specifies how the information on the disc is organized. In November 1985 members of the CD-ROM community met and developed a standard format for placing files and directories on CD-ROMs. This standard was named the High Sierra format and was submitted to the International

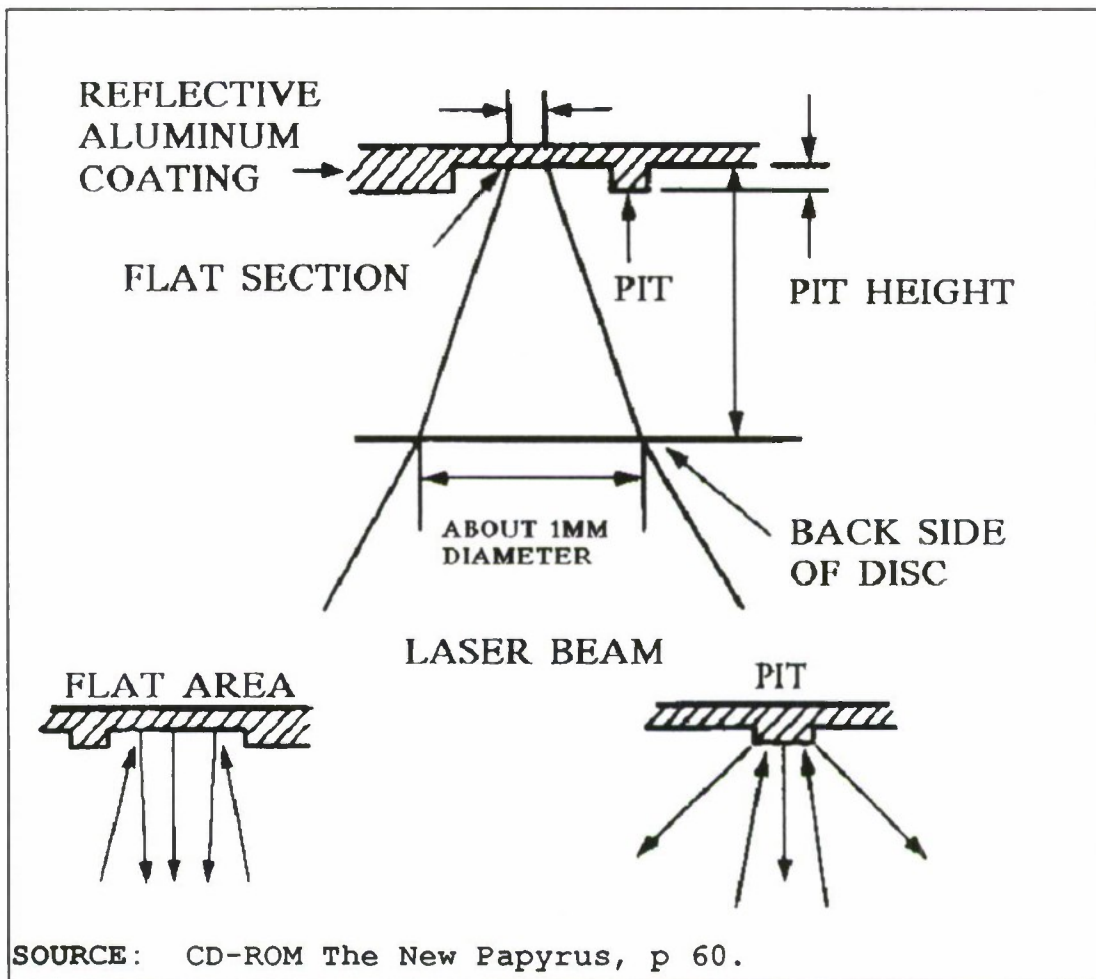


Figure 2-1. Reflectivity of Pits and Lands

Standards Organization (ISO). ISO accepted the High Sierra format and adopted it as ISO 9660 [Ref. 2]. Together these two standards ensure that all CD-ROMs may be read on virtually all combinations of CD-ROM playback units and computers.

Data are arranged on CD-ROMs in one continuous spiral. This is in contrast to magnetic disks which record information in concentric tracks, each of which holds a specified number of sectors. These differences require correspondingly different methods of operation when storing and retrieving the

data. The magnetic disk operates under a constant angular velocity (CAV) system where the disk spins at a constant rate but information is stored at a much lower linear density on the outside tracks compared to the inside. The reason for varying the density of the data is to compensate for the fact that the outer tracks of the disk pass by the reading mechanism much faster than the inner tracks. A CD-ROM disc operates under a constant linear velocity (CLV) system which varies the speed at which the disc spins in order to allow a constant linear density of the data throughout the disc. This approach allows for a much greater volume of information to be stored on the CD-ROM because the outside sectors can be as dense as the inner sectors.

III. ADVANTAGES AND DISADVANTAGES OF CD-ROM

A. ADVANTAGES OF CD-ROM AS A STORAGE MEDIUM

1. Storage Capacity

A single disc can hold up to 680 megabytes of data. This is equivalent to about 275,000 pages of text or 1,800 (360K) floppy disks. Table 3-1 summarizes the capacity of CD-ROM relative to other storage media. [Ref. 3]

Table 3-1. SUMMARY OF CD-ROM CAPACITY EQUIVALENTS

A SINGLE CD-ROM STORES AS MUCH INFORMATION AS:
- 275,000 PAGES OF TEXT OR,
- 1,800 5 1/4" FLOPPY DISKS OR,
- 1,200 MICROFICHE CARDS OR,
- 1,104 HOURS (46 DAYS) OF DATA TRANSMISSION AT 1200 BAUD OR,
- 17 40-MEG HARD DRIVES OR,
- 10 STANDARD 9-TRACK TAPES
Source: NPS Thesis by David Lind "Optical Laser Technology", p. 24, 1987.

2. Low Cost to Mass Produce

After a "master" has been produced, additional copies can be made for as little as \$2.00 each.

3. Industry Standard Access to Data

Because the information is stored and retrieved in accordance with standard formats, CD-ROMs produced by one manufacturer can be read by other manufacturer's drives. This is critical since the discs are usually mass produced and distributed to a large number of users with many different system configurations.

4. Relatively Low Cost Equipment Needed to Read Disks

CD-ROM drives are available commercially for as little as \$400 and can be connected to standard desktop computers. Nine track tape drives can also be connected to desktop computers but typically cost more than \$1,000. Other types of media may be inexpensive to install at the user end but cost much more when used to distribute data. Removable hard drives, for example, work well when transferring data from one machine to another but do not offer an economical means to distribute data to many users. Their capacity is also much less, typically only 40 megabytes.

5. Data are Non-volatile

Data are pressed into the disk during the manufacturing process and can not be rewritten by the user. Consequently, data are not subject to loss or corruption due to user error or computer viruses after the discs are pressed. Data are also unaffected by environmental hazards such as

magnetic fields, x-rays, extreme humidity, and moderate heat. This is in contrast to data stored on other media which could be lost due to power surges or exposure to magnetic flux generated by electric motors, monitors, etc.

6. Durable Medium

The polycarbonate material CD-ROM discs are made out of is durable and can withstand much more abuse than other media. Only severe damage will make a disc unreadable. Finger prints, dust particles and even minor scratches have no effect on a disc's readability because data are stored well beneath the surface. Consequently, users do not have to worry about mishandling the discs under normal working conditions. CD-ROM discs are also less affected by age than magnetic tape media. Data integrity of magnetic tape lasts approximately 2 to 7 years. Testing indicates data integrity of CD-ROM lasts from 10 to 50 years.

7. Removable and Portable

The discs are easily loaded and removed from their drives. Usually there is some type of ejection system which automatically pushes the disc out of the drive. This ease of loading/unloading makes changing discs and moving them from one machine to another a simple process.

8. Low Transportation Costs

Because the discs are small and light in weight, they are an inexpensive medium to mail and ship. Additionally, because they are not susceptible to adverse environmental conditions or electromagnetic radiation, their packaging is simpler.

9. Simplifies Security of Classified Material

The small size and portability of the discs make it easier to secure information they contain by securing the disc itself.

10. User Familiarity

Users are not intimidated by the technology because they have been exposed to the same type of discs used in the audio-CD music industry. Consequently, users are more comfortable with receiving and using CD-ROM discs than with other media such as 9-track tape.

B. DISADVANTAGES OF CD-ROM AS A STORAGE MEDIUM

1. Slow Access Speeds

Access time to retrieve data from a CD-ROM is much slower than a hard disk. Figures 3-1 and 3-2 graph the access and data transfer rates of CD-ROM, hard drives, and floppy drives. Modern indexing routines have been developed to enhance the "look up" function on CD-ROMs. Most CD-ROM applications respond to user requests within 2 seconds.

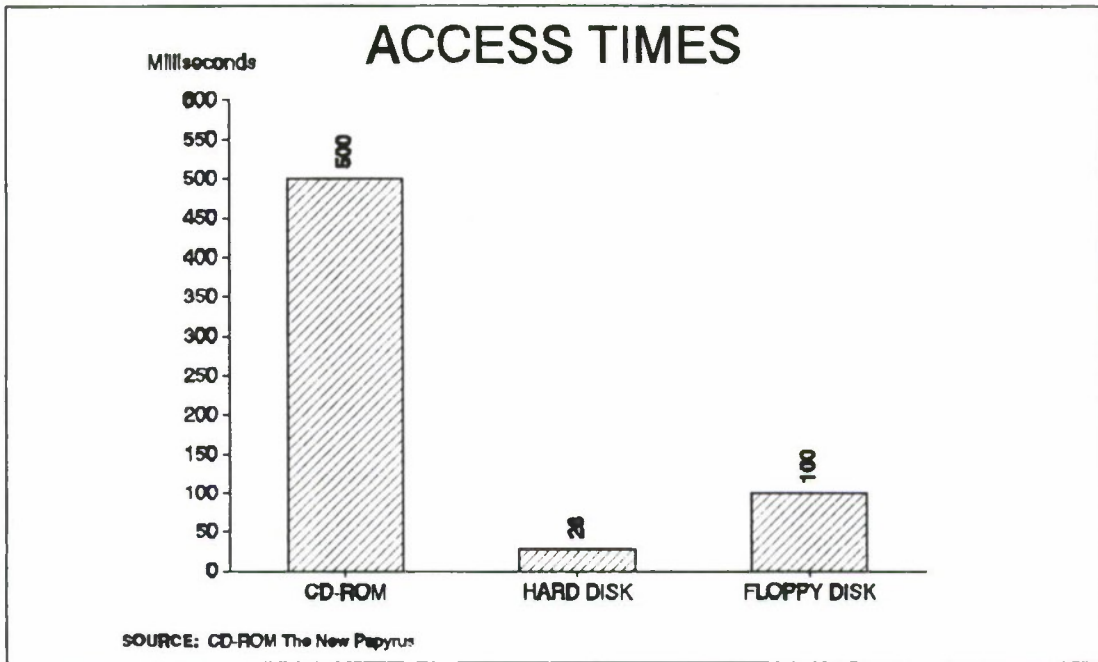


Figure 3-1. Relative Access Times

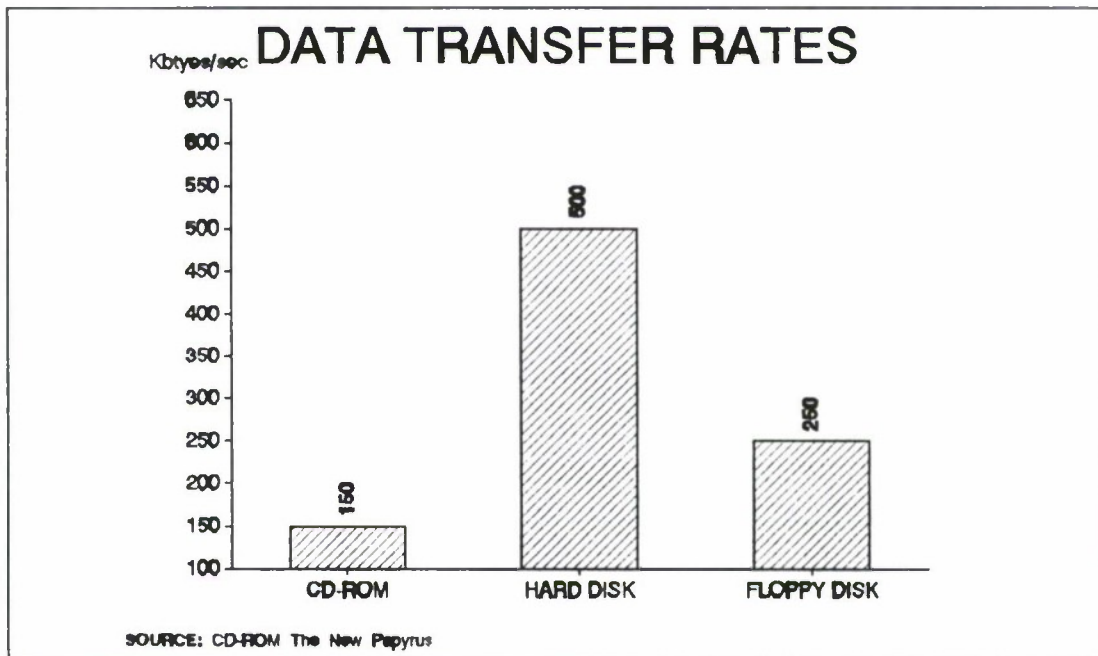


Figure 3-2. Relative Data Transfer Rates

2. Limited Number of CD-ROM Readers in Place Today

The technology is relatively new but the number of applications distributed on CD-ROM is growing. CD-ROM readers are being "pushed" to most Navy commands to support some CD-ROM applications which have already been developed. This means most users will have access to a CD-ROM reader but not necessarily within their own department. The number of CD-ROM readers distributed throughout the Navy will grow now that they are a line item on the DOD Desktop III contract and have a National Stock Number (NSN) assigned to them.

3. Read Only Media

Read only was a benefit to prevent unauthorized alterations to the data, however, it is also a disadvantage in that it prevents the making of legitimate changes to the data. Users must wait for an update to be produced and distributed or use a second medium as an addendum to the CD such as floppy disks.

4. Expensive Process When Dealing With Small Numbers

Though copies of the disc can be made for as little as \$2.00, you must produce a large enough quantity to recover the initial pre-master and master development expenses. This expense is approximately \$2,000 or more depending on the application.

IV. COST COMPARISON OF CD-ROM VERSUS PAPER

A. SELECTION OF A SUITABLE SOURCE FILE

One of the primary research questions for this thesis was to compare the cost of distributing information on CD-ROM with distributing the same information on paper products. To perform this comparison it was necessary to locate a source document which was being distributed on paper but also maintained on a magnetic medium. The magnetic medium would be used to input the data for the CD-ROM pre-mastering process. The following criteria were used to select a source document:

Document with wide distribution requiring a large number of duplicate copies.

Document of sufficient size to represent an average publication or manual.

Unclassified document whose source file was readily available on a magnetic medium.

Document currently being distributed in paper form and whose cost of printing and shipping could be easily identified.

The document chosen for this project was The Register of Commissioned and Warrant Officers of the United States Navy on Active Duty. This publication is about 650 pages long and is commonly referred to as the Officer Lineal List.

B. HOW THE INFORMATION IS DISTRIBUTED TODAY

The source file for the Officer Lineal List is maintained by Naval Military Personnel Command (NMPC) in Washington, DC. The information is maintained and updated by the Total Force Information System Management Department (NMPC Code 16) on their mainframe computers. When a new edition is to be printed, NMPC Code 16 provides a "camera ready" hard copy print-out to the Printing Department (NMPC Code 12). The Printing Department awards the print job to a contractor and the average number of copies made is 7,200. Most of the copies are turned over to Naval Publications and Forms Center (NPFC) in Philadelphia for distribution throughout the DOD. [Ref. 4] NPFC ships the publication via US mail, book rate.

C. CURRENT PRINTING AND SHIPPING COSTS

The printing cost to produce 7,200 copies of the Officer Lineal List is approximately \$30,000. NPFC reported the book rate shipping charges were \$1.95 per book for a total of approximately \$14,000. [Ref. 5] Thus the total cost of printing and shipping the publication is approximately \$44,000.

D. COST TO PRODUCE AND SHIP ON CD-ROM

Producing a CD-ROM is a three step process; preparing a pre-master, cutting the master disc, and producing copies of the master. For this thesis a pre-master was produced at

Naval Postgraduate School (NPS) using a piece of equipment called the CD-ROM Publisher from Meridian Data Corporation. Most activities will not have access to the specialized equipment required to prepare a pre-master and will have to have the job performed by an outside activity. Cutting the master and producing copies is usually done at a manufacturing activity such as 3M or Sony. The fees charged to produce a pre-master are approximately \$500.00. These fees will vary depending on what format the source data are in and the type of medium on which it is delivered. Charges to produce a master disc are about \$2,000. This cost will also vary depending on how short of a turnaround time is required. Tables 4-1 and 4-2 list the fees charged by the 3M company, as of March 1990, for pre-mastering and producing master discs.

Table 4-1. COSTS OF PRODUCING A CD-ROM PRE-MASTER

3M CD-ROM PRE-MASTERING FEES EFFECTIVE MARCH 1990	
SOURCE DATA MEDIA AND FORMAT	COST
REWITABLE OPTICAL MEDIA, 3M STANDARD FORMAT	\$300.00
DC6150 CARTRIDGE INPUT, 3M STANDARD FORMAT	\$500.00
1/2" 9-TRACK TAPE INPUT, 3M STANDARD FORMAT	\$700.00
* SURCHARGE OF \$300.00 IF DATA SET EXCEEDS 1,000 FILES.	
* SURCHARGE OF \$600.00 IF DATA SET EXCEEDS 10,000 FILES.	
Source: 3M Optical Recording CD-ROM Price Sheet, March 1990	

Table 4-2. COSTS OF PRODUCING A CD-ROM MASTER DISC

3M CD-ROM MASTERING FEES EFFECTIVE MARCH 1990						
TURNAROUND WORK DAYS	SAME DAY	1 DAY	2 DAY	3 DAY	5 DAY	10 DAY
PRICE (\$)	4,900	2,900	2,400	2,000	1,600	1,400
* PRICE BASED ON 9-TRACK TAPE INPUT IN 3M STANDARD FORMAT, OR DATA PRE-MASTERED BY 3M.						
Source: 3M Optical Recording CD-ROM Price Sheet, March 1990						

The distribution copies of the master disc cost about \$2.00 each, depending on the quantity required. The total cost of producing 7,200 CD-ROMs, including the pre-master and master process, would be \$16,900. This compares with the \$30,000 cost of printing the 7,200 manuals. The postage fee to mail a CD-ROM first class is 45 cents. Multiplied by 7,200 gives a total shipping fee for of \$3,240. Thus the total cost to produce and ship the 7,200 CDs would be \$20,140. As stated earlier the current cost of printing and shipping the manuals is approximately \$44,000. Figure 4-1 graphs the relative expenses involved for both processes.

As the numbers above indicate, there can be significant savings realized when distributing information on CD-ROM versus paper. These savings are multiplied many times over when the capacity of the disc is fully used. This project used 50 megabytes of storage space on the disc which has a capacity of 680 megabytes. This represents less than an 8

\$ CD ROM vs PRINTED MANUAL

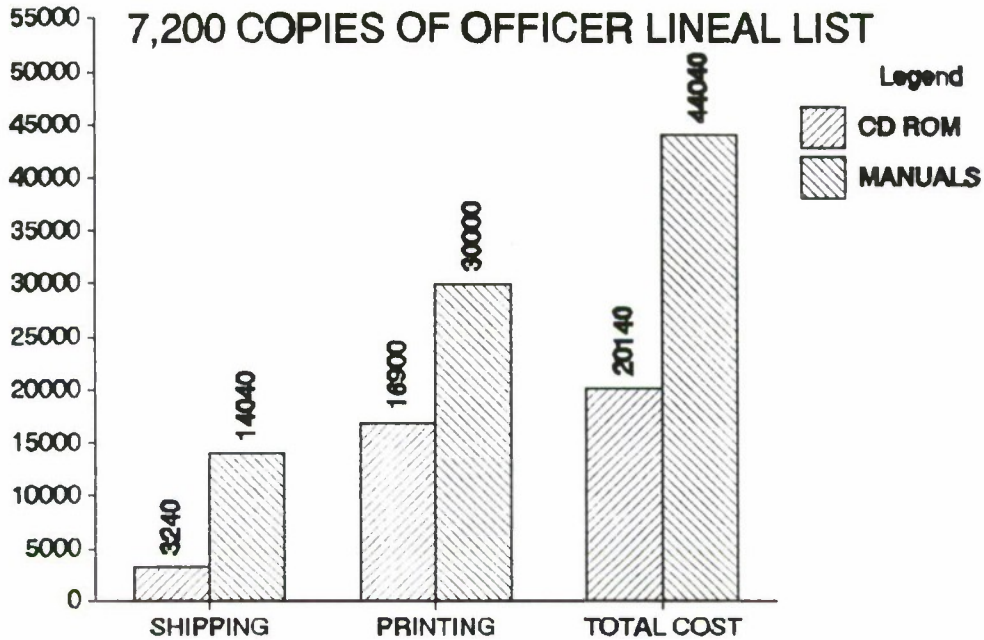


Figure 4-1. Cost of CD ROM vs Manuals for Officer Lineal List

percent usage factor. Up to the capacity of the disc, there is no additional cost to include more information (ie. other manuals) on the same disc. Conventional paper printing and shipping costs would go up proportionally with increased volumes of material. Figure 4-2 shows the total savings which could be realized if the capacity of the CD-ROM were fully utilized. The graph assumes the same requirement of 7,200

copies but now with thirteen times the amount of information to publish. The cost to print and ship the hard copy manuals rises to \$572,520. This is 28 times the expense to produce and mail the same information on CD-ROM.

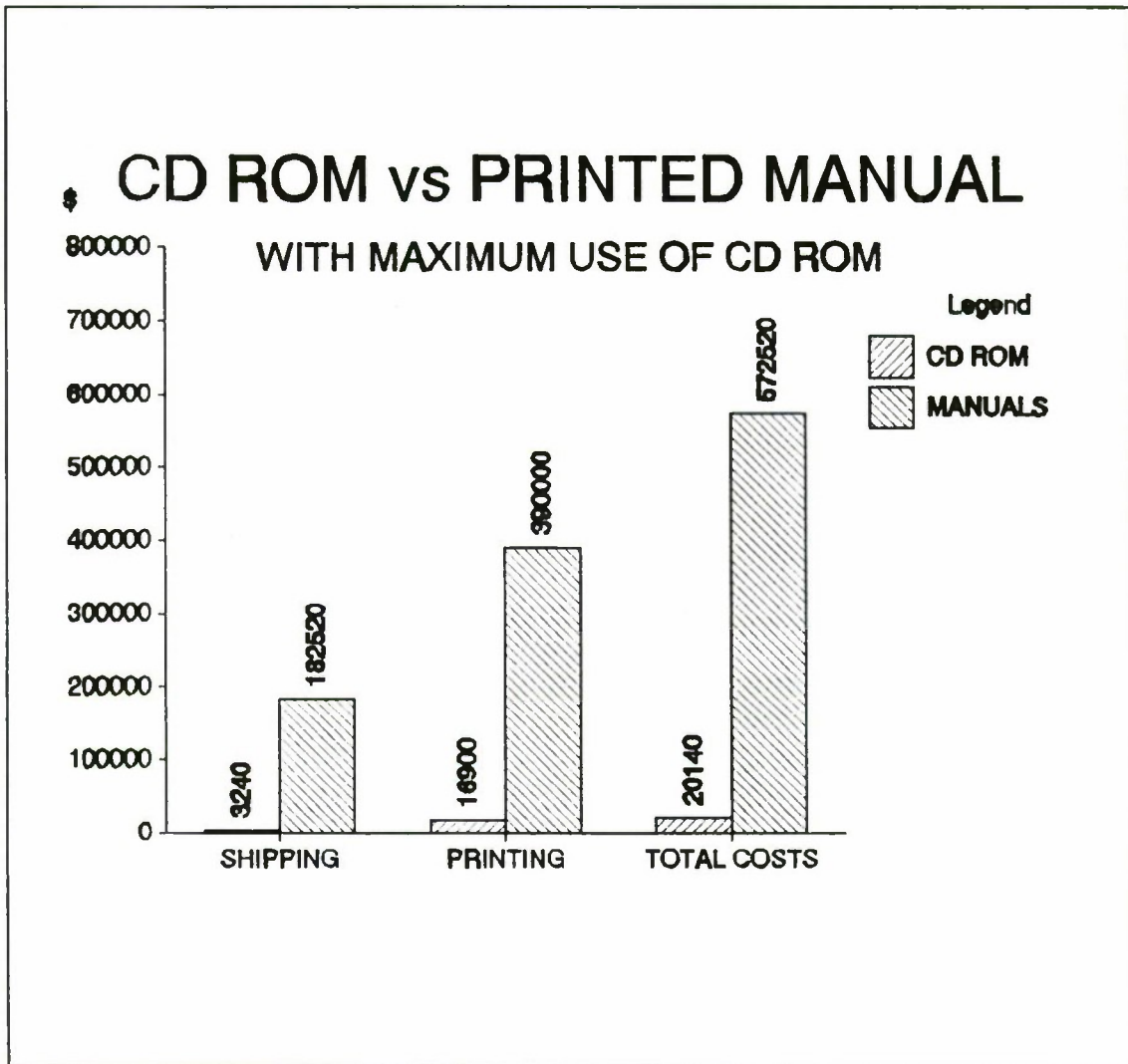


Figure 4-2. Cost Comparison with Maximum Use of CD-ROM

The figures above assume two thirds the capacity of the CD-ROM disc will be dedicated to data retrieval software and the indexes required to ensure rapid response when accessing

the information. If this is not a requirement, all of the disc could be used for storing data and the cost savings of using CD-ROM versus paper would be multiplied by a factor of three.

The cost of equipment used to read a CD-ROM has not been included in this analysis because micro-computers with CD-ROM readers are already being distributed to Navy commands in support of other CD-ROM applications. It is uncertain how many additional CD-ROM readers each command will require as more databases are distributed on CD-ROM. Because the discs can be changed quickly, a single CD-ROM reader can be used for several different applications. The number of readers required will be a function of the total demand from all applications.

V. DESIGN AND IMPLEMENTATION OF THE DATABASE APPLICATION

A. DEFINITION PHASE

1. Background of Problem Area

Many of the databases used by the Navy on a day to day basis, such as the Autovon phonebook, GSA catalogs, and Defense Property Disposal catalogs, are distributed on paper products in the form of manuals, catalogs and printouts. Because of this, extracting desired information requires a tedious manual look-up procedure. Distributing the information via paper also prevents the user from conducting ad hoc inquiries of specific information or easily producing individualized reports.

2. Definition of the Problem

Two readily available but difficult to use databases are: Register of Commissioned and Warrant Officers of the U.S. Navy on Active Duty (NAVPERS 15018, NSN 0500-LP-264-0119), and United States Navy Supply Corps Officers Directory (NAVSUP PUB 365).

Both of these databases are distributed annually as paper publications. These publications provide information such as name, lineal number, location, type of assignment and rank pertaining to U.S. Naval officers. This project addresses the problem of accessing, manipulating and

extracting information from these publications. The premise is that the same data, provided in an automated database format, will facilitate much more effective and efficient information access and distribution.

3. Project Scope

The project scope is limited to developing an operational database system capable of accessing the information contained in the two annual publications listed above. A "user-friendly" menu system allows authorized users to conduct specific searches and obtain associated reports. The system will support user specified queries and ad hoc report generation. Additions, deletions and modifications to the database would not be possible if this database were distributed on CD-ROM. The intended users of the system include any command with access to an IBM-compatible microcomputer and a CD-ROM reader who are on the distribution list for the current paper publications. The design and implementation of the database system is based upon the work of Dolan and Kroenke [Ref. 6].

B. REQUIREMENTS PHASE

1. Data Requirements

a. User Environment

The Command and Officer Data Information System (CODIS) contains unclassified Navy officer manpower data provided by the Naval Military Personnel Command (NMPC) via

the Defense Manpower Data Center (DMDC) located in Monterey, California. The intent of the system is to provide quick, easy access to data which currently are not available or are available only in paper form. CODIS eliminates the tedious manual effort required to extract data and reports from the paper publications. This system is intended for use by all Naval officers. It is written in dBase IV for IBM and compatible microcomputers.

The dBase programming language was chosen for these reasons:

It is widely used for programming micro-computer database applications.

It is relatively easy to use and program.

Applications can easily be expanded or enhanced.

Sophisticated users are not limited by the application program. Using dBase commands and functions, users can access and manipulate data in ways the developer did not envision.

The Developer's Edition provides the capability to produce runtime versions of application programs. Runtime versions do not require the dBase software in order to run.

b. Object Descriptions

The objects in the database are: the COMMAND object and the OFFICER object. The COMMAND object and the OFFICER object are compound objects as they each contain at least one object property. The object diagrams are shown in Figures 5-1 and 5-2.

The OFFICER object describes all active duty U.S. Naval officers. Name is the key of the OFFICER object. Social Security Number, although a unique identifier, is not included in this object due to Privacy Act constraints. The COMMAND object is a single valued object property of the OFFICER object. The OFFICER object also contains fourteen simple properties. Appendix C describes each of these properties in detail.

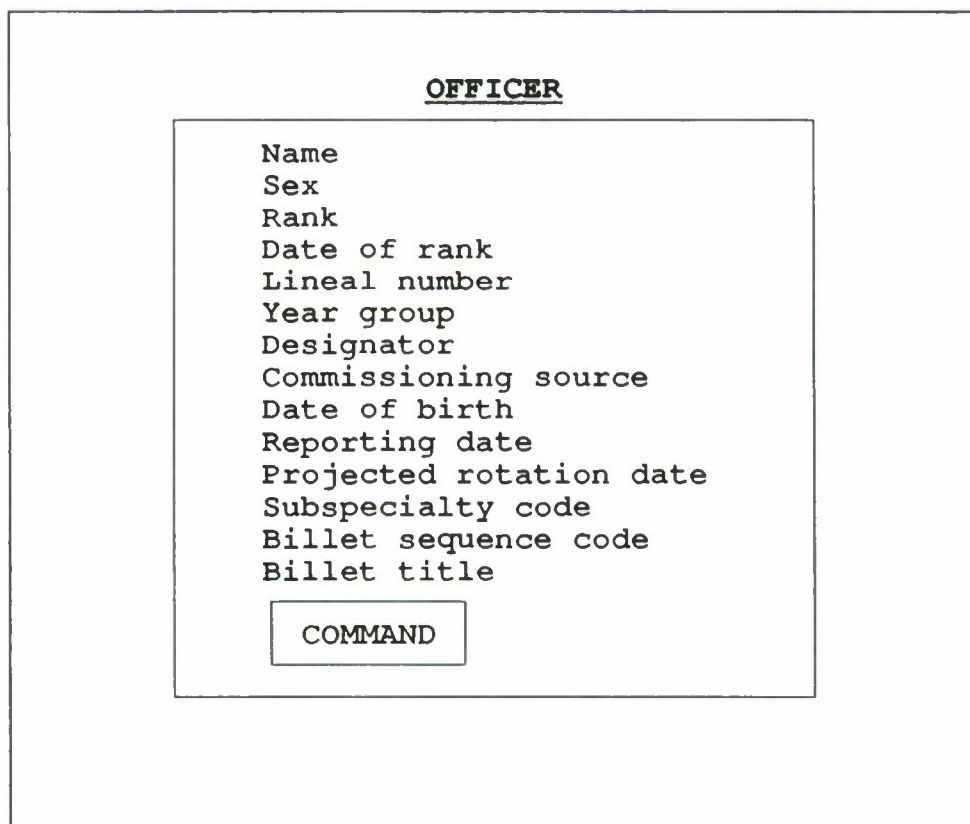


Figure 5-1. OFFICER Object Diagram

The COMMAND object describes all commands, activities and civilian companies which are assigned a Unit Identification Code (UIC). The property, UIC, is the key of the COMMAND object. The OFFICER object is a multi-valued

object property of the COMMAND object. The COMMAND object additionally contains five simple properties. Appendix C describes each of these properties in detail.

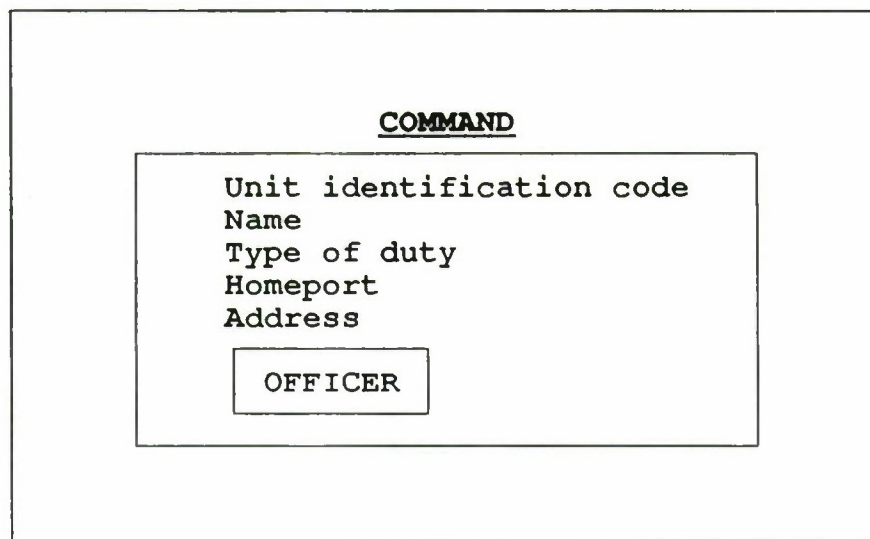


Figure 5-2. COMMAND Object Diagram

c. Methodology

These objects were derived through a combination of methods. Current published reports were reviewed to determine most of the objects and their properties. Users (i.e. NPS students and some NPS faculty members) were interviewed to determine their data and report needs.

The COMMAND object and the OFFICER object and most of their properties were derived from current published reports. The OFFICER object and the remaining properties were also requested by interviewed users.

2. Application Requirements

a. Data Flows/Processes

As the level one data flow diagram in Figure 5-3 shows, the Defense Manpower Data Center (DMDC) provides an extract of the Navy's Master Officer and Unit Identification Code files. This information is provided annually to DMDC by NMPC via magnetic tape. A copy of these files was provided by DMDC for this application. The information is then reformatted into two objects for use in the CODIS system. The objects created are COMMAND and OFFICER. These objects are stored in the CODIS database on CD-ROM for use by the application program modules. These modules are query and report generation.

The query module allows the user to extract information with searches based on the key fields of each of the objects. These key fields are Command Unit Identification Code (UIC) and Officer Name (Name).

The report generation module allows the user to select a specific report from a list of options. This module extracts information from the database as required and outputs reports to a screen display, printer, or disk file.

b. Object Update/Display Mechanisms

The CODIS database system consists of two major objects: OFFICER and COMMAND. Each of these objects contain subordinate single or multi-variable object properties, as

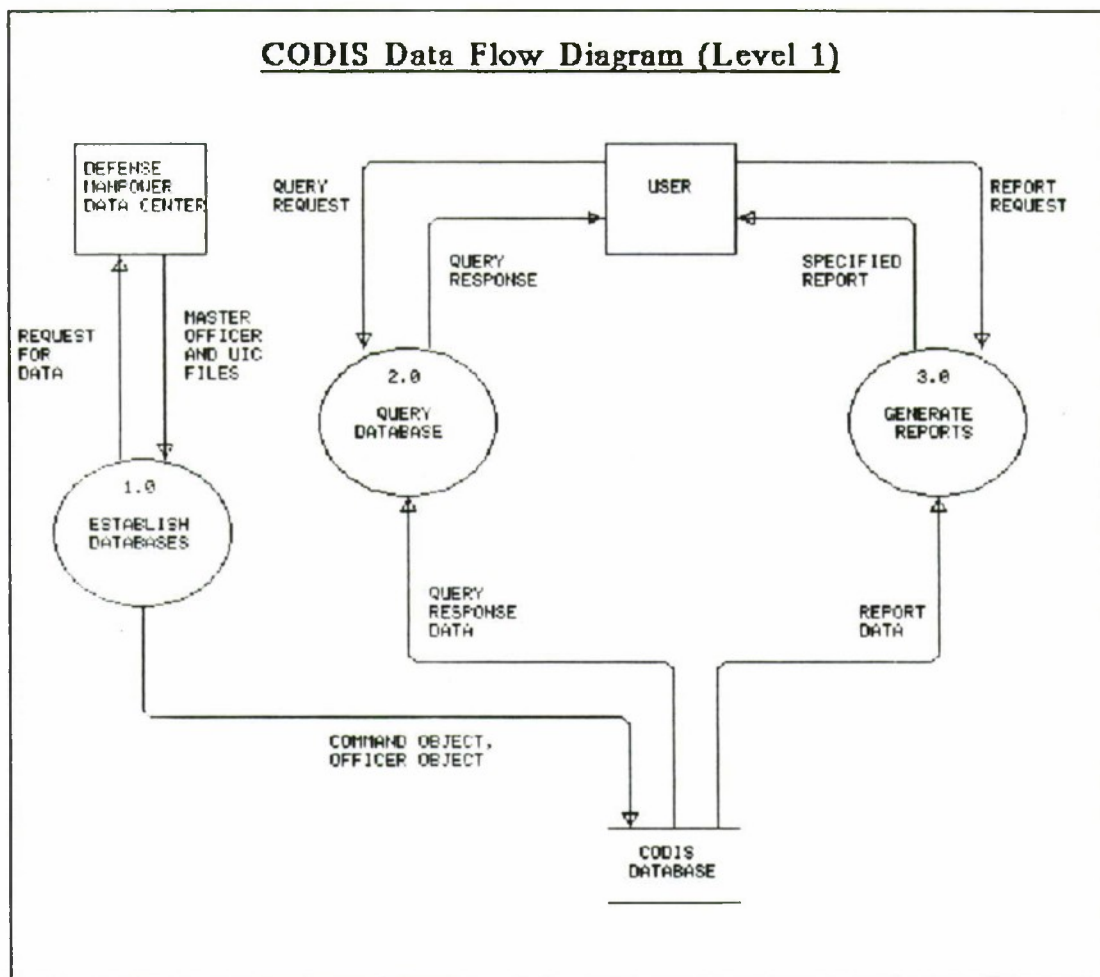


Figure 5-3. CODIS Data Flow Diagram

well as non-object properties. The update, display and control mechanisms for the objects are defined in Appendix D.

The object OFFICER consists of the non-object properties Name, Sex, Rank, Date of Rank, Lineal Number, Year Group, Designator, Commissioning Source, Date of Birth, Reporting Date, Projected Rotation Date, Subspecialty Code, Billet Sequence Code and Billet Title and the object property COMMAND (Appendix C). The OFFICER object is restricted to user queries, views and displays.

The object **COMMAND** consists of the non-object properties Unit Identification Code, Name, Type of Duty, Homeport and Address and the object property **OFFICER** (Appendix C). The **COMMAND** object is restricted to user queries, views and displays.

C. DESIGN PHASE

1. Logical Database Design

The **CODIS** database objects were transformed into a relational diagram as illustrated in Figure 5-4 below and Appendix E.

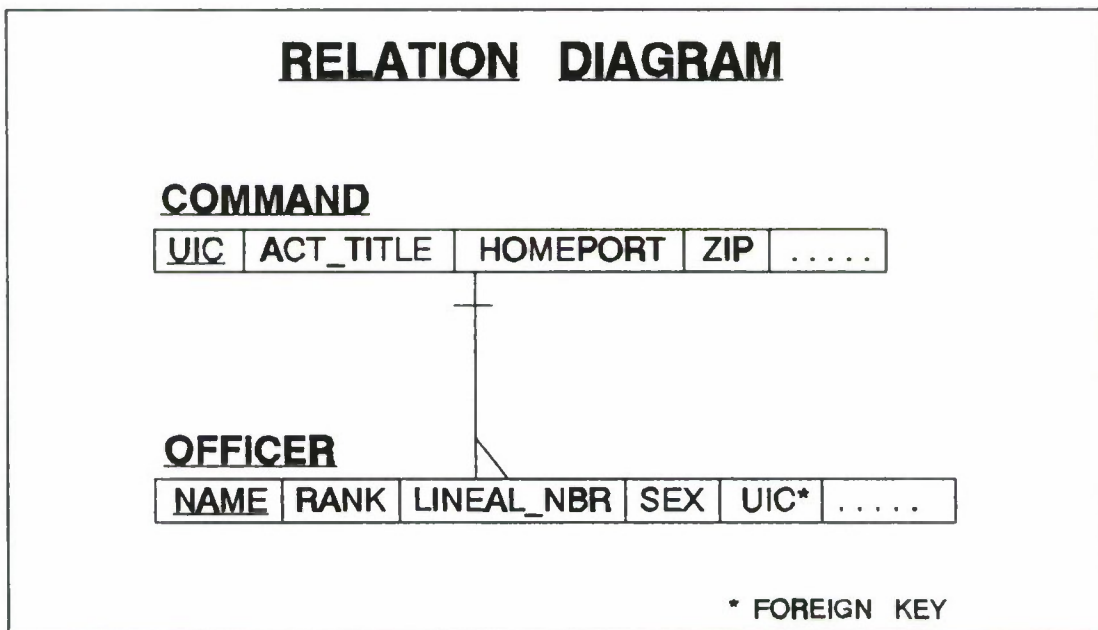


Figure 5-4. CODIS Relation Diagram

The **COMMAND** object is a compound object containing multi-valued occurrences of the object **OFFICER** in addition to several other descriptive properties (see Appendix E). The

COMMAND object contains the unique identifier UIC (Unit Identification Code) as its key.

The OFFICER object is a compound object containing the single-valued object COMMAND in addition to several other descriptive properties (see Appendix E). The OFFICER object contains the identifier Name as its key. The OFFICER relation contains the foreign key¹ UIC.

Commands may have many officers but each officer is assigned to only one command. The relationship between OFFICER and COMMAND is optional-many to mandatory-one.

2. Application Design

a. Control Mechanism

The CODIS database system will allow authorized users to direct and control application processing. This will be accomplished using a menu-driven control mechanism. The menu hierarchy will guide the user through the application by offering only appropriate options at each particular level.

The menu-driven control mechanism was selected because it offers several advantages over the alternative command-control mechanism. Although slower and considered cumbersome, the menu-driven mechanism is self-explanatory and easy to use. The CODIS menu system will not require the user to memorize any individual processing commands. Users will

¹A foreign key is an attribute that is also a key of a different relation.

follow the menu structure in order to invoke a desired application feature.

An additional advantage of the menu-driven control mechanism will be realized in access control. This system will deny access to all areas of the database outside of the menu structure. Users will be unable to access any data outside of the selected menu from the application.

Finally, a major criticism of the menu-driven control mechanism has been slow processing speed. In order to offset this deficiency, CODIS menus are capable of proceeding by either one of two methods. First, the cursor may be moved to highlight the desired function. Pressing the "ENTER" key will then access the next menu or desired function. A second and much faster means of procession is to press the capitalized first letter of the desired function. This will immediately access the next menu or the desired application function.

b. Menu Hierarchy

The CODIS menu hierarchy and sample menus are shown in Figures 5-5 thru 5-7. The CODIS application menu is described below to clarify the menu screens, options and program flow.

The action/object structuring strategy was used in the CODIS menu hierarchy. Users begin with a list of actions (see Main Menu, Figure 5-6). Subsequent lower level menus

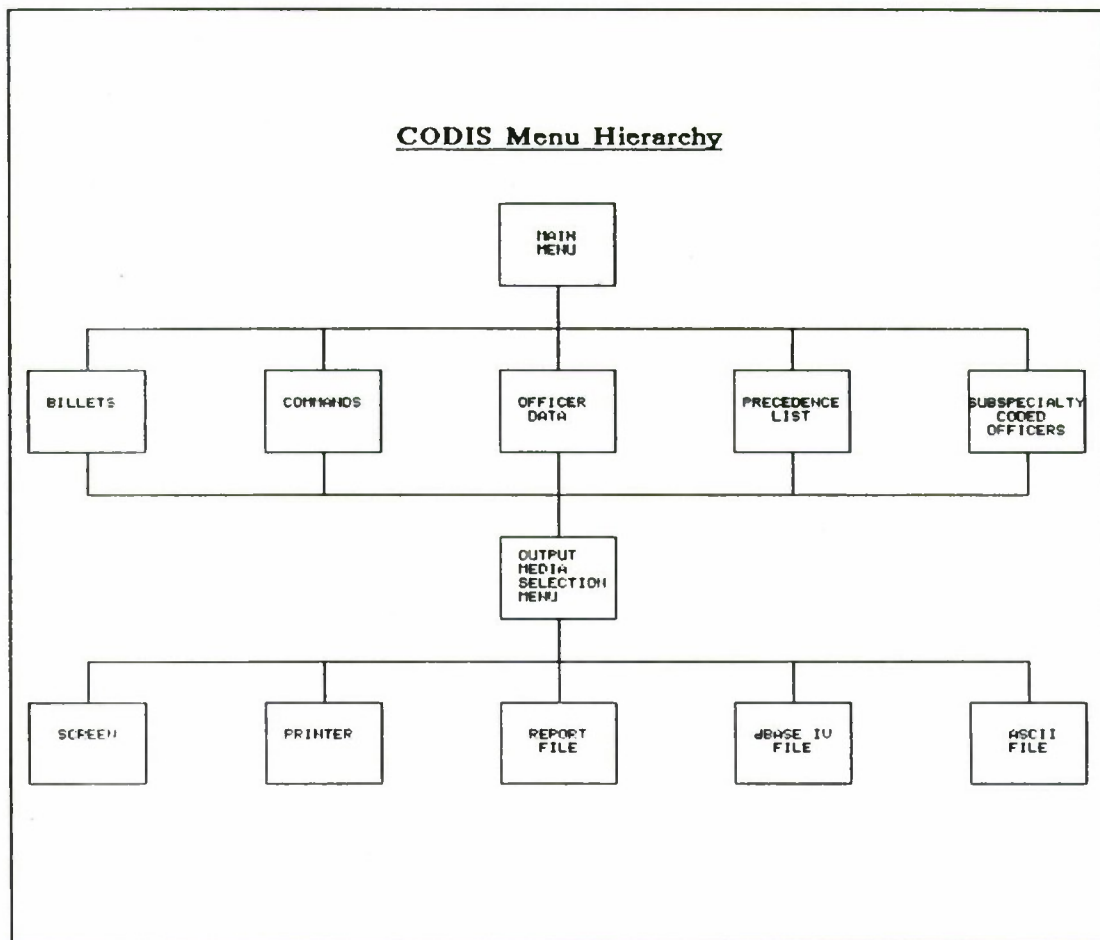


Figure 5-5. CODIS Menu Hierarchy

allow the user to select an object on which to perform some desired action or function. As the user progresses, the application is controlled by selecting the desired processing option from the displayed menu.

A detailed illustration of the CODIS Main Menu is shown in Figure 5-6. The Main Menu display offers the user several processing options. Option selection will determine subsequent menu displays and the database processing view. The Main Menu displays report options: Billets, Commands, Officer Data, Precedence List and Subspecialty Coded Officers.

Additionally, the Exit option may be selected from this menu. Selecting the Exit option will terminate the program. Selecting any of the other displayed options will access a query form that will request processing parameters. Upon completion of this form, the system will display the Output Menu.

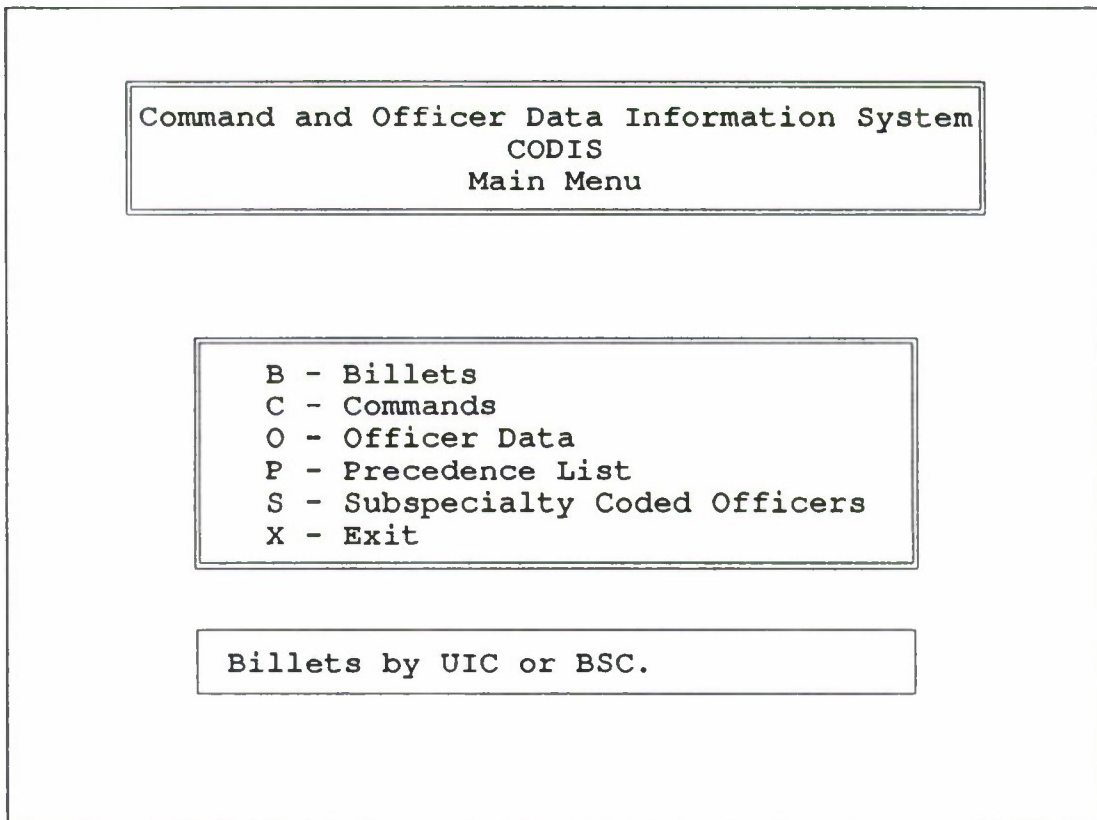


Figure 5-6. CODIS Main Menu

The Output Menu will allow the user to select the output medium. Specific options include Screen, Printer, Report File, dBase IV File and ASCII File. The user may also elect to return to the Main Menu by selecting the Return option or terminate the program by selecting the Exit option.

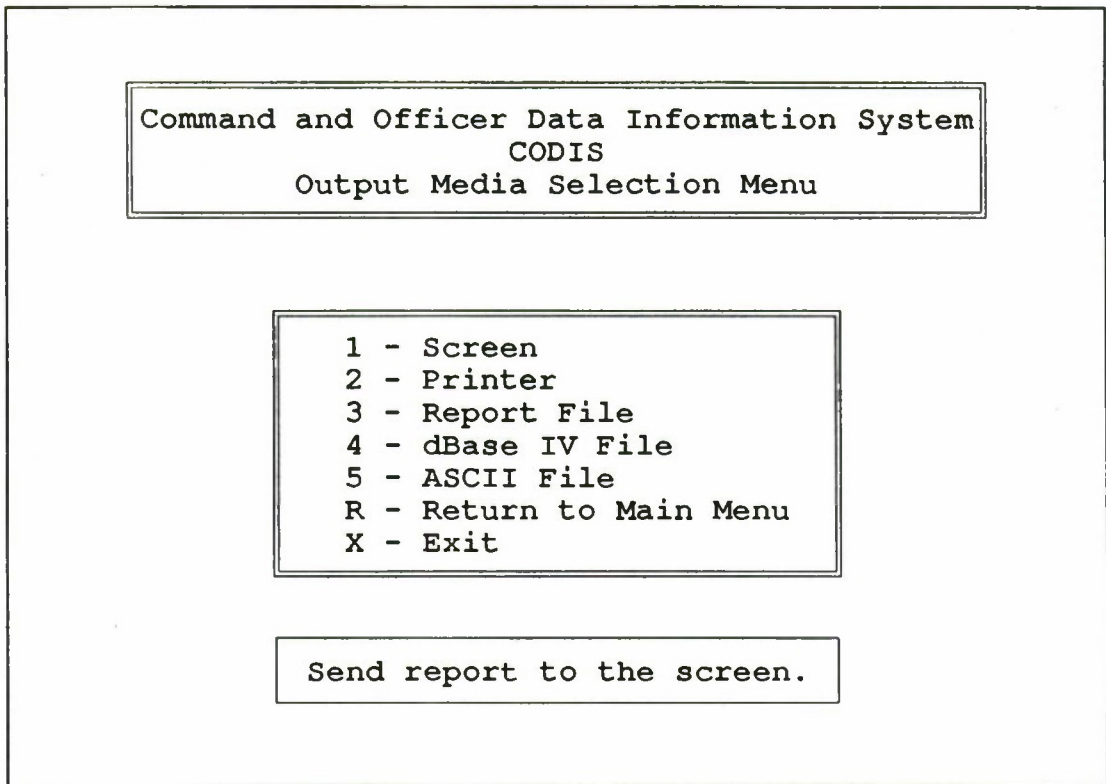


Figure 5-7. CODIS Output Media Selection Menu
c. Materialization

Materialization is the physical presentation of data in the form of a screen or report. Each of the reports generated by the CODIS system requires materialization. It is accomplished by selecting required properties from the application objects OFFICER and COMMAND.

The Billets By Navy Command report and the Billets By Billet Sequence Code (BSC) report are materialized by combining the properties Unit Identification Code, Name, Type of Duty and Homeport from the COMMAND object with the properties Billet Sequence Code and Billet Title from the OFFICER object. An example of each of these reports is shown in Figures 5-8 and 5-9.

<u>BILLETS BY BSC</u>					
<u>VIC</u>	<u>COMMAND</u>	<u>BSC</u>	<u>BILLET TITLE</u>	<u>HOMEPORT</u>	<u>SEA/ SHORE</u>
0586A	FT INTCPEN EURLNT	00015	XO	NORVA	1
57025	COMNAVAIRPAC	00015	EXECUTIVE ASSI	NORTI	1
65870	SUPSHIP LBCH CA	00015	SUPSHIP/CO SHR	LNGBCH	1
42192	NSSC OP SUPFDWAS	00015	OOC-DIR OCEAN	ARLING	1

Figure 5-8. Billets By BSC Report

<u>BILLETS BY NAVY COMMAND</u>					
<u>VIC</u>	<u>COMMAND</u>	<u>BSC</u>	<u>BILLET TITLE</u>	<u>HOMEPORT</u>	<u>SEA/ SHORE</u>
0586A	FT INTCENEURLNT	00025	CINCUSNAVEUR P	NORVA	1
0586A	FT INTCENEURLNT	04220	MILITARY CAPAB	NORVA	1
0586A	FT INTCENEURLNT	02300	NUC STRIKE DIV	NORVA	1
0586A	FT INTCENEURLNT	00015	XO	NORVA	1

Figure 5-9. Billets By Navy Command Report

The Navy Command By Geographic Location report is materialized by obtaining the properties Unit Identification Code, Name, Type of Duty, Homeport and Zip Code from the COMMAND object. Figure 5-10 is an example of this report.

The Navy Command By UIC report is materialized by obtaining the properties: Unit Identification Code, Name, Address, homeport and Type of Duty from the COMMAND object. An example of this report is shown in Figure 5-11.

NAVY COMMAND BY GEOGRAPHIC LOCATION

CITY: KEY WEST

<u>UIC</u>	<u>COMMAND</u>	<u>SEA/ SHORE</u>	<u>HOMEPORT</u>	<u>ZIP CODE</u>
00213	NAS KEY WEST	1	K WEST	33040-5000
44320	NAS KW AIMD	1	K WEST	33040-5000
47854	USCINCLANT JTF-4	1	K WEST	33040-5000
63425	NCU KEY WEST	1	K WEST	33040-5000

Figure 5-10. Navy Command By Geographic Location Report

NAVY COMMAND BY UIC

COMNAVDAC WASHINGTON DC UIC: 68519
NAVAL DATA AUTOMATION COMMAND HOMEPORT: WASHDC
WASHINGTON NAVY YARD BLDG 166 SEA/SHORE: 1
WASHINGTON DC 20374-1662

Figure 5-11. Navy Command By UIC Report

The Officer Data report is materialized by obtaining the properties Name, Rank, Date of Rank, Sex, Date of Birth, Year Group, Designator, Lineal Number, Commissioning Source, Reporting Date, Projected Rotation Date, Subspecialty Code, Billet Sequence Code and Billet Title from the OFFICER object. These properties are combined with the properties Unit Identification Code, Name, Type of Duty and Homeport from the

COMMAND object. An example of this report is shown in Figure 5-12.

<u>OFFICER DATA</u>		
NAME: <u>THOMPSON JOSEPH H</u>	RANK: <u>LT</u>	DATE OF RANK: <u>880525</u>
SEX: <u>M</u>	DATE OF BIRTH: <u>580917</u>	DESIGNATOR: <u>1160</u> YEAR GROUP: <u>82</u>
SUBSPECIALTIES: <u>0095/ /</u>	LINEAL NUMBER: <u>L10746220</u>	
UIC: <u>07976</u>	COMMAND: <u>MSO 446 FORTIFY</u>	SOURCE CODE: <u>06</u>
HOMEPORT: <u>L CRK</u>	DATE REPORTED: <u>890515</u>	FRD: <u>9106</u>
BSC: <u>60020</u>	BILLET TITLE: <u>DC ASST</u>	SEA/SCORE: <u>2</u>

Figure 5-12. Officer Data Report

The Precedence List report is materialized by obtaining the properties Lineal Number, Name, Rank, Date of Rank, Designator and Sex from the OFFICER object. An example of this report is shown in Figure 5-13.

<u>PRECEDENCE LIST REPORT</u>						
<u>LINEAL NUMBER</u>	<u>NAME</u>	<u>RANK</u>	<u>DESIG- NATOR</u>	<u>DATE OF RANK</u>	<u>SEX</u>	
L02001375	SMITH JOHN W	LCDR	1315	741001	M	
L02001834	JOHNSON THOMAS H	LCDR	1315	741101	M	
L02006805	BOYD JOSEPH N	LCDR	2200	750321	M	
L02006870	THOMPSON JAMES S	LCDR	2200	750321	M	

Figure 5-13. Precedence List Report

The Subspecialty Coded Officer report is also materialized by combining certain properties from each of the database objects. The Name, Rank, Subspecialty Code, Billet Sequence Code and Billet Title properties from the OFFICER object are combined with the properties Unit Identification Code and Name from the COMMAND object. An example of this report is shown in Figure 5-14.

<u>SUBSPECIALTY CODED OFFICERS</u>					
<u>SUBSPECIALTIES</u>	<u>NAME</u>	<u>RANK</u>	<u>BSC</u>	<u>BILLET TITLE</u>	<u>VIC</u>
0095G/ /	JOHNSON TIMOTHY J	LT	04790	FLIGHT CREW	09244
0095G/ /	COOK PAUL T	LT	99990	STUDENT	30465
0095G/0037S/	JAMES WILLIAM T	LT	03100	ADP OFFICER	00011
0095G/4053F/4044R	THOMAS JONATHAN A	CAPT	03640	STR REDI SW/OP	42553

Figure 5-14. Subspecialty Coded Officers Report

D. IMPLEMENTATION PHASE

1. Database Implementation

The implementation phase of this project includes the physical database design and the code generated to execute the CODIS application. A complete listing of dBase IV file structures is included in Appendix F. The program code is included in Appendix G.

The key field from the COMMAND file was selected as the primary index because it uniquely identifies each record. The COMMAND file was indexed using the UIC field.

The Name field was selected as the primary index for the OFFICER file. Although this may not be a unique identifier, reality dictates its use. In addition to Privacy Act considerations, experience suggests most users will not be able to provide any other unique identifying data (ie. SSN) to access records other than their own.

Normal database design practices restrict indexing to required fields only. However, because this database was designed for implementation on CD-ROM, indexes have been created for all fields in the database and some combinations of fields.

2. Data Preparation

The data provided by DMDC required additional manipulation and reformatting to move it from DMDC's files on the NPS mainframe computer to the target IBM-compatible micro-computer. IBM utility programs were used to copy the data from DMDC's Navy Standard Address file and the Navy Officer Master File to temporary datasets.

The CODIS application is intended for implementation on IBM-compatible micro-computers, therefore, it was necessary to partition the files into smaller segments for downloading to floppy disk. Fortran programs (written by Mr. Dennis Mar

of the NPS Computer Center) were used to access the datasets and partition them into one megabyte files. The partitioned datasets were then downloaded to 5 1/4 inch high-density floppy disks for transfer to the micro-computer.

Two dBase IV data files were created on the target micro-computer's hard disk, one for the Navy Standard Address file (hereafter referred to as the UIC file) and one for the Navy Officer Master file (hereafter referred to as the Officer file). Using dBase IV, data was appended to each of these files from the floppy disks.

A problem was encountered upon examination of the Officer file. Only part of the total file could be accessed. Through close examination of the file, numerous end of file characters were detected. Further research determined the EBCDIC to ASCII conversion process performed during the file download from the IBM mainframe to be the source of these characters. The program translated certain characters in a filler field as end of file characters. The solution to the problem was to delete all characters from the filler field in the temporary data set on the mainframe. The data set was repartitioned, downloaded to floppy disk and appended to a new dBase IV Officer file. No further problems were encountered with either file.

VI. PROS AND CONS OF THE CD-ROM APPLICATION AND THE PRINTED MANUAL

A. CD-ROM APPLICATION PROS

1. Data Protection

The CD-ROM medium protects the data from unauthorized changes, user errors, computer viruses and environmental hazards. Once the data is mastered to the disk, it cannot be changed.

2. Access Speed

A sample group of five individuals was used to compare the average access times for retrieving information from both the Lineal List manual and the CD-ROM pre-master application. Each person was given three tasks to accomplish: 1) find an officer's lineal number when given his/her name, 2) find an officer's name when given a lineal number and 3) count the number of officers with a specific designator within a given range of lineal numbers. Each task was timed separately. The CD Publisher SIM program was used to simulate the response times of an actual CD-ROM playback unit. The CD-ROM application outperformed manual look-ups in every instance. Appendix B provides the detailed test results. See Figure 6-1 for a graphic comparison of these results.

AVG RETRIEVAL TIMES

CD-ROM vs LINEAL LIST MANUAL

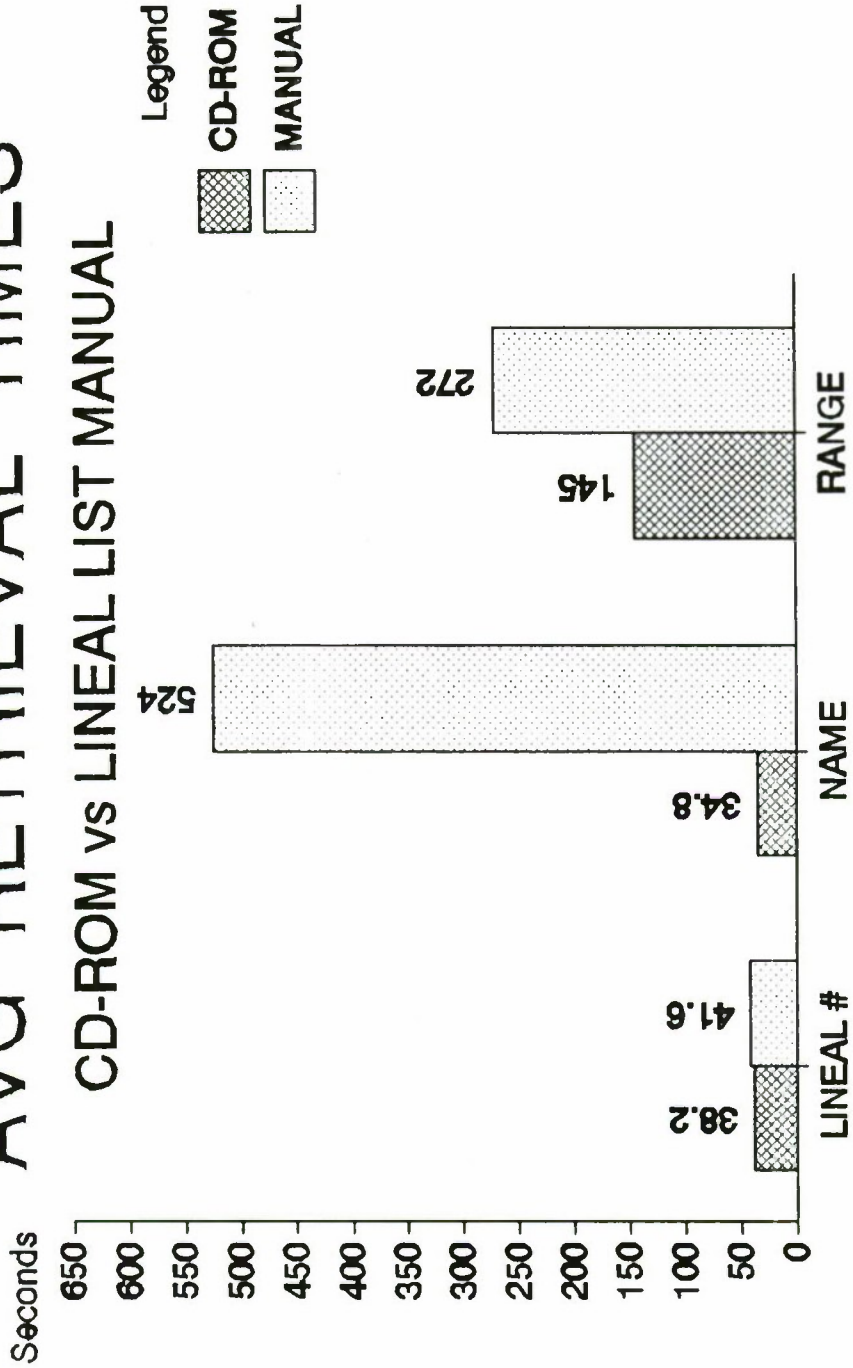


Figure 6-1. ACCESS TIME COMPARISONS

3. Accuracy

Data extraction using the CODIS application is consistent and errorless. Using a programmed set of procedures, data is accessed and retrieved the same way each time.

4. Availability of Reports

The CD-ROM application provides a variety of reports for accessing and extracting specific information. The Lineal List manual requires look up of individual entries, one at a time. The CD-ROM application offers individual and/or group queries. The following reports are available:

Lineal Numbers: individual or range

Billets by location or Billet Sequence Code

Commands by geographic location or Unit Identification Code (UIC)

Individual officer data

Officers with specific subspecialties.

5. Data Manipulation/Extraction

Because the data is stored in digital format, extraction and manipulation is quick and effortless. The CODIS application extracts information from the database and can produce various reports. Additional data manipulation may be performed by any user familiar with the dBase program.

6. Standard Off-the-Shelf Software

The dBase program has established itself as an unofficial standard for both government and business use. Having the CD-ROM database in this format ensures data compatibility. More sophisticated users can write their own applications to access/manipulate the data for additional ad-hoc queries and reports.

B. CD-ROM APPLICATION CONS

1. Information Retrieval Performance

Although the CD-ROM application outperformed manual searches, there was not a significant improvement, as expected, over a non-automated process.

2. Hardware Requirements

The application, as developed, requires a microcomputer with a hard disk drive and a CD-ROM playback unit. Obviously the Lineal List manual itself requires no additional hardware and is more portable than the equipment required to access the CD-ROM application.

3. Read Only

Because the data is pressed on the medium during the mastering process, it cannot be changed. This prevents making legitimate changes to the database. Users must wait for an update CD-ROM to be produced and distributed.

C. PROS OF THE PRINTED LINEAL LIST MANUAL

1. Data Modification

Interim changes can be made to the data at any time by writing the correct data on the page or replacing pages. It is not necessary to reprint the entire manual for these interim updates.

2. Portability

The Lineal List manual can be hand-carried from place to place. No additional equipment is required in order to access the data.

D. CONS OF THE PRINTED LINEAL LIST MANUAL

1. Durability

Paper manuals are subject to deterioration. Pages tear and print fades and smudges with use and age making readability difficult.

2. Accuracy

The process of locating information in the Lineal List manual is error prone. The print is small and tightly spaced making it difficult to read. Pen and ink changes made by many individuals may not be correct.

3. Data Format

The Lineal List manual provides data in only two sequences: lineal number within designator and name. If data is needed in any other sequence or format, it must be extracted and manipulated manually.

4. Information Retrieval Performance

Information retrieval performance from the Lineal List manual is inconsistent. The speed and accuracy is totally dependent on the individual performing the retrieval.

VII. SUMMARY

A. CONCLUSION

CD-ROM is a preferred medium for the distribution of large databases to many users. It has distinct advantages over paper publications.

Because of its capacity, CD-ROM technology greatly reduces storage and shipping costs. "An Army division's records that weigh more than 11 tons in paper form will fit onto only 35 pounds of optical disks." [Ref. 7]

The cost to print and ship hard copy manuals is 28 times the expense to produce and mail the same information on CD-ROM.

Distributing information on CD-ROM gives users an automated tool to manipulate and retrieve data. Just as word processors have changed the way we write, CD-ROM will revolutionize the way information is accessed and used.

A barrier to wide spread acceptance and everyday use of CD-ROM technology has been the sophisticated equipment required to access information on a disc. Though a CD-ROM disc itself is compact and easily transported, the requirement for a microcomputer with a CD-ROM reader negated this advantage. New products just released will overcome this barrier. Sony Corporation recently released a handheld CD-ROM

reader called "Data Discman" [Ref. 8]. Similar products will give users even more ability to carry and access vast amounts of information wherever they go.

After the hardware and software have been installed, producing a CD-ROM pre-master is a straight forward process provided the source data is stored in digital format. The process would be more complex if it were necessary to convert the source data from a paper medium.

B. RECOMMENDATIONS

Because CD-ROM readers are already available to most Navy commands, the Navy should promote the conversion of paper manuals to CD-ROM. Budget reductions and personnel cutbacks demand improved efficiency in all areas of operations. CD-ROM would provide a more economical and efficient method of publishing, distributing, and using data.

Follow-on research should be conducted to evaluate CD-ROM authoring and retrieval software to identify those that best meet the application needs of the Navy. Most commercial CD-ROM databases on the market today do not use products such as dBase as the retrieval engine. dBase and similar products are not optimized for use with CD-ROM. Because a compact disc has one spiral track that winds from the inside of the disc to the outer edge, the speed of the disc's rotation varies to ensure the data moves past the drive's laser beam detector at a constant velocity. This optimizes the density of data on the

disc but takes its toll in access time, which can be ten times as long as that of typical hard disks. CD-ROM specific authoring and retrieval software organizes the data in a way to minimize access time. It accomplishes this task by indexing key words, adding cross-references and dividing information into logical blocks to allow for efficient retrieval. Two commercial CD-ROM applications, Microsoft Bookshelf and Computer Library, were examined to compare the efficiency of dBase to other retrieval software developed specifically for use with CD-ROM. A direct comparison cannot be made between the dBase application and the commercial CD-ROM products because the composition of the databases is different. However, response rates to queries made from the commercial CD-ROM applications were at least twice as fast as responses given from the dBase application for similar queries. These observations indicate that using retrieval software products designed specifically for CD-ROM vice dBase could significantly enhance the application's performance.

The Naval Postgraduate School should acquire the necessary equipment to produce CD-ROM discs and develop additional applications which demonstrate the capabilities of this technology.

THE PRE-MASTERING PROCESS

A. INTRODUCTION

The purpose of this appendix is to give the reader a brief synopsis of the essential steps involved in producing a CD-ROM pre-master tape. The source for this information was the CD-Publisher User's Manual from Meridian Data, Inc. (MDI), technical support from MDI and other readings. The overview presented here is intended to provide a general understanding of the various steps required. Though these procedures specifically address the use of the CD-Publisher from MDI, systems from other manufacturers should include similar functions.

B. DESCRIPTION OF HARDWARE

The equipment used to create the pre-master tape for this project was a CD Publisher model 88-600 from Meridian Data Inc. (MDI) and a Compaq Deskpro 386 microcomputer. The CD-Publisher includes a large capacity tape-and-hard-disk subsystem which connects with IBM PC and Macintosh host computers via a Small Computer Standard Interface (SCSI) bus. To provide the necessary interface between PC hosts and the CD Publisher, MDI supplies a SCSI host adapter board that fits into an IBM-AT sixteen bit expansion slot. The SCSI interface for our system was a Western Digital model WD7000. Macintosh

Appendix A

computers do not require a separate interface because the CD Publisher connects to the machine's built-in SCSI interface. The tape subsystem consists of a 1/2 inch 9-track 100-IPS Industry Standard Interface tape drive with recording densities of 6250 BPI and 1600 BPI. The hard disk subsystem is expandable in increments of just over 300 MB up to 2.4 Gigabytes in one cabinet, and up to 7.2 Gigabytes in three cabinets. Our model 88-600 consisted of two hard drives with a total capacity of 635 Megabytes. Though the CD Publisher may contain several physical hard drives, they function and are partitioned just as a single drive would be. Table 6-1 lists the physical characteristics of the CD Publisher.

CD Publisher Physical Characteristics

Width: 22 5/32 inches
Height: 43 1/2 inches
Depth: 30 7/8 inches
Weight: 375 to 425 pounds (uncrated)
575 to 625 pounds (crated for air transport)

Source: Meridian Data, Inc. CD Publisher User Manual

Table 6-1. CD Publisher Physical Characteristics

C. PC HOST COMPATIBILITY

The original host microcomputer connected to the CD Publisher was a Zenith Z-248 IBM AT compatible. The installation phase was completed without a problem and the Z-

Appendix A

248 was able to transfer files to the hard drive subsystem of the CD Publisher. It was also able to address and read the files contained on the hard drive subsystem. Incompatibility problems surfaced when an attempt was made to execute a program from the CD Publisher hard drive. The Z-248 locked up and displayed the message "Memory Parity Error". After exhaustive troubleshooting and a step by step verification of the installation procedures, a call was made to the Meridian Data, Inc. technical support staff. A service engineer stated the "Memory Parity Error" was frequently encountered when the CD Publisher was connected to a microcomputer which used back-plane technology, i.e. the CPU of the system is located on an expansion card and plugged into the system bus just as other cards would be. The Z-248 uses this back-plane design and was therefore suspected of being incompatible with the CD Publisher. The service engineer stated the problem is a result of the PC host not being compatible with the WD7000 SCSI interface card connecting the CD Publisher to the microcomputer. Meridian Data, Inc. provided a list of other microcomputers which have been verified as compatible or not compatible with the CD Publisher interface system. Table 6-2 is a summary of that information. The incompatibility of the Z-248 was verified by reinstalling the CD Publisher on a Compaq Deskpro 386 and having the system operate correctly.

PC's COMPATIBLE WITH A WD7000 HOST ADAPTER

COMPAQ 386 33 MHZ	TEXAS MICRO 386 20 MHZ
COMPAQ 386 25 MHZ	HP VECTRA 386 20 MHZ
COMPAQ 386 20 MHZ	IBM AT/XT 286/386 16/20 MHZ
COMPAQ 286 16 MHZ	OLIVETTI AT CLONE
COMPAQ 280 12 MHZ	ACER 386 33 MHZ
WYSE MODEL WY2112-01	ARC 286
MYLEX 386 16 MHZ	PC LIMITED 386

**PC's NOT COMPATIBLE WITH A
WD7000 HOST ADAPTER**

DELL 386 W/ VIDEO 7	TELEVIDEO TELE386
WYSE MODEL WY2214-01	AST 386 33 MHZ
WYSE MODEL WY2200-01	HAUPAUGE 386
ALTECH 386SX	JOY 386 20 MHZ W/CPU CACHE
ALR FLEXCACHE 386 20/25 MHZ	
ZENITH Z-248 (Reported during this thesis)	

* Also other systems with certain versions of the Vega Video 7 board and the Orchid VGA board.

Source: Meridian Data, Inc.

Table 6-2. PC HOST COMPATIBILITY

D. SYSTEM INSTALLATION

The first step of the installation procedure is to become familiar with the various components of the CD Publisher, including the SCSI interface card which connects the system to the host microcomputer. A separate Installation Manual is supplied with the CD Publisher and includes a checklist of items which should have been delivered with the system. The Installation Manual gives a description and drawing of the major system components to assist with the hardware installation phase. Installing the SCSI host adapter was

Appendix A

straightforward and similar to installing any other sixteen bit expansion card. After the hardware installation was completed, the next step was to install the CD Publisher operating software. Most of the software was provided on a 9-track tape. Before the CD Publisher tape subsystem could be used, a limited number of programs and drivers were installed onto the microcomputer from the supplied installation floppy disk. An empty bootable floppy disk was used to make a copy of the programs and drivers found on the installation disk. The 9-track installation tape was made ready and loaded onto the tape drive subsystem. The tape drive was put on-line and then the microcomputer was booted from the installation floppy disk. The purpose of this was to load the SCSI driver so the micro could communicate with/operate the CD Publisher. The next step was to type "INSTALL C:." This prompted the INSTALL.BAT program on the floppy disk to create several directories on the hard drive and begin copying the CD Publisher operating system files from the tape to the microcomputer.

The final step to complete the software installation was to modify the AUTOEXEC.BAT and CONFIG.SYS. Table 6-3 shows both of these new files. The device commands in the CONFIG.SYS file allow the microcomputer host to address the SCSI adapter and MS-DOS/High Sierra partitions on the CD Publisher, respectively. The MSCDEX command in the AUTOEXEC.BAT file is

CONFIG.SYS

```
LASTDRIVE=Z
FILES=20
BUFFERS=20
DEVICE=C:\MDIREL4.4\CDPUB\DRIVERS\SCSIWD70.SYS
DEVICE=C:\MDIREL4.4\CDPUB\DRIVERS\MCDPUB.SYS
DEVICE=C:\MDIREL4.4\CDPUB\DRIVERS\HCDPUB.SYS
```

AUTOEXEC.BAT

```
PATH=C:\MDIREL4.4\CDPUB;
MSCDEX /D:MDIHSDVR
```

Source: Meridian Data, Inc. CD Publisher User Manual

Table 6-3. NEW CONFIG.SYS AND AUTOEXEC.BAT FILES

the MS-DOS extension utility which allows the PC host to address High Sierra partitions on the CD Publisher as another drive device.

E. HIGH SIERRA/ISO 9660 PRE-MASTER TAPE DEVELOPMENT CYCLE

Depending on the amount of disk storage that is available on the CD Publisher, either of two different procedures can be used when making a High Sierra/ISO 9660 disc image (pre-master tape). The first procedure is used when the size of CD Publisher storage is at least twice the size of the disc image to be generated. The second procedure is used when the size of CD Publisher is less than twice the size of the final disc image. For example, if the user has a 600 MB CD Publisher, procedure 1 would be chosen to make a 200 MB disc

Appendix A

image, and procedure 2 would be chosen to make a 550 MB disc image. Procedure 1 is easier, quicker and more flexible to use, but a smaller sized CD Publisher can be used for procedure 2. Tables 6-4 and 6-5 outline the steps involved for both procedures.

Procedure 1: CD Publisher Twice Capacity of Disc Image

1. Partition CD Publisher hard disk, using PARTDISK, to make the appropriately sized MS DOS and HS or ISO partitions.
2. Copy data from input medium to the MS DOS partition. Modify data as required.
3. Zero partition using CDRFILL (optional).
4.
 - a. Initialize a HS partition with INITHS.
 - or
 - b. Initialize an ISO partition with INITISO.
5. Create any need HS/ISO subdirectories with MDHS/MDISO (optional).
6. Copy files from MS DOS partition to HS or ISO partition with COPYHS or COPYISO.
7. Invoke Microsoft extensions.
8. Perform simulations with SIM.
9. Make final modifications/corrections, then retest.
10. Run FINALHS or FINALISO to convert HS/ISO partitions into final format.
11. Verify final disc image using SIM.
12. Create pre-master tape with MASTERHS.

Source: Meridian Data, Inc. CD Publisher User Manual

Table 6-4. PROCEDURE 1 FOR DEVELOPING PRE-MASTER TAPE

Appendix A

It should be noted that the capacity of all CD Publishers can be upgraded in the field to at least 1.2 Gigabytes. Because procedure 1 requires double the hard disk capacity of the CD-ROM disc image, this is the maximum size needed to make a pre-master tape for a 600 MB HS/ISO CD-ROM.

The procedures outlined refer to several utility programs supplied with the CD Publisher system. PARTDISK partitions the disk subsystem into logical mass storage units in the MS-DOS, High Sierra, or ISO 9660 format. Files contained in partitions other than MS-DOS format cannot be modified with standard MS-DOS functions. If performed on a disk that has data on it, PARTDISK destroys all of the file organization data on the disk subsystem by reallocating the partitions, but does not zero the data in any partition. CDRFILL is used to write a sector address at the beginning of each sector of the partition and fill the balance of the sector with a data test pattern. This is a useful security measure for over-writing old data to prevent its recovery. The program is only used on High Sierra or ISO 9660 partitions. INITHS/INITISO writes (initializes or reinitializes) the root High Sierra directory structure and associated data structures into the specified High Sierra partition. INITHS/INITISO must be executed prior to executing any other HS/ISO function on that partition. MDHS/MDISO makes a subdirectory (directory) in a HS/ISO partition. COPYHS/COPYISO copies files from a MS-DOS

**Procedure 2: CD Publisher With Less Than Twice
the Size of Disc Image**

1. Partition CD Publisher hard disk, with PARTDISK.
2. Copy data from input medium to MS DOS partition.
3. Use TBACKUP to back up the data set to tape.
4. Repartition the CD Publisher with PARTDISK to make the appropriately sized HS or ISO partition.
5. Initialize the partition with INITHS or INITISO as appropriate.
6. Create any needed HS or ISO subdirectories with MDHS or MDISO.
7. Restore the files from the TBACKUP tape with TRESHS for HS or TRESISO for ISO partitions.
8. Invoke the Microsoft extensions.
9. Perform simulation with SIM.
10. Make final modifications/corrections, then retest. Because the MS DOS data no longer exists on disc, last minute changes are more difficult to make.
 - a. If the file to be changed is small, it may be changed on the host computer's resident hard disk.
 - b. If a large file must be modified, back up files to tape and repartition the CD Publisher back to MS DOS. Restore tape to disk. Make final modifications and go back to step 3.
11. Run FINALHS or FINALISO to convert HS/ISO partitions into final format.
12. Verify final disc image using SIM.
13. Create pre-master tape with MASTERHS.

Source: Meridian Data, Inc. CD Publisher User Manual

Table 6-5. PROCEDURE 2 FOR DEVELOPING PRE-MASTER TAPE

Appendix A

partition to a HS/ISO partition on the CD Publisher. The program must be executed after INITHS/INITISO and before FINALHS/FINALISO. SIM is used to invoke and revoke a CD-ROM timing simulation. When executed, it causes the partition driver to respond to requests for data at the speed at which the CD-ROM playback unit would provide the data. This feature is valuable in determining the approximate performance characteristics the final CD-ROM will have. FINALHS/FINALISO transforms the extended High Sierra/ISO 9660 directory structures into a conventional High Sierra/ISO 9660 format. No further file modifications are possible after executing this program. SIM must be revoked prior to executing FINALHS/FINALISO. MASTERHS is used to generate a 9-track pre-mastering tape from a High Sierra or ISO 9660 partition. This is the tape which is sent to a CD-ROM mastering facility.

There are approximately 40 more utility programs and several popular CD-ROM drivers also provided with the CD Publisher system. The programs described in the previous procedures are those which are essential to the pre-mastering process. The other programs serve useful functions in various unique situations.

CD-ROM vs PAPER MANUAL TEST RESULTS

A. TEST PROCEDURES

A sample group of five individuals was used to compare the average access times for retrieving information from both the Lineal List manual and the CD-ROM pre-master application. Each person was given three tasks to accomplish:

- 1) Find an officer's lineal number given his/her name.
- 2) Find an officer's name when given a lineal number.
- 3) Count the number of officers with a specific designator within a given range of lineal numbers.

The test data given to each subject was identical. Each task was timed separately. The CD Publisher SIM program was used to simulate the response times of an actual CD-ROM playback unit.

B. TEST RESULTS

The CD-ROM application outperformed manual look-ups in every instance. Test results are as shown in Tables B-1 and B-2.

TABLE B-1. PAPER MANUAL TEST RESULTS

TEST SUBJECT	FIND LINEAL # FOR: DANIEL, THOMAS E	FIND NAME FOR: L006830-50	FIND QTY OF DESIG: 13xx BTWN L006800-20 & L006838-90
NUMBER 1	27 SEC	9 MIN, 43 SEC	3 MIN, 48 SEC
NUMBER 2	55 SEC	8 MIN, 8 SEC	5 MIN, 25 SEC
NUMBER 3	42 SEC	8 MIN, 50 SEC	3 MIN, 48 SEC
NUMBER 4	53 SEC	9 MIN, 22 SEC	5 MIN, 03 SEC
NUMBER 5	31 SEC	7 MIN, 36 SEC	4 MIN, 10 SEC
AVERAGE	41.6 SEC	8 MIN, 44 SEC	4 MIN, 32 SEC

TABLE B-2. SIMULATED CD-ROM TEST RESULTS

TEST SUBJECT	FIND LINEAL # FOR: DANIEL, THOMAS E	FIND NAME FOR: L006830-50	FIND QTY OF DESIG: 13xx BTWN L006800-20 & L006838-90
NUMBER 1	39 SEC	35 SEC	2 MIN, 24 SEC
NUMBER 2	37 SEC	33 SEC	2 MIN, 22 SEC
NUMBER 3	42 SEC	37 SEC	2 MIN, 27 SEC
NUMBER 4	36 SEC	33 SEC	2 MIN, 24 SEC
NUMBER 5	39 SEC	36 SEC	2 MIN, 26 SEC
AVERAGE	38.2 SEC	34.8 SEC	2 MIN, 25 SEC

Object Definitions for CODIS

COMMAND OBJECT

Unit identification code; UIC
Name; Activity-title
Type of duty; Sea-shore-code
Homeport; Homeport
Address; Address
OFFICER; OFFICER object; MV

OFFICER OBJECT

Name; Officer-name
Sex; Sex
Rank; Rank
Date of rank; DOR
Lineal number; Lineal-number
Year group; Year-group
Designator; Designator
Commissioning source; Source-code
Date of birth; DOB
Reporting date; Reporting-date
Projected rotation date; PRD
Subspecialty code; Subspecialty-code
Billet sequence code; BSC
Billet title; Billet-title
COMMAND; COMMAND object

Domain Definitions for CODIS**Activity-title:**

Text 16

Official name of a military command

Address:

Text 132, Mask:

Address-line-count Numeric 1

Address-line-1 Text 30

Address-line-2 Text 30

Address-line-3 Text 30

Address-line-4 Text 30

State Text 2

Zip Text 5

Zip-extent Text 4

Complete mailing address of a military command

Billet-title:

Text 14

Official title of a billet

BSC:

Text 5

Unique billet sequence code assigned to each billet

Designator:

Numeric 4

General specialty category of officer

DOB:

Numeric 6, Mask:

Year Numeric 2

Month Numeric 2

Day Numeric 2

Date of birth of officer

DOR:

Numeric 6, Mask:

Year Numeric 2

Month Numeric 2

Day Numeric 2

Date of present rank

Homeport:

Text 6

Location where a military command is stationed

Lineal-number:

Text 9, Mask: NNNNNN-NN,

where N is a number
Unique precedence number assigned to each officer

Officer-name:

Text 27
Names of officers in the US Navy

PRD:

Numeric 6, Mask:
Year Numeric 2
Month Numeric 2
Projected rotation date

Rank:

Text 4
Present rank of an officer

Reporting-date:

Numeric 6, Mask:
Year Numeric 2
Month Numeric 2
Day Numeric 2
Date officer reported to current command

Sea-shore-code:

Text 1
Type of duty

Sex:

Text 1
Gender of an officer

Source-code:

Numeric 3
Procurement program under which officer qualified for
current appointment

Subspecialty-code:

Text 15, Mask:
First Text 5
Second Text 5
Third Text 5
Training or experience subspecialty qualifications of
officer

UIC:

Text 5
Unique Unit Identification Code assigned to each command

Appendix C

Year-group:

Numeric 3

Fiscal year in which officer was commissioned an Ensign

Update, Display, and Control Mechanisms

A summary of the update, display, and control mechanisms for each object are described below. This structure is defined by a single object view.

UPDATE MECHANISMS

Once the database is physically placed on a CD-ROM, users are unable to create, modify, or delete any data. All necessary creations, modifications, and deletions must be performed by NMPC to their master database. Each time a new CD-ROM database produced, a new copy of the data containing all changes must be requested from NMPC.

DISPLAY MECHANISMS

I. OFFICER

- A. Description:** OFFICER queries will access all instances of the OFFICER object contained in the database.
- B. Source data:** OFFICER containing the object COMMAND.
- C. Output:** Output may take the form of user generated reports, system query responses, or exportable text files.
- D. Frequency:** As requested.

II. COMMAND

- A. **Description:** COMMAND queries will access all instances of the COMMAND object contained in the database.
- B. **Source data:** COMMAND containing the object OFFICER.
- C. **Output:** Output may take the form of user generated reports, system query responses, or exportable text files.
- D. **Frequency:** As requested.

CONTROL MECHANISMS

Database control is accomplished by physically limiting access to the database, processing application limitations, and restricted processing via menus.

Unauthorized modification of the database can be controlled by physically locating the resource on a CD-ROM and limiting distribution to users who are currently on the paper publication distribution list. This distribution can be further restricted to those users possessing an IBM-compatible microcomputer and a CD-ROM reader.

Processing application restrictions are enforced by disallowing any additions, modifications, or deletions to the objects within the database. There are no query or view restrictions.

CODIS Relation DefinitionCOMMAND OBJECT

<u>ITEM</u>	<u>LENGTH</u>	<u>TYPE</u>
UIC	5	Character
ADDR_LINES	1	Numeric
ADDR1	30	Character
ADDR2	30	Character
ADDR3	30	Character
ADDR4	30	Character
STATE	2	Character
ZIP	5	Character
ZIP_EXT	4	Character
CONTINENT	2	Character
ST_COUNTRY	2	Character
CITY	4	Character
ADC	1	Character
SEA_SHORE	1	Character
AMC	2	Character
HOMEPORT	6	Character
ACT_TITLE	16	Character

CODIS Relation DefinitionOFFICER OBJECT

<u>ITEM</u>	<u>LENGTH</u>	<u>TYPE</u>
NAME	27	Character
RANK	4	Character
LINEAL_NBR	9	Character
YEAR_GROUP	3	Character
DESIGNATOR	4	Character
SOURCE_COD	3	Character
SEX	1	Character
DOB	6	Character
SPECIALTY1	5	Character
SPECIALTY2	5	Character
SPECIALTY3	5	Character
DOR	6	Character
DATE_RPTD	6	Character
PRD	4	Character
BSC	5	Character
BILLET_TTL	14	Character
UIC	5	Character

dBase IV File StructuresUIC.DBF

Structure for database: UIC.DBF

Number of data records: 13126

Date of last update : 07/28/90

Field	Field Name	Type	Width	Dec	Index
1	UIC	Character	5		Y
2	ADDR_LINES	Numeric	1		Y
3	ADDR1	Character	30		Y
4	ADDR2	Character	30		Y
5	ADDR3	Character	30		Y
6	ADDR4	Character	30		Y
7	STATE	Character	2		Y
8	ZIP	Character	5		Y
9	ZIP_EXT	Character	4		Y
10	CONTINENT	Character	2		Y
11	ST_COUNTRY	Character	2		Y
12	CITY	Character	4		Y
13	ADC	Character	1		Y
14	SEA_SHORE	Character	1		Y
15	AMC	Character	2		Y
16	HOMEPORT	Character	6		Y
17	ACT_TITLE	Character	16		Y
** Total **			172		

dBse IV File StructuresOFFICER.DBF

Structure for database: OFFICER.DBF

Number of data records: 72255

Date of last update : 08/01/90

Field	Field Name	Type	Width	Dec	Index
1	NAME	Character	27		Y
2	RANK	Character	4		Y
3	LINEAL_NBR	Character	9		Y
4	YEAR_GROUP	Character	3		Y
5	DESIGNATOR	Character	4		Y
6	SOURCE_COD	Character	3		Y
7	SEX	Character	1		Y
8	DOB	Character	6		Y
9	SPECIALTY1	Character	5		Y
10	SPECIALTY2	Character	5		Y
11	SPECIALTY3	Character	5		Y
12	DOR	Character	6		Y
13	DATE_RPTD	Character	6		Y
14	PRD	Character	4		Y
15	BSC	Character	5		Y
16	ACT_CAT	Character	1		Y
17	DT_ASGN_BL	Character	4		Y

dBase IV File StructuresOFFICER.DBF (continued)

Field	Field Name	Type	Width	Dec	Index
18	BILLET_TTL	Character	14		Y
19	BSC2	Character	5		Y
20	PDS_FROM	Character	4		Y
21	PDS_TO	Character	4		Y
22	ACT_TITLE	Character	16		Y
23	TYPE_ASSIG	Character	1		Y
24	HOMEPORT	Character	6		Y
25	UIC	Character	5		Y
26	SEA_SHORE	Character	5		Y
**	Total	**	159		

dBase IV PROGRAM CODE

```
* Program...: MAIN.PRG
* Menu.....: Main Menu
* Author....: LT Paul G. Scyoc
* Date.....: 09-03-1990
*
* Menu Title: Command and Officer Data Information System
* SubTitle 1: CODIS
* SubTitle 2: Main Menu
* Purpose...: Main program to access the CODIS database.
*
* Merge Code Area 1
LSTPGM="MAIN"
* Set Up Working Environment
SET TALK OFF
SET HEADING OFF
SET SAFETY OFF
SET STATUS OFF
CLEAR
SET BELL OFF
SET SCOREBOARD OFF
*
* Background Color: Black (N)
* Foreground Color: Cyan (BG)
* Backlight Color: Green (G)
* LightBar Color: Cyan (BG)
*
SET COLOR TO BG/N
* Variables will start with the letter "A" so other variables
in memory
* will not be overwritten.
*
* Initialize User Input Variable To Zero
AEL=0
AENU=.T.
* AXP1 & AXP2 contain user instructions.
AXP1=" "+CHR(24)+CHR(25)+" Change Selection "
AXP2=" "+CHR(17)+CHR(196)+CHR(217)+" Make Selection "
* Initialize Menu Choice Variables
APT1=" B - Billets "
ASG1="Billets by UIC or BSC."
APT2=" C - Commands "
ASG2="Commands by geographic location or UIC."
APT3=" O - Officer Data "
ASG3="Data on a particular officer."
APT4=" P - Precedence List "
ASG4="Officers by Lineal Number."
APT5=" S - Subspecialty Coded Officers "
```

dBase IV PROGRAM CODE

```

ASG5="Officers with a particular subspecialty code."
APT6=" X - Exit "
ASG6="Exit to DOS"
* Set Number Of Options
AUMOP=6
* Initialize Choice Indicator To #1
ASG=1
* Initialize Column Position and Options/Column
ALM1= 6
ALM2= 0
AIDDL=40-(LEN(APT5)/2)
AAIDDL=40-(LEN(ASG5)/2)

DO WHILE AENU
* Merge Code Area 2
* Set Up Menu Screen
CLEAR
@ 1,01 SAY "Main Menu"
@ 1,70 SAY DATE()
@ 1,18 SAY "Command and Officer Data Information System"
@ 2,37 SAY "CODIS"
@ 3,35 SAY "Main Menu"
SET COLOR TO G/N
@ 0,16 TO 4,62 DOUBLE
* Merge Code Area 3

@ 7,AIDDL-2 TO 8+ALM1,(AIDDL+(LEN(APT5))+1) DOUBLE
@ ALM1+10,AAIDDL-2 TO ALM1+12,(AAIDDL+(LEN(ASG5))+1)
* Display User Instructions/Help Window
SET COLOR TO N/BG
@ 22,16 SAY AXP1
@ 22,40 SAY AXP2
SET COLOR TO BG/N

* Display Menu Choices

@ 8,AIDDL SAY APT1
@ 9,AIDDL SAY APT2
@ 10,AIDDL SAY APT3
@ 11,AIDDL SAY APT4
@ 12,AIDDL SAY APT5
@ 13,AIDDL SAY APT6
ACHOICE=.T.

DO WHILE ACHOICE
* Merge Code Area 4

```

dBase IV PROGRAM CODE

* Determine which Help To Display from the Message

```
AMP="ASG"+LTRIM(STR(ASG))
SET COLOR TO BG/N
@ AUMOP+11,AAIDDL SAY SPACE(LEN(ASG5))
@ AUMOP+11,AAIDDL SAY &AMP
APT="APT"+LTRIM(STR(ASG))
SET COLOR TO N/BG
@ ASG+7,AIDDL SAY &APT
SET COLOR TO BG/N
AEL=0
```

* Keyboard Key Trapping

```
DO WHILE AEL=0
```

* Merge Code Area 5

```
AEL= INKEY()
ENDDO
@ ASG+7,AIDDL SAY &APT
```

* Check If Valid Key Was Depressed

```
IF CHR(AEL) $ "BbCcOoPpSsXx"
```

* Merge Code Area 6

```
IF CHR(AEL) $ "Bb"
  ASG= 1
ENDIF
```

```
IF CHR(AEL) $ "Cc"
  ASG= 2
ENDIF
```

```
IF CHR(AEL) $ "Oo"
  ASG= 3
ENDIF
```

```
IF CHR(AEL) $ "Pp"
  ASG= 4
ENDIF
```

```
IF CHR(AEL) $ "Ss"
  ASG= 5
ENDIF
```

```
IF CHR(AEL) $ "Xx"
  ASG= 6
ENDIF
```

dBase IV PROGRAM CODE

```
        EXIT
    ENDIF

* Carrage Return Depressed?

    IF AEL = 13
        EXIT
    ENDIF

* Up Arrow Depressed?

    IF AEL = 5
        ASG=ASG-1

        IF ASG<1
            ASG=AUMOP
        ENDIF
    ENDIF

* Down Arrow Depressed?

    IF AEL =24
        ASG=ASG+1

        IF ASG>AUMOP
            ASG=1
        ENDIF
    ENDIF
ENDDO

* Merge Code Area 7

* Perform Item Selected

    DO CASE

        CASE ASG=1
            CLEAR
            DO BILLETS

        CASE ASG=2
            CLEAR
            DO COMMANDS

        CASE ASG=3
            CLEAR
            DO OFFICER
```

dBase IV PROGRAM CODE

```

CASE ASG=4
  CLEAR
  DO LIN_NBR

CASE ASG=5
  CLEAR
  DO SPECIALT

CASE ASG=6
  CLEAR
  CLOSE ALL
  QUIT
ENDCASE
* Merge Code Area 8
  ENDDO
RETURN
* EOF: MAIN.PRG

* Program...: RPT_MENU.PRG
* Menu.....: Output Menu
* Author....: LT Paul G. SCYOC
* Date.....: 09-03-1990
*
* Menu Title: Command and Officer Data Information System
* SubTitle 1: CODIS
* SubTitle 2: Output Media Selection Menu
* Purpose...: Select Output Media
*
* Merge Code Area 1
LSTPGM="RPT_MENU"
* Set Up Working Environment
SET TALK OFF
SET HEADING OFF
SET SAFETY OFF
SET STATUS OFF
CLEAR
SET BELL OFF
SET SCOREBOARD OFF
*
PUBLIC mwhere, mrpt, mfile, dsg
*
* Background Color: Blue (B)
* Foreground Color: White (W)
* Backlight Color: Black (N)
* LightBar Color: White (W)
*
```

dBase IV PROGRAM CODE

```

SET COLOR TO W/B
* Variables will start with the letter "D" so other variables
in memory
* will not be overwritten.
*
* Initialize User Input Variable To Zero
DEL=0
DENU=.T.
* DXP1 & DXP2 contain user instructions.
DXP1=" "+CHR(24)+CHR(25)+" Change Selection "
DXP2=" "+CHR(17)+CHR(196)+CHR(217)+" Make Selection  "
* Initialize Menu Choice Variables
DPT1=" 1 - Screen "
DSG1="Send report to the screen."
DPT2=" 2 - Printer "
DSG2="Send report to the printer."
DPT3=" 3 - Report File"
DSG3="Send report to a file."
DPT4=" 4 - dBase IV File "
DSG4="Send data to a dBase IV file."
DPT5=" 5 - ASCII File "
DSG5="Send data to an ASCII text file."
* Set Number Of Options
DUMOP=5
* Initialize Choice Indicator To #1
DSG=1
* Initialize Column Position and Options/Column
DLM1= 5
DLM2= 0
DIDDL=40-(LEN(DPT4)/2)
DDIDDL=40-(LEN(DSG5)/2)

DO WHILE DENU
* Merge Code Area 2
* Set Up Menu Screen
  CLEAR
  @ 1,18 SAY "Command and Officer Data Information System"
  @ 2,37 SAY "CODIS"
  @ 3,30 SAY "Output Media Selection Menu"
  SET COLOR TO N/B
  @ 0,16 TO 4,62 DOUBLE
* Merge Code Area 3

  @ 8,DIDDL-2 TO 9+DLM1,(DIDDL+(LEN(DPT4))+1) DOUBLE
  @ DLM1+11,DDIDDL-2 TO DLM1+13,(DDIDDL+(LEN(DSG5))+1)
* Display User Instructions/Help Window
  SET COLOR TO B/W

```

dBase IV PROGRAM CODE

```
@ 22,16 SAY DXP1
@ 22,40 SAY DXP2
SET COLOR TO W/B
```

* Display Menu Choices

```
@ 9,DIDDL SAY DPT1
@ 10,DIDDL SAY DPT2
@ 11,DIDDL SAY DPT3
@ 12,DIDDL SAY DPT4
@ 13,DIDDL SAY DPT5
```

* Merge Code Area 4

* Determine which Help To Display from the Message

```
DMP="DSG"+LTRIM(STR(DSG))
SET COLOR TO W/B
@ DUMOP+12,DDIDDL SAY SPACE(LEN(DSG5))
@ DUMOP+12,DDIDDL SAY &DMP
DPT="DPT"+LTRIM(STR(DSG))
SET COLOR TO B/W
@ DSG+8,DIDDL SAY &DPT
SET COLOR TO W/B
DEL=0
```

* Keyboard Key Trapping

```
DO WHILE DEL=0
```

* Merge Code Area 5

```
DEL= INKEY()
ENDDO
@ DSG+8,DIDDL SAY &DPT
```

* Check If Valid Key Was Depressed

```
IF CHR(DEL) $ "12345"
```

* Merge Code Area 6

```
IF CHR(DEL) $ "1"
DSG= 1
ENDIF
```

```
IF CHR(DEL) $ "2"
DSG= 2
ENDIF
```

```
IF CHR(DEL) $ "3"
DSG= 3
```

dBase IV PROGRAM CODE

```
ENDIF
    IF CHR(DEL) $ "4"
        DSG= 4
    ENDIF
    IF CHR(DEL) $ "5"
        DSG= 5
    ENDIF
EXIT
ENDIF
* Carrage Return Depressed?
    IF DEL = 13
        EXIT
    ENDIF
* Up Arrow Depressed?
    IF DEL = 5
        DSG=DSG-1
        IF DSG<1
            DSG=DUMOP
        ENDIF
    ENDIF
* Down Arrow Depressed?
    IF DEL =24
        DSG=DSG+1
        IF DSG>DUMOP
            DSG=1
        ENDIF
    ENDIF
ENDDO
* Merge Code Area 7
* Perform Item Selected
    DO CASE
CASE DSG=1
    CLEAR
    STORE " " TO mwhere
```

dBase IV PROGRAM CODE

```

STORE "Y" TO mrpt

CASE DSG=2
  CLEAR
  STORE "TO PRINT NOEJECT" TO mwhere
  STORE "Y" TO mrpt

CASE DSG=3
  CLEAR
  ACCEPT "ENTER DRIVE & FILE FOR OUTPUT REPORT FILE: "
TO mfile
  STORE "TO FILE " + mfile TO mwhere
  STORE "Y" TO mrpt

CASE DSG=4
  CLEAR
  STORE "N" TO mrpt
  ACCEPT "ENTER DRIVE & FILE FOR OUTPUT DBASE FILE: "
TO mfile

CASE DSG=5
  CLEAR
  STORE "N" TO mrpt
  ACCEPT "ENTER DRIVE & FILE FOR OUTPUT ASCII FILE: "
TO mfile

ENDCASE
* Merge Code Area 8
RETURN
* EOF: RPT_MENU.PRG

* Program...: BILLETS.PRG
* Author....: LT Paul G. Scyoc
* Date.....: 09-03-1990
*
* Purpose...: Program to produce the Billet reports.
*
set talk off
* Query by BSC or UIC?
store " " to CHOICE
store 0 to MFOUND
clear
@ 5, 10 say "Look up billets by BSC (B) or UIC (U)?"
@ 7, 20 say "(enter B or U)"
@ 5, 55 get CHOICE

```

dBase IV PROGRAM CODE

```

read
store upper(CHOICE) to CHOICE

* Select Report Media
if CHOICE = "B" .or. CHOICE = "U"
  do RPT_MENU
else
  @ 8, 32 say "Invalid Entry!!"
  wait
endif

* Look up Billets by UIC
if CHOICE = "U"
  store " " to MUIC
  store 0 to MNBR_UIC
  clear
  use OFFICER order UIC
  @ 5, 10 say "Enter UIC to find billets for: "
  @ 5, 50 get MUIC
  read
  find &MUIC

  if found()
    store 1 to MFOUND
    count to MNBR_UIC while UIC = MUIC
  endif

  @ 10, 10 say "The UIC " + MUIC + " has "
  ?? ltrim(str(MNBR_UIC))
  ?? " billet(s)."
  ?
  ?
  wait

  if MFOUND = 1

    if MRPT <> "N"
      find &MUIC
      report form BLT_UIC &MWHERE while UIC = MUIC
      wait

    else

      if MFILE <> " "
        find &MUIC
        set talk on
        copy to &MFILE while UIC = MUIC

```

dBase IV PROGRAM CODE

```

set talk off

if DSG = 5
  use &MFILE
  copy to &MFILE SDF
  store &mfile + ".dbf" to MFILE
  use
  delete file &MFILE
endif

endif

endif

endif

* Look up Billets by BSC
if CHOICE = "B"

  store "      " to MBSC
  store 0 to MNBR_BSC
  clear
  use OFFICER order BSC2
  @ 5, 5 say "Enter BSC of BILLET to be found: "
  @ 5, 38 get MBSC
  read
  store upper(MBSC) to MBSC
  find &MBSC

  if found()
    store 1 to MFOUND
    count to MNBR_BSC while BSC2 = MBSC
  endif

  @ 10, 5 say "The BSC " + MBSC + " was found "
  ?? ltrim(str(MNBR_BSC))
  ?? " time(s). "
  ?
  ?
  wait

  if MFOUND = 1

    if MRPT <> "N"
      find &MBSC
    endif
  endif
endif

```

dBase IV PROGRAM CODE

```
report form BLT_BSC &MWHERE while BSC2 = MBSC
wait
else

  if MFILE <> " "
    find &MBSC
    set talk on
    copy to &MFILE while BSC2 = MBSC
    set talk off

    if DSG = 5
      use &MFILE
      copy to &MFILE SDF
      store &mfile + ".dbf" to MFILE
      use
      delete file &MFILE
    endif

  endif

endif

endif

endif

close all
return
* EOF: BILLETS.PRG

* Program...: COMMANDS.PRG
* Author....: LT Paul G. Scyoc
* Date.....: 09-03-1990
*
* Purpose...: Program to produce the Command report.
*
set talk off
* Query by city or UIC?
store " " to CHOICE
store 0 to MFOUND1
store 0 to MFOUND2
clear
@ 5, 10 say "Look up commands by City (C) or UIC (U)?"
@ 7, 20 say "(enter C or U)"
@ 5, 55 get CHOICE
```

dBase IV PROGRAM CODE

```

read
store upper(CHOICE) to CHOICE

* Select Report Media
if CHOICE = "C" .or. CHOICE = "U"
  do RPT_MENU
else
  @ 8, 32 say "Invalid Entry!!"
  wait
endif

* Look up Command by UIC
if CHOICE = "U"
  store " " to MUIC
  store 0 to MNBR_UIC
  clear
  use UIC order UIC
  @ 5, 10 say "Enter UIC of Command to be found: "
  @ 5, 50 get MUIC
  read
  find &MUIC
  if found()
    store 1 to MFOUND
    count to MNBR_UIC while UIC = MUIC
  endif
  @ 10, 10 say "The UIC " + MUIC + " was found "
  ?? ltrim(str(MNBR_UIC))
  ?? " times."
  ?
  ?
  wait

  if MFOUND = 1

    if MRPT <> "N"
      find &MUIC
      report form CMD_UIC &MWHERE while UIC = MUIC
      wait

    else

      if mfile <> " "
        find &MUIC
        set talk on
        copy to &MFILE while UIC = MUIC
        set talk off

```

dBase IV PROGRAM CODE

```

    if DSG = 5
        use &MFILE
        copy to &MFILE SDF
        store mfile + ".dbf" to MFILE
        use
        delete file &MFILE
    endif

endif

endif

endif

endif

* Look up Command by City
if CHOICE = "C"
    store " " to MCITY_NAME
    store " " to MCITY
    store 0 to MNBR_CITY
    store 0 to MNBR_CITY2
    store 0 to MTTL_CITY
    clear
    use UIC order ADDR3
    @ 5, 5 say "Enter name of CITY to be found: "
    @ 5, 38 get MCITY_NAME
    read
    store upper(trim(MCITY_NAME)) to MCITY
    find &MCITY
    if found()
        store 1 to MFOUND1
        count to MNBR_CITY while ADDR3=MCITY
    endif
    *
    set order to ADDR4
    find &MCITY
    if found()
        store 1 to MFOUND2
        count to MNBR_CITY2 while ADDR4=MCITY
    endif
    store MNBR_CITY + MNBR_CITY2 to MTTL_CITY
    @ 10, 5 say "The city of " + MCITY + " was found "
    ?? ltrim(str(MTTL_CITY))
    ?? " time(s)."
    ?

```

dBase IV PROGRAM CODE

```
?  
wait  
  
if MFOUND1 = 1 .or. MFOUND2 = 1  
  if MRPT <> "N"  
    if MFOUND1 = 1  
      set order to ADDR3  
      find &MCITY  
      report form CMD_CITY &MWHERE while ADDR3 = MCITY  
      wait  
    endif  
  
    if MFOUND2 = 1  
      set order to ADDR4  
      find &MCITY  
      wait  
      report form CMD_CITY &MWHERE while ADDR4 = MCITY  
    endif  
  
  else  
  
    if mfile <> " "  
      set order to ADDR3  
      find &MCITY  
      copy to TEMP1 while ADDR3 = MCITY  
      *  
      set order to ADDR4  
      find &MCITY  
      copy to TEMP2 while ADDR4 = MCITY  
      *  
      copy structure to &MFILE  
      use &MFILE  
      set talk on  
      append from TEMP1  
      append from TEMP2  
      set talk off  
      delete file TEMP1.DBF  
      delete file TEMP2.DBF  
      *  
      if DSG = 5  
        use &MFILE  
        copy to &MFILE SDF  
        store mfile + ".dbf" to MFILE  
        use  
        delete file &MFILE  
      endif  
    endif  
  
  endif  
endif
```

dBase IV PROGRAM CODE

```
endif
endif
endif
endif
endif
close all
return
* EOF:  COMMANDS.PRG

* Program...: OFFICER.PRG
* Author....: LT Paul G. Scyoc
* Date.....: 09-06-1990
*
* Purpose...: Program to produce the Officer Data report.
*
set talk off
set exact off
store space(27) to MNAME
clear
@ 5, 10 say "Enter the name of the officer:  "
@ 7, 15 say "(last first middle)"
@ 5, 47 get MNAME
read
store upper(trim(MNAME)) to MNAME
do RPT_MENU
use OFFICER order NAME
set talk on
find &MNAME

if found()

if MRPT <> "N"
report form OFFICER &MWHERE while NAME = MNAME

else

if MFILE <> " "
copy to &MFILE while NAME = MNAME

if DSG = 5
```

dBase IV PROGRAM CODE

```
        use &MFILE
        copy to &MFILE SDF
        store &MFILE + ".dbf" to MFILE
        use
        delete file &MFILE
    endif

endif

endif

endif

close all
set talk off
return
* EOF: OFFICER.PRG

* Program...: LIN_NBR.PRG
* Author....: LT Paul G. Scyoc
* Date.....: 09-03-1990
*
* Purpose...: Program to produce the Lineal Number reports.
*
set talk off
* Query by Lineal Number or Range of Lineal Numbers?
store " " to CHOICE
clear
@ 5, 6 say "Look up a Single Lineal Number (S) or a Range of
Numbers (R)?"
@ 7, 20 say "(enter S or R)"
@ 5, 72 get CHOICE
read
store upper(CHOICE) to CHOICE

* Select Report Media
if CHOICE = "S" .or. CHOICE = "R"
    do RPT_MENU

else

    @ 8, 32 say "Invalid Entry!!"
    ??
    ??
    wait
```

dBase IV PROGRAM CODE

endif

* Look up a single Lineal Number.

if CHOICE = "S"

store " " to MLINNBR

clear

use OFFICER order LIN_NBR

@ 5, 10 say "Enter last 8 digits of Lineal Number: "

@ 5, 50 get MLINNBR

read

find &MLINNBR

?

?

wait

if found()

if MRPT <> "N"

find &MLINNBR

report form LIN_NBR &MWHERE while substr(LINEAL_NBR,2,8)
= MLINNBR

wait

else

if MFILE <> " "

find &MLINNBR

copy to &MFILE while substr(LINEAL_NBR,2,8) = MLINNBR

if DSG = 5

use &MFILE

copy to &MFILE SDF

store MFILE + ".dbf" to MFILE

use

delete file &MFILE

endif

endif

endif

endif

endif

* Look up a range of Lineal Numbers.

if CHOICE = "R"

dBase IV PROGRAM CODE

```

clear
set exact off
store " " to FIRST
store " " to LAST
store " " to DESIG
store 0 to MNBR_LIN
store 0 to MFOUND
@ 5, 17 say "Input first Lineal Number of the range: "
@ 5, 57 get FIRST
@ 7, 17 say "Input last Lineal Number of the range: "
@ 7, 57 get LAST
@ 9, 17 say "Input first 2 digits of designator: "
@ 9, 57 get DESIG
read
store DESIG + FIRST to TARGET
use OFFICER order DESIG_LIN
find &TARGET

if found()
  store 1 to MFOUND
  count to MNBR_LIN while DESIGNATOR=DESIG .and. substr
    (LINEAL_NBR,2,8) <= LAST
endif

@ 15, 5 say "In the range " + FIRST
?? " thru " + LAST + ", "
@ 17,5 say ltrim(str(MNBR_LIN))
?? " officers were found with the designator " + DESIG
?? "xx."
?
?
wait

if MFOUND = 1
  find &TARGET

  if MRPT <> "N"
    report form LIN_NBR &MWHERE while DESIGNATOR=DESIG .and.
      substr(LINEAL_NBR,2,8) <= LAST
    wait
  else

    if MFILE <> " "
      copy to &MFILE while DESIGNATOR=DESIG .and. substr
        (LINEAL_NBR,2,8) <= LAST
    endif
  endif
endif

```

dBase IV PROGRAM CODE

```
    if DSG = 5
        use &MFILE
        copy to &MFILE SDF
        store MFILE + ".dbf" to MFILE
        use
        delete file &MFILE
    endif

endif

endif

endif

endif

close all
return
* EOF:  LIN_NBR.PRG

* Program...:  SPECIALT.PRG
* Author....:  LT Paul G. Scyoc
* Date.....:  09-06-1990
*
* Purpose...:  Program to produce the Officer Subspecialty
* report.
*
set talk off
set exact off
store "      " to MSPECIALT
store 0 to MSPEC
store 0 to MSPEC_TTL
store 0 to MFOUND1
store 0 to MFOUND2
store 0 to MFOUND3
clear
@ 5, 10 say "Enter Subspecialty Code:  "
@ 5, 47 get MSPECIALT
read
store upper(MSPECIALT) to MSPECIALT
do RPT_MENU
use OFFICER order SPECIALTY1
find &MSPECIALT

if found()
```

dBase IV PROGRAM CODE

```

store 1 to MFOUND1
count to MSPEC while SPECIALTY1 = MSPECIALT
store MSPEC to MSPEC_TTL
endif

set order to SPECIALTY2
find &MSPECIALT

if found()
store 1 to MFOUND2
count to MSPEC while SPECIALTY2 = MSPECIALT
store MSPEC_TTL + MSPEC to MSPEC_TTL
endif

set order to SPECIALTY3
find &MSPECIALT

if found()
store 1 to MFOUND3
count to MSPEC while SPECIALTY3 = MSPECIALT
store MSPEC_TTL + MSPEC to MSPEC_TTL
endif

@ 10,5 say "The subspecialty code " + MSPECIALT + " was found
"
?? MSPEC_TTL
?? " time(s)."
?
?
wait

if MFOUND1 = 1 .or. MFOUND2 = 1 .or. MFOUND3 = 1

if MRPT <> "N"

if MFOUND1 = 1
set order to SPECIALTY1
find &MSPECIALT
report form SPECIALT &MWHERE while SPECIALTY1=MSPECIALT
wait
endif

if MFOUND2 = 1
set order to SPECIALTY2
find &MSPECIALT
report form SPECIALT &MWHERE while SPECIALTY2=MSPECIALT
wait

```

dBase IV PROGRAM CODE

```

endif

if MFOUND3 = 1
  set order to SPECIALTY3
  find &MSPECIALT
  report form SPECIALT &MWHERE while SPECIALTY3=MSPECIALT
  wait
endif

else

if MFILE <> " "
  set order to SPECIALTY1
  find &MSPECIALT
  copy to TEMP1 while SPECIALTY1 = MSPECIALT
  *
  set order to SPECIALTY2
  find &MSPECIALT
  copy to TEMP2 while SPECIALTY2 = MSPECIALT
  *
  set order to SPECIALTY3
  find &MSPECIALT
  copy to TEMP3 while SPECIALTY3 = MSPECIALT
  *
  copy structure to &MFILE
  use &MFILE
  append from TEMP1
  append from TEMP2
  append from TEMP3
  delete file TEMP1.DBF
  delete file TEMP2.DBF
  delete file TEMP3.DBF
  *
  if DSG = 5
    use &MFILE
    copy to &MFILE SDF
    store MFILE + ".dbf" to MFILE
    use
    delete file &MFILE
  endif
endif

endif

endif

endif

```

dBase IV PROGRAM CODE

```

close all
return
* EOF: SPECIALT.PRG

* Program.....: BLT_BSC.FRG
* Date.....: 9-10-90
*
* Notes:
* -----
* Prior to running this procedure with the DO command
* it is necessary use LOCATE because the CONTINUE
* statement is in the main loop.
*
*-- Parameters
PARAMETERS gl_noeject, gl_plain, gl_summary, gc_heading,
gc_extra
** The first three parameters are of type Logical.
** The fourth parameter is a string. The fifth is extra.
PRIVATE _peject, _wrap

*-- Test for no records found
IF EOF() .OR. .NOT. FOUND()
    RETURN
ENDIF

*-- turn word wrap mode off
_wrap=.F.

IF _plength < 1 + 1 + 2
    SET DEVICE TO SCREEN
    DEFINE WINDOW gw_report FROM 7,17 TO 11,62 DOUBLE
    ACTIVATE WINDOW gw_report
    @ 0,1 SAY "Increase the page length for this report."
    @ 2,1 SAY "Press any key ..."
    x=INKEY(0)
    DEACTIVATE WINDOW gw_report
    RELEASE WINDOW gw_report
    RETURN
ENDIF

_plineno=0          && set lines to zero
*-- NOEJECT parameter
IF gl_noeject
    IF _peject="BEFORE"
        _peject="NONE"
    
```

dBase IV PROGRAM CODE

```

ENDIF
  IF _peject="BOTH"
    _peject="AFTER"
  ENDIF
ENDIF

*-- Set-up environment
ON ESCAPE DO Prnabort
IF SET("TALK")="ON"
  SET TALK OFF
  gc_talk="ON"
ELSE
  gc_talk="OFF"
ENDIF
gc_space=SET("SPACE")
SET SPACE OFF
gc_time=TIME()      && system time for predefined field
gd_date=DATE()      && system date " " " "
gl_fandl=.F.        && first and last page flag
gl_prntflg=.T.      && Continue printing flag
gl_widow=.T.        && flag for checking widow bands
gn_length=LEN(gc_heading) && store length of the HEADING
gn_level=2          && current band being processed
gn_page=_pageno    && grab current page number
gn_pspace=_pspacing && get current print spacing

*-- Set up procedure for page break
gn_atline=_plength - 1
ON PAGE AT LINE gn_atline EJECT PAGE

*-- Print Report

PRINTJOB

IF gl_plain
  ON PAGE AT LINE gn_atline DO Pgplain
ELSE
  ON PAGE AT LINE gn_atline DO Pgfoot
ENDIF

DO Pghead

gl_fandl=.T.        && first physical page started

DO Rintro

```

dBase IV PROGRAM CODE

```

*-- File Loop
DO WHILE FOUND() .AND. .NOT. EOF() .AND. gl_prntflg
  gn_level=0
  *-- Detail lines
  IF gl_summary
    DO Upd_Vars
  ELSE
    DO __Detail
  ENDIF
  gl_widow=.T.          && enable widow checking
  CONTINUE
ENDDO

IF gl_prntflg
ELSE
  DO Reset
  RETURN
ENDIF

ON PAGE

ENDPRINTJOB

DO Reset
RETURN
* EOP: BLT_BSC.FRG

*-- Update summary fields and/or calculated fields.
PROCEDURE Upd_Vars
RETURN
* EOP: Upd_Vars

*-- Set flag to get out of DO WHILE loop when escape is
pressed.
PROCEDURE Prnabort
gl_prntflg=.F.
RETURN
* EOP: Prnabort

PROCEDURE Pghead

*-- Print HEADING parameter ie. REPORT FORM <name> HEADING
<expC>
IF .NOT. gl_plain .AND. gn_length > 0
  ?? gc_heading FUNCTION "I;V"+LTRIM(STR(_rmargin-_lmargin))
  ?
ENDIF

```

dBase IV PROGRAM CODE

```

RETURN
* EOP: Pghead

PROCEDURE Rintro
?? "BILLETS BY NAVY COMMAND" STYLE "B" AT 29
?
?
?? "SEA/" STYLE "B" AT 73
?
?? "UIC" STYLE "B" AT 3,;
"COMMAND" STYLE "B" AT 16,;
"BSC" STYLE "B" AT 34,;
"BILLET TITLE" STYLE "B" AT 44,;
"HOMEPORT" STYLE "B" AT 61,;
"SHORE" STYLE "B" AT 73
?
RETURN
* EOP: Rintro

PROCEDURE __Detail
IF 2 * gn_pspace < gn_atline - 1
    IF gl_widow .AND. _plinen+2 * gn_pspace > gn_atline + 1
        EJECT PAGE
    ENDIF
ENDIF
DO Upd_Vars
?
?? Uic FUNCTION "T" AT 2,;
Act_title FUNCTION "T" AT 12,;
Bsc2 FUNCTION "T" AT 33,;
Billet_ttl FUNCTION "T" AT 43,;
Homeport FUNCTION "T" AT 62,;
Sea_shore FUNCTION "T" AT 73
?
RETURN
* EOP: __Detail

PROCEDURE Pgfoot
PRIVATE _box
gl_widow=.F.          && disable widow checking
EJECT PAGE
*-- is the page number greater than the ending page
IF _pageno > _pepage
    GOTO BOTTOM
    SKIP
    gn_level=0

```

dBase IV PROGRAM CODE

```

ENDIF
IF .NOT. gl_plain .AND. gl_fandl
    _pspacing=gn_pspace
    DO Pghead
ENDIF
RETURN
* EOP: Pgfoot

*-- Process page break when PLAIN option is used.
PROCEDURE Pgplain
PRIVATE _box
EJECT PAGE
RETURN
* EOP: Pgplain

*-- Reset dBASE environment prior to calling report
PROCEDURE Reset
SET SPACE &gc_space.
SET TALK &gc_talk.
ON ESCAPE
ON PAGE
RETURN
* EOP: Reset

* Program.....: BLT_UIC.FRG
* Date.....: 9-10-90
*
* Notes:
* -----
* Prior to running this procedure with the DO command
* it is necessary use LOCATE because the CONTINUE
* statement is in the main loop.
*
*-- Parameters
PARAMETERS gl_noeject, gl_plain, gl_summary, gc_heading,
gc_extra
** The first three parameters are of type Logical.
** The fourth parameter is a string. The fifth is extra.
PRIVATE _peject, _wrap

*-- Test for no records found
IF EOF() .OR. .NOT. FOUND()
    RETURN
ENDIF

```

dBase IV PROGRAM CODE

```

*-- turn word wrap mode off
_wrap=.F.

IF _plength < 1 + 1 + 2
  SET DEVICE TO SCREEN
  DEFINE WINDOW gw_report FROM 7,17 TO 11,62 DOUBLE
  ACTIVATE WINDOW gw_report
  @ 0,1 SAY "Increase the page length for this report."
  @ 2,1 SAY "Press any key ..."
  x=INKEY(0)
  DEACTIVATE WINDOW gw_report
  RELEASE WINDOW gw_report
  RETURN
ENDIF

_plineno=0          && set lines to zero
*-- NOEJECT parameter
IF gl_noeject
  IF _peject="BEFORE"
    _peject="NONE"
  ENDIF
  IF _peject="BOTH"
    _peject="AFTER"
  ENDIF
ENDIF

*-- Set-up environment
ON ESCAPE DO Prnabort
IF SET("TALK")="ON"
  SET TALK OFF
  gc_talk="ON"
ELSE
  gc_talk="OFF"
ENDIF
gc_space=SET("SPACE")
SET SPACE OFF
gc_time=TIME()      && system time for predefined field
gd_date=DATE()      && system date " " " "
gl_fandl=.F.        && first and last page flag
gl_prntflg=.T.      && Continue printing flag
gl_widow=.T.        && flag for checking widow bands
gn_length=LEN(gc_heading) && store length of the HEADING
gn_level=2          && current band being processed
gn_page=_pageno     && grab current page number
gn_pspace=_pspacing && get current print spacing

```

dBase IV PROGRAM CODE

```

*-- Set up procedure for page break
gn_atline=_plength - 1
ON PAGE AT LINE gn_atline EJECT PAGE

*-- Print Report

PRINTJOB

IF gl_plain
  ON PAGE AT LINE gn_atline DO Pgplain
ELSE
  ON PAGE AT LINE gn_atline DO Pgfoot
ENDIF

DO Pghead

gl_fandl=.T.          && first physical page started

DO Rintro

*-- File Loop
DO WHILE FOUND() .AND. .NOT. EOF() .AND. gl_prntflg
  gn_level=0
  *-- Detail lines
  IF gl_summary
    DO Upd_Vars
  ELSE
    DO __Detail
  ENDIF
  gl_widow=.T.        && enable widow checking
  CONTINUE
ENDDO

IF gl_prntflg
ELSE
  DO Reset
  RETURN
ENDIF

ON PAGE

ENDPRINTJOB

DO Reset
RETURN
* EOP: BLT_UIC.FRG

```

dBase IV PROGRAM CODE

```

*-- Update summary fields and/or calculated fields.
PROCEDURE Upd_Vars
RETURN
* EOP: Upd_Vars

*-- Set flag to get out of DO WHILE loop when escape is
pressed.
PROCEDURE Prnabort
gl_prntflg=.F.
RETURN
* EOP: Prnabort

PROCEDURE Pghead

*-- Print HEADING parameter ie. REPORT FORM <name> HEADING
<expC>
IF .NOT. gl_plain .AND. gn_length > 0
    ?? gc_heading FUNCTION "I;V"+LTRIM(STR(_rmargin-_lmargin))
    ?
ENDIF
RETURN
* EOP: Pghead

PROCEDURE Rintro
?? "BILLETS BY BSC" STYLE "B" AT 31
?
?
?? "SEA/" STYLE "B" AT 73
?
?? "UIC" STYLE "BU" AT 3,;
"COMMAND" STYLE "BU" AT 16,;
"BSC" STYLE "B" AT 34,;
"BILLET TITLE" STYLE "B" AT 44,;
"HOMEPORT" STYLE "B" AT 61,;
"SHORE" STYLE "B" AT 73
?
RETURN
* EOP: Rintro

PROCEDURE __Detail
IF 2 * gn_pspace < gn_atline - 1
    IF gl_widow .AND. _plinen+2 * gn_pspace > gn_atline + 1
        EJECT PAGE
    ENDIF
ENDIF
DO Upd_Vars
?
```

dBase IV PROGRAM CODE

```

?? Uic FUNCTION "T" AT 2,;
  Act_title FUNCTION "T" AT 12,;
  Bsc2 FUNCTION "T" AT 33,;
  Billet_ttl FUNCTION "T" AT 43,;
  Homeport FUNCTION "T" AT 62,;
  Sea_shore FUNCTION "T" AT 73
?
RETURN
* EOP: __Detail

PROCEDURE Pgfoot
PRIVATE _box
gl_widow=.F.          && disable widow checking
EJECT PAGE
*-- is the page number greater than the ending page
IF _pageno > _pepage
  GOTO BOTTOM
  SKIP
  gn_level=0
ENDIF
IF .NOT. gl_plain .AND. gl_fandl
  _pspacing=gn_pspace
  DO Pghead
ENDIF
RETURN
* EOP: Pgfoot

*-- Process page break when PLAIN option is used.
PROCEDURE Pgplain
PRIVATE _box
EJECT PAGE
RETURN
* EOP: Pgplain

*-- Reset dBASE environment prior to calling report
PROCEDURE Reset
SET SPACE &gc_space.
SET TALK &gc_talk.
ON ESCAPE
ON PAGE
RETURN
* EOP: Reset

```

dBase IV PROGRAM CODE

```

* Program.....: CMD_CITY.FRG
* Date.....: 9-10-90
*
* Notes:
* -----
* Prior to running this procedure with the DO command
* it is necessary use LOCATE because the CONTINUE
* statement is in the main loop.
*
*-- Parameters
PARAMETERS gl_noject, gl_plain, gl_summary, gc_heading,
gc_extra
** The first three parameters are of type Logical.
** The fourth parameter is a string. The fifth is extra.
PRIVATE _peject, _wrap

*-- Test for no records found
IF EOF() .OR. .NOT. FOUND()
    RETURN
ENDIF

*-- turn word wrap mode off
_wrap=.F.

IF _plength < 1 + 1 + 2
    SET DEVICE TO SCREEN
    DEFINE WINDOW gw_report FROM 7,17 TO 11,62 DOUBLE
    ACTIVATE WINDOW gw_report
    @ 0,1 SAY "Increase the page length for this report."
    @ 2,1 SAY "Press any key ..."
    x=INKEY(0)
    DEACTIVATE WINDOW gw_report
    RELEASE WINDOW gw_report
    RETURN
ENDIF

_plineno=0          && set lines to zero
*-- NOEJECT parameter
IF gl_noject
    IF _peject="BEFORE"
        _peject="NONE"
    ENDIF
    IF _peject="BOTH"
        _peject="AFTER"
    ENDIF
ENDIF
ENDIF

```

dBase IV PROGRAM CODE

```

*-- Set-up environment
ON ESCAPE DO Prnabort
IF SET("TALK")="ON"
    SET TALK OFF
    gc_talk="ON"
ELSE
    gc_talk="OFF"
ENDIF
gc_space=SET("SPACE")
SET SPACE OFF
gc_time=TIME()      && system time for predefined field
gd_date=DATE()      && system date " " " "
gl_fandl=.F.        && first and last page flag
gl_prntflg=.T.      && Continue printing flag
gl_widow=.T.        && flag for checking widow bands
gn_length=LEN(gc_heading) && store length of the HEADING
gn_level=2          && current band being processed
gn_page=_pageno     && grab current page number
gn_pspace=_pspacing && get current print spacing

*-- Set up procedure for page break
gn_atline=_plength - 1
ON PAGE AT LINE gn_atline EJECT PAGE

*-- Print Report

PRINTJOB

IF gl_plain
    ON PAGE AT LINE gn_atline DO Pgplain
ELSE
    ON PAGE AT LINE gn_atline DO Pgfoot
ENDIF

DO Pghead

gl_fandl=.T.        && first physical page started

DO Rintro

*-- File Loop
DO WHILE FOUND() .AND. .NOT. EOF() .AND. gl_prntflg
    gn_level=0
    *-- Detail lines
    IF gl_summary
        DO Upd_Vars

```

dBase IV PROGRAM CODE

```

ELSE
  DO __Detail
ENDIF
gl_widow=.T.          && enable widow checking
CONTINUE
ENDDO

IF gl_prntflg
ELSE
  DO Reset
  RETURN
ENDIF

ON PAGE

ENDPRINTJOB

DO Reset
RETURN
* EOP: CMD_CITY.FRG

*-- Update summary fields and/or calculated fields.
PROCEDURE Upd_Vars
RETURN
* EOP: Upd_Vars

*-- Set flag to get out of DO WHILE loop when escape is
pressed.
PROCEDURE Prnabort
gl_prntflg=.F.
RETURN
* EOP: Prnabort

PROCEDURE Pghead

*-- Print HEADING parameter ie. REPORT FORM <name> HEADING
<expC>
IF .NOT. gl_plain .AND. gn_length > 0
  ?? gc_heading FUNCTION "I;V"+LTRIM(STR(_rmargin-_lmargin))
  ?
ENDIF
RETURN
* EOP: Pghead

PROCEDURE Rintro
?? "NAVY COMMAND BY GEOGRAPHIC LOCATION" STYLE "BU" AT 23
?
```

dBase IV PROGRAM CODE

```

?
?? "CITY:  " STYLE "B" AT 29
?? MCITY
?
?
?? "SEA/" STYLE "B" AT 38
?
?? "UIC" STYLE "B" AT 8,;
"COMMAND" STYLE "B" AT 21,;
"SHORE" STYLE "BU" AT 38,;
"HOMEPORT" STYLE "BU" AT 47,;
"ZIP CODE" STYLE "BU" AT 61
?
RETURN
* EOP: Rintro

PROCEDURE __Detail
IF 2 * gn_pspace < gn_atline - 1
  IF gl_widow .AND. _plineno+2 * gn_pspace > gn_atline + 1
    EJECT PAGE
  ENDIF
ENDIF
DO Upd_Vars
?
?? Uic FUNCTION "T" AT 7,;
Act_title FUNCTION "T" AT 17,;
Sea_shore FUNCTION "T" AT 40,;
Homeport FUNCTION "T" AT 48,;
Zip FUNCTION "T" AT 60,;
"-" ,;
Zip_ext FUNCTION "T"
?
RETURN
* EOP: __Detail

PROCEDURE Pgfoot
PRIVATE _box
gl_widow=.F.          && disable widow checking
EJECT PAGE
*-- is the page number greater than the ending page
IF _pageno > _pepage
  GOTO BOTTOM
  SKIP
  gn_level=0
ENDIF
IF .NOT. gl_plain .AND. gl_fandl

```

dBase IV PROGRAM CODE

```

    _pspacing=gn_pspace
    DO Pghead
ENDIF
RETURN
* EOP: Pgfoot

*-- Process page break when PLAIN option is used.
PROCEDURE Pgplain
PRIVATE _box
EJECT PAGE
RETURN
* EOP: Pgplain

*-- Reset dBASE environment prior to calling report
PROCEDURE Reset
SET SPACE &gc_space.
SET TALK &gc_talk.
ON ESCAPE
ON PAGE
RETURN
* EOP: Reset

* Program.....: CMD_UIC.FRG
* Date.....: 9-10-90
*
* Notes:
* -----
* Prior to running this procedure with the DO command
* it is necessary use LOCATE because the CONTINUE
* statement is in the main loop.
*
*-- Parameters
PARAMETERS gl_noeject, gl_plain, gl_summary, gc_heading,
gc_extra
** The first three parameters are of type Logical.
** The fourth parameter is a string. The fifth is extra.
PRIVATE _peject, _wrap

*-- Test for no records found
IF EOF() .OR. .NOT. FOUND()
    RETURN
ENDIF

*-- turn word wrap mode off
_wrap=.F.

```

dBase IV PROGRAM CODE

```

IF _plength < 1 + 1 + 2
  SET DEVICE TO SCREEN
  DEFINE WINDOW gw_report FROM 7,17 TO 11,62 DOUBLE
  ACTIVATE WINDOW gw_report
  @ 0,1 SAY "Increase the page length for this report."
  @ 2,1 SAY "Press any key ..."
  x=INKEY(0)
  DEACTIVATE WINDOW gw_report
  RELEASE WINDOW gw_report
  RETURN
ENDIF

_plineno=0          && set lines to zero
*-- NOEJECT parameter
IF gl_noeject
  IF _peject="BEFORE"
    _peject="NONE"
  ENDIF
  IF _peject="BOTH"
    _peject="AFTER"
  ENDIF
ENDIF
ENDIF

*-- Set-up environment
ON ESCAPE DO Prnabort
IF SET("TALK")="ON"
  SET TALK OFF
  gc_talk="ON"
ELSE
  gc_talk="OFF"
ENDIF
gc_space=SET("SPACE")
SET SPACE OFF
gc_time=TIME()      && system time for predefined field
gd_date=DATE()      && system date " " " "
gl_fandl=.F.        && first and last page flag
gl_prntflg=.T.      && Continue printing flag
gl_widow=.T.        && flag for checking widow bands
gn_length=LEN(gc_heading) && store length of the HEADING
gn_level=2          && current band being processed
gn_page=_pageno    && grab current page number
gn_pspace=_pspacing && get current print spacing

*-- Set up procedure for page break
gn_atline=_plength - 1
ON PAGE AT LINE gn_atline EJECT PAGE

```

dBase IV PROGRAM CODE

```
*-- Print Report
```

```
PRINTJOB
```

```
IF gl_plain
  ON PAGE AT LINE gn_atline DO Pgplain
ELSE
  ON PAGE AT LINE gn_atline DO Pgfoot
ENDIF
```

```
gl_fandl=.T.          && first physical page started
```

```
*-- File Loop
```

```
DO WHILE FOUND() .AND. .NOT. EOF() .AND. gl_prntflg
  gn_level=0
  *-- Detail lines
  IF gl_summary
    DO Upd_Vars
  ELSE
    DO __Detail
  ENDIF
  gl_widow=.T.          && enable widow checking
  CONTINUE
```

```
ENDDO
```

```
IF gl_prntflg
ELSE
  DO Reset
  RETURN
ENDIF
```

```
ON PAGE
```

```
ENDPRINTJOB
```

```
DO Reset
```

```
RETURN
```

```
* EOP: CMD_UIC.FRG
```

```
*-- Update summary fields and/or calculated fields.
```

```
PROCEDURE Upd_Vars
```

```
RETURN
```

```
* EOP: Upd_Vars
```

```
*-- Set flag to get out of DO WHILE loop when escape is
pressed.
```

```
PROCEDURE Prnabort
```

dBase IV PROGRAM CODE

```
gl_prntflg=.F.
RETURN
* EOP: Prnabort
```

```
PROCEDURE __Detail
IF 13 * gn_pspace < gn_atline - 1
  IF gl_widow .AND. _plinen+13 * gn_pspace > gn_atline + 1
    EJECT PAGE
  ENDIF
ENDIF
DO Upd_Vars
?
?? "NAVY COMMAND BY UIC" STYLE "BU" AT 32
?
?
?
?? Addr1 FUNCTION "T" STYLE "BU" AT 11,;
  "UIC:" STYLE "BU" AT 54,;
  Uic FUNCTION "T" STYLE "BU" AT 60
?
?
?? Addr2 FUNCTION "T" STYLE "BU" AT 11,;
  "HOMEPORT:" STYLE "BU" AT 49,;
  Homeport FUNCTION "T" STYLE "BU" AT 60
?
?
?? Addr3 FUNCTION "T" STYLE "BU" AT 11,;
  "SEA/SHORE:" STYLE "BU" AT 48,;
  Sea_shore FUNCTION "T" STYLE "BU" AT 60
?
?
?? Addr4 FUNCTION "T" STYLE "BU" AT 11,;
  "ZIP-CODE:" STYLE "BU" AT 49,;
  Zip FUNCTION "T" STYLE "BU" AT 60,;
  "-" STYLE "B" ,;
  Zip_ext FUNCTION "T" STYLE "B"
?
?
?
RETURN
* EOP: __Detail
```

```
PROCEDURE Pgfoot
PRIVATE _box
```

dBaSe IV PROGRAM CODE

```

gl_widow=.F.          && disable widow checking
EJECT PAGE
*-- is the page number greater than the ending page
IF _pageno > _pepage
    GOTO BOTTOM
    SKIP
    gn_level=0
ENDIF
RETURN
* EOP: Pgfoot

*-- Process page break when PLAIN option is used.
PROCEDURE Pgplain
PRIVATE _box
EJECT PAGE
RETURN
* EOP: Pgplain

*-- Reset dBASE environment prior to calling report
PROCEDURE Reset
SET SPACE &gc_space.
SET TALK &gc_talk.
ON ESCAPE
ON PAGE
RETURN
* EOP: Reset

* Program.....: LIN_NBR.FRG
* Date.....: 9-10-90
*
* Notes:
* -----
* Prior to running this procedure with the DO command
* it is necessary use LOCATE because the CONTINUE
* statement is in the main loop.
*
*-- Parameters
PARAMETERS gl_noeject, gl_plain, gl_summary, gc_heading,
gc_extra
** The first three parameters are of type Logical.
** The fourth parameter is a string. The fifth is extra.
PRIVATE _pject, _wrap

*-- Test for no records found
IF EOF() .OR. .NOT. FOUND()

```

dBase IV PROGRAM CODE

```

RETURN
ENDIF

*-- turn word wrap mode off
_wrap=.F.

IF _plength < 1 + 1 + 2
  SET DEVICE TO SCREEN
  DEFINE WINDOW gw_report FROM 7,17 TO 11,62 DOUBLE
  ACTIVATE WINDOW gw_report
  @ 0,1 SAY "Increase the page length for this report."
  @ 2,1 SAY "Press any key ..."
  x=INKEY(0)
  DEACTIVATE WINDOW gw_report
  RELEASE WINDOW gw_report
  RETURN
ENDIF

_plineno=0          && set lines to zero
*-- NOEJECT parameter
IF gl_noeject
  IF _peject="BEFORE"
    _peject="NONE"
  ENDIF
  IF _peject="BOTH"
    _peject="AFTER"
  ENDIF
ENDIF
ENDIF

*-- Set-up environment
ON ESCAPE DO Prnabort
IF SET("TALK")="ON"
  SET TALK OFF
  gc_talk="ON"
ELSE
  gc_talk="OFF"
ENDIF
gc_space=SET("SPACE")
SET SPACE OFF
gc_time=TIME()      && system time for predefined field
gd_date=DATE()      && system date " " " "
gl_fandl=.F.        && first and last page flag
gl_prntflg=.T.      && Continue printing flag
gl_widow=.T.        && flag for checking widow bands
gn_length=LEN(gc_heading) && store length of the HEADING
gn_level=2          && current band being processed
gn_page=_pageno    && grab current page number

```

dBase IV PROGRAM CODE

```

gn_pspace=_pspacing && get current print spacing

*-- Set up procedure for page break
gn_atline=_plength - 1
ON PAGE AT LINE gn_atline EJECT PAGE

*-- Print Report

PRINTJOB

IF gl_plain
  ON PAGE AT LINE gn_atline DO Pgplain
ELSE
  ON PAGE AT LINE gn_atline DO Pgfoot
ENDIF

DO Pghead

gl_fandl=.T.          && first physical page started

DO Rintro

*-- File Loop
DO WHILE FOUND() .AND. .NOT. EOF() .AND. gl_prntflg
  gn_level=0
  *-- Detail lines
  IF gl_summary
    DO Upd_Vars
  ELSE
    DO __Detail
  ENDIF
  gl_widow=.T.          && enable widow checking
  CONTINUE
ENDDO

IF gl_prntflg
ELSE
  DO Reset
  RETURN
ENDIF

ON PAGE

ENDPRINTJOB

DO Reset

```

dBase IV PROGRAM CODE

```

RETURN
* EOP: LIN_NBR.FRG

*-- Update summary fields and/or calculated fields.
PROCEDURE Upd_Vars
RETURN
* EOP: Upd_Vars

*-- Set flag to get out of DO WHILE loop when escape is
pressed.
PROCEDURE Prnabort
gl_prntflg=.F.
RETURN
* EOP: Prnabort

PROCEDURE Pghead

*-- Print HEADING parameter ie. REPORT FORM <name> HEADING
<expC>
IF .NOT. gl_plain .AND. gn_length > 0
    ?? gc_heading FUNCTION "I;V"+LTRIM(STR(_rmargin-_lmargin))
    ?
ENDIF
RETURN
* EOP: Pghead

PROCEDURE Rintro
?? "PRECEDENCE LIST REPORT" AT 28
?
?
?? "DESIG-" STYLE "B" AT 58,;
"DATE OF" STYLE "B" AT 67
?
?? "NUMBER" AT 4,;
"NAME" AT 28,;
"RANK" AT 49,;
"NATOR" AT 58,;
"RANK" AT 68,;
"SEX" AT 76
?
RETURN
* EOP: Rintro

PROCEDURE __Detail
IF 2 * gn_pspace < gn_atline - 1
    IF gl_widow .AND. _plineno+2 * gn_pspace > gn_atline + 1
        EJECT PAGE
    
```

dBase IV PROGRAM CODE

```

    ENDIF
ENDIF
DO Upd_Vars
?
?? Lineal_nbr FUNCTION "T" PICTURE "XXXXXXXX-XX" AT 2,;
   Name FUNCTION "T" AT 17,;
   Rank FUNCTION "T" AT 49,;
   Designator FUNCTION "T" AT 58,;
   Dor FUNCTION "T" AT 67,;
   Sex FUNCTION "T" AT 77
?
RETURN
* EOP: __Detail

```

```

PROCEDURE Pgfoot
PRIVATE _box
gl_widow=.F.          && disable widow checking
EJECT PAGE
*-- is the page number greater than the ending page
IF _pageno > _pepage
   GOTO BOTTOM
   SKIP
   gn_level=0
ENDIF
IF .NOT. gl_plain .AND. gl_fandl
   _pspacing=gn_pspace
   DO Pghead
ENDIF
RETURN
* EOP: Pgfoot

```

```

*-- Process page break when PLAIN option is used.
PROCEDURE Pgplain
PRIVATE _box
EJECT PAGE
RETURN
* EOP: Pgplain

```

```

*-- Reset dBASE environment prior to calling report
PROCEDURE Reset
SET SPACE &gc_space.
SET TALK &gc_talk.
ON ESCAPE
ON PAGE
RETURN
* EOP: Reset

```

dBase IV PROGRAM CODE

```

* Program.....: OFFICER.FRG
* Date.....: 9-06-90
*
* Notes:
* -----
* Prior to running this procedure with the DO command
* it is necessary use LOCATE because the CONTINUE
* statement is in the main loop.
*
*-- Parameters
PARAMETERS gl_noeject, gl_plain, gl_summary, gc_heading,
gc_extra
** The first three parameters are of type Logical.
** The fourth parameter is a string. The fifth is extra.
PRIVATE _peject, _wrap

*-- Test for no records found
IF EOF() .OR. .NOT. FOUND()
    RETURN
ENDIF

*-- turn word wrap mode off
_wrap=.F.

IF _plength < 1 + 1 + 2
    SET DEVICE TO SCREEN
    DEFINE WINDOW gw_report FROM 7,17 TO 11,62 DOUBLE
    ACTIVATE WINDOW gw_report
    @ 0,1 SAY "Increase the page length for this report."
    @ 2,1 SAY "Press any key ..."
    x=INKEY(0)
    DEACTIVATE WINDOW gw_report
    RELEASE WINDOW gw_report
    RETURN
ENDIF

_plineno=0          && set lines to zero
*-- NOEJECT parameter
IF gl_noeject
    IF _peject="BEFORE"
        _peject="NONE"
    ENDIF
    IF _peject="BOTH"
        _peject="AFTER"
    ENDIF
ENDIF
ENDIF

```

dBase IV PROGRAM CODE

```

*-- Set-up environment
ON ESCAPE DO Prnabort
IF SET("TALK")="ON"
    SET TALK OFF
    gc_talk="ON"
ELSE
    gc_talk="OFF"
ENDIF
gc_space=SET("SPACE")
SET SPACE OFF
gc_time=TIME()      && system time for predefined field
gd_date=DATE()      && system date " " " "
gl_fandl=.F.        && first and last page flag
gl_prntflg=.T.      && Continue printing flag
gl_widow=.T.        && flag for checking widow bands
gn_length=LEN(gc_heading) && store length of the HEADING
gn_level=2          && current band being processed
gn_page=_pageno     && grab current page number
gn_pspace=_pspacing && get current print spacing

*-- Set up procedure for page break
gn_atline=_plength - 1
ON PAGE AT LINE gn_atline EJECT PAGE

*-- Print Report

PRINTJOB

IF gl_plain
    ON PAGE AT LINE gn_atline DO Pgplain
ELSE
    ON PAGE AT LINE gn_atline DO Pgfoot
ENDIF

gl_fandl=.T.        && first physical page started

*-- File Loop
DO WHILE FOUND() .AND. .NOT. EOF() .AND. gl_prntflg
    gn_level=0
    *-- Detail lines
    IF gl_summary
        DO Upd_Vars
    ELSE
        DO __Detail
    ENDIF
    gl_widow=.T.      && enable widow checking

```

dBase IV PROGRAM CODE

```

    IF dsg=1
      WAIT
    ENDIF
    CONTINUE
ENDDO

IF gl_prntflg
ELSE
  DO Reset
  RETURN
ENDIF

ON PAGE

ENDPRINTJOB

DO Reset
RETURN
* EOP: OFFICER.FRG

*-- Update summary fields and/or calculated fields.
PROCEDURE Upd_Vars
RETURN
* EOP: Upd_Vars

*-- Set flag to get out of DO WHILE loop when escape is
pressed.
PROCEDURE Prnabort
gl_prntflg=.F.
RETURN
* EOP: Prnabort

PROCEDURE __Detail
IF 21 * gn_pspace < gn_atline - 1
  IF gl_widow .AND. _plinen+21 * gn_pspace > gn_atline + 1
    EJECT PAGE
  ENDIF
ENDIF
DO Upd_Vars
?
?
?
?? ;
"=====
=====";

```

dBase IV PROGRAM CODE

```

+ "=====";
AT 0
?
?
?? "Name:" STYLE "B" AT 4,;
Name STYLE "U" AT 11,;
"Rank:" STYLE "B" AT 40,;
Rank FUNCTION "T" STYLE "U" AT 47,;
"Date of Rank:" STYLE "B" AT 53,;
Dor FUNCTION "T" STYLE "U" AT 68
?
?
?? "Sex:" STYLE "B" AT 4,;
Sex STYLE "U" AT 10,;
"Date of Birth:" STYLE "B" AT 13,;
Dob FUNCTION "T" STYLE "B" AT 29,;
"Designator:" STYLE "B" AT 38,;
Designator STYLE "U" AT 51,;
"Year Group:" STYLE "BU" AT 58,;
Year_group STYLE "U" AT 71
?
?
?? "Subspecialties:" STYLE "B" AT 4,;
Specialty1 STYLE "U" AT 23,;
"/" STYLE "BU" ,;
Specialty2 STYLE "U" ,;
"/" STYLE "BU" ,;
Specialty3 STYLE "U" ,;
"Lineal Number:" STYLE "B" AT 47,;
Lineal_nbr STYLE "U" AT 65
?
?
?? "UIC:" STYLE "B" AT 4,;
Uic FUNCTION "T" STYLE "U" AT 12,;
"Command:" STYLE "B" AT 22,;
Act_title FUNCTION "T" STYLE "B" AT 33,;
"Source Code:" STYLE "B" AT 57,;
Source_cod STYLE "U" AT 71
?
?
?? "HOMEPOR:" STYLE "BU" AT 4,;
Homeport FUNCTION "T" STYLE "U" AT 16,;
"Date Reported:" STYLE "B" AT 30,;
Date_rptd FUNCTION "T" STYLE "U" AT 49,;
"PRD:" STYLE "B" AT 61,;
Prd STYLE "U" AT 70
?

```

dBase IV PROGRAM CODE

```

?
?? "BSC:" STYLE "B" AT 4,;
  Bsc STYLE "U" AT 11,;
  "Billet Title:" STYLE "B" AT 21,;
  Billet_ttl STYLE "U" AT 37,;
  "Sea/Shore:" STYLE "B" AT 56,;
  Sea_shore FUNCTION "T" STYLE "B" AT 69
?
?
?? ;
"=====
=====";
+ "=====";
STYLE "U" AT 0
?
?
?
?
RETURN
* EOP: __Detail

PROCEDURE Pgfoot
PRIVATE _box
gl_widow=.F.          && disable widow checking
EJECT PAGE
*-- is the page number greater than the ending page
IF _pageno > _pepage
  GOTO BOTTOM
  SKIP
  gn_level=0
ENDIF
RETURN
* EOP: Pgfoot

*-- Process page break when PLAIN option is used.
PROCEDURE Pgplain
PRIVATE _box
EJECT PAGE
RETURN
* EOP: Pgplain

*-- Reset dBASE environment prior to calling report
PROCEDURE Reset
SET SPACE &gc_space.
SET TALK &gc_talk.
ON ESCAPE

```

dBase IV PROGRAM CODE

```

ON PAGE
RETURN
* EOP: Reset

* Program.....: SPECIALT.FRG
* Date.....: 9-10-90
*
* Notes:
* -----
* Prior to running this procedure with the DO command
* it is necessary use LOCATE because the CONTINUE
* statement is in the main loop.
*
*-- Parameters
PARAMETERS gl_noeject, gl_plain, gl_summary, gc_heading,
gc_extra
** The first three parameters are of type Logical.
** The fourth parameter is a string. The fifth is extra.
PRIVATE _peject, _wrap

*-- Test for no records found
IF EOF() .OR. .NOT. FOUND()
    RETURN
ENDIF

*-- turn word wrap mode off
_wrap=.F.

IF _plength < 1 + 1 + 2
    SET DEVICE TO SCREEN
    DEFINE WINDOW gw_report FROM 7,17 TO 11,62 DOUBLE
    ACTIVATE WINDOW gw_report
    @ 0,1 SAY "Increase the page length for this report."
    @ 2,1 SAY "Press any key ..."
    x=INKEY(0)
    DEACTIVATE WINDOW gw_report
    RELEASE WINDOW gw_report
    RETURN
ENDIF

_plineno=0          && set lines to zero
*-- NOEJECT parameter
IF gl_noeject
    IF _peject="BEFORE"
        _peject="NONE"

```

dBase IV PROGRAM CODE

```

ENDIF
IF _peject="BOTH"
  _peject="AFTER"
ENDIF
ENDIF

*-- Set-up environment
ON ESCAPE DO Prnabort
IF SET("TALK")="ON"
  SET TALK OFF
  gc_talk="ON"
ELSE
  gc_talk="OFF"
ENDIF
gc_space=SET("SPACE")
SET SPACE OFF
gc_time=TIME()      && system time for predefined field
gd_date=DATE()      && system date " " " "
gl_fandl=.F.        && first and last page flag
gl_prntflg=.T.     && Continue printing flag
gl_widow=.T.       && flag for checking widow bands
gn_length=LEN(gc_heading) && store length of the HEADING
gn_level=2         && current band being processed
gn_page=_pageno   && grab current page number
gn_pspace=_pspacing && get current print spacing

*-- Set up procedure for page break
gn_atline=_plength - 1
ON PAGE AT LINE gn_atline EJECT PAGE

*-- Print Report

PRINTJOB

IF gl_plain
  ON PAGE AT LINE gn_atline DO Pgplain
ELSE
  ON PAGE AT LINE gn_atline DO Pgfoot
ENDIF

DO Pghead

gl_fandl=.T.      && first physical page started

DO Rintro

```

dBase IV PROGRAM CODE

```

*-- File Loop
DO WHILE FOUND() .AND. .NOT. EOF() .AND. gl_prntflg
  gn_level=0
  *-- Detail lines
  IF gl_summary
    DO Upd_Vars
  ELSE
    DO __Detail
  ENDIF
  gl_widow=.T.          && enable widow checking
  CONTINUE
ENDDO

IF gl_prntflg
ELSE
  DO Reset
  RETURN
ENDIF

ON PAGE

ENDPRINTJOB

DO Reset
RETURN
* EOP: SPECIALT.FRG

*-- Update summary fields and/or calculated fields.
PROCEDURE Upd_Vars
RETURN
* EOP: Upd_Vars

*-- Set flag to get out of DO WHILE loop when escape is
pressed.
PROCEDURE Prnabort
gl_prntflg=.F.
RETURN
* EOP: Prnabort

PROCEDURE Pghead

*-- Print HEADING parameter ie. REPORT FORM <name> HEADING
<expC>
IF .NOT. gl_plain .AND. gn_length > 0
  ?? gc_heading FUNCTION "I;V"+LTRIM(STR(_rmargin-_lmargin))
  ?
ENDIF

```

dBase IV PROGRAM CODE

```

RETURN
* EOP: Pghead

PROCEDURE Rintro
?? "SUBSPECIALTY CODED OFFICERS" AT 25
?
?
?
?? "SUBSPECIALTIES" STYLE "BU" AT 1,;
  "NAME" STYLE "BU" AT 31,;
  "RANK" STYLE "BU" AT 47,;
  "BSC" STYLE "BU" AT 54,;
  "BILLET TITLE" STYLE "BU" AT 61,;
  "UIC" STYLE "BU" AT 76
?
RETURN
* EOP: Rintro

PROCEDURE __Detail
IF 2 * gn_pspace < gn_atline - 1
  IF gl_widow .AND. _plineno+2 * gn_pspace > gn_atline + 1
    EJECT PAGE
  ENDIF
ENDIF
DO Upd_Vars
?
?? Specialty1 FUNCTION "T" AT 0,;
  "/" STYLE "B" ,;
  Specialty2 FUNCTION "T" STYLE "B" ,;
  "/" STYLE "B" ,;
  Specialty3 FUNCTION "T" STYLE "B" ,;
  Name FUNCTION "T" STYLE "B" AT 19,;
  Rank FUNCTION "T" STYLE "B" AT 47,;
  Bsc2 FUNCTION "T" STYLE "B" AT 53,;
  Billet_ttl FUNCTION "T" STYLE "B" AT 60,;
  Uic FUNCTION "T" STYLE "B" AT 75
?
RETURN
* EOP: __Detail

PROCEDURE Pgfoot
PRIVATE _box
gl_widow=.F.          && disable widow checking
EJECT PAGE
*-- is the page number greater than the ending page
IF _pageno > _pepage

```

dBase IV PROGRAM CODE

```
GOTO BOTTOM
SKIP
gn_level=0
ENDIF
IF .NOT. gl_plain .AND. gl_fandl
    _pspacing=gn_pspace
    DO Pghead
ENDIF
RETURN
* EOP: Pgfoot

*-- Process page break when PLAIN option is used.
PROCEDURE Pgplain
PRIVATE _box
EJECT PAGE
RETURN
* EOP: Pgplain

*-- Reset dBASE environment prior to calling report
PROCEDURE Reset
SET SPACE &gc_space.
SET TALK &gc_talk.
ON ESCAPE
ON PAGE
RETURN
* EOP: Reset
```

LIST OF REFERENCES

1. Lambert, Steve, and Ropiequet, Suzanne, CD ROM THE NEW PAPYRUS, Microsoft Press, p. 58, 1986.
2. CD PUBLISHER USER MANUAL, Meridian Data, Inc., p. 59, 1988.
3. Lind, David, OPTICAL LASER TECHNOLOGY, SPECIFICALLY CD-ROM, AND ITS APPLICATION TO THE STORAGE AND RETRIEVAL OF INFORMATION, NPS Thesis, p. 24, 1987.
4. Telephone conversation conversation between Ms. Gloria Travers, Code 12, Naval Military Personnel Command and the author, 15 June 1990.
5. Telephone conversation conversation between LCDR Craig Pulver, Naval Publications and Forms Center and the author, 15 June 1990.
6. Dolan, Kathleen and Kroenke, David M., DATABASE PROCESSING: FUNDAMENTALS, DESIGN, IMPLEMENTATION, 3rd Edition, Science Research Associates, Inc., 1988.
7. Shields, Jim, "Singing Praises of CD-ROM Technology", GOVERNMENT COMPUTER NEWS, Vol. 9, No. 3, p. 35, February 5, 1990.
8. Hawkins, William J., "Electronic Book", POPULAR SCIENCE, p. 42, September 1990.

BIBLIOGRAPHY

Bonner, Paul, and Jerome, Marty, "CD-ROM Power: Knowledge in Hand", PC COMPUTING, Vol. 3, No. 2, February 1990.

Buddine, Laura and Young, Elizabeth, THE BRADY GUIDE TO CD-ROM, Prentice Hall Press, 1987.

Dolan, Kathleen and Kroenke, David M., DATABASE PROCESSING: FUNDAMENTALS, DESIGN, IMPLEMENTATION, 3rd Edition, Science Research Associates, Inc., 1988.

Hammitt, James E. and Prague, Cary N., dBASE IV PROGRAMMING, Windcrest Books, 1989.

Hawkins, William J., "Electronic Book", POPULAR SCIENCE, p. 42, September 1990.

Lambert, Steve, and Ropiequet, Suzanne, CD ROM THE NEW PAPYRUS, Microsoft Press, 1986.

Lind, David, OPTICAL LASER TECHNOLOGY, SPECIFICALLY CD-ROM, AND ITS APPLICATION TO THE STORAGE AND RETRIEVAL OF INFORMATION, NPS Thesis, p. 24, 1987.

McCormick, John, "CD-ROM SYSTEMS: New Hardware Expands Uses for Disks' Vast Storage", GOVERNMENT COMPUTER NEWS, Vol. 9, No. 9, 30 April 1990.

Meridian Data, Inc., CD PUBLISHER USER MANUAL, Meridian Data, Inc., 1988.

Ropiequet, Suzanne, CD ROM OPTICAL PUBLISHING, Microsoft Press, 1987.

Shields, Jim, "Singing Praises of CD-ROM Technology", GOVERNMENT COMPUTER NEWS, Vol. 9, No. 3, p. 35, February 5, 1990.

Simpson, Alan, dBASE IV PROGRAMMER'S INSTANT REFERENCE, Sybex, Inc., 1989.

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center 2
Cameron Station
Alexandria, Virginia 22304-6145
2. Library, Code 52 2
Naval Postgraduate School
Monterey, California 93943-5002
3. Dr. William J. Haga 1
Naval Postgraduate School
Code AS/HG
Monterey, California 93942
4. Barry A. Frew 6
Administrative Sciences Department
Code 54FW
Naval Postgraduate School
Monterey, California 93943
5. Defense Manpower Data Center 1
99 Pacific St.
Suite 155A
Monterey, California 93940-2453
6. Commander Naval Air Force 2
United States Atlantic Fleet
LT Douglas E. Brinkley
Code 72
Norfolk, Virginia 23511-5188
7. Commander 2
Naval Computer and Telecommunications Command
LT Paul G. Scyoc
4401 Massachusetts Ave., NW
Washington, DC 20394-5000
8. Commanding Officer 1
Naval Military Personnel Command
Dave Skeen (NMPC-16)
Navy Department
Washington, DC 20370
9. Meridian Data, Incorporated 1
Dean Quarnstrom
5615 Scotts Valley Dr.
Scotts Valley, California 95066