1987

A management information system for the 341st Security Police Group

David K. Hazlett
The University of Montana

A MANAGEMENT INFORMATION SYSTEM FOR THE 341ST SECURITY POLICE GROUP

By

David K. Hazlett

B.S., University of Toledo, 1984

Presented in Partial Fulfillment of the
Requirements for the Degree of

Master of Business Administration

UNIVERSITY OF MONTANA

1987

APPROVED BY:

Thomas O. Kirkpatrick
Chairman, Board of Examiners

Dean, Graduate School

August 18, 1987
Date

TABLE OF CONTENTS

Acknowledgements

The researcher expresses his thanks to those who helped make this effort possible. They include Dr. Thomas Kirkpatrick, Dr. Bernard Bowlen, Dr. James Novitzki, and Mr. Marty Melander of the University of Montana; TSgt. Johnny McCarty of the 341st Security Police Group; Mr. Mike Parker of American Telephone Advertising; his wife and daughters; and a special thanks to the typist, Mrs. Becky Hunter.

Chapter I. INTRODUCTION

Like many other business, non-profit, and governmental organizations, the 341st Security Police Group grapples with the technological advances made possible by the microcomputer. Widespread acceptance and integration of the microcomputers has been hampered by poor software development. The researcher resolves some of these problems by analyzing information needs and building a data base management system to serve the Security Police Group.

341st Security Police Group Overview

The 341st Security Police Group is a subunit of the 341st Strategic Missile Wing, located at Malmstrom Air Force Base near Great Falls, Montana. The Security Police Group is composed of just over one thousand persons of various specialties, who are organized into a small "overhead" or staff structure with the bulk of the strength assigned to one of four separate squadrons.

The basic mission of the Security Police Group is to provide security for 200 Minuteman missile launch sites and 20 launch control facilities, and to provide base security and law enforcement services. To accomplish these tasks, four fundamental Air Force Specialties are involved: Security Specialist, Law Enforcement Specialist, Administrative Specialist, and Security Police Officer. Air Force Specialty Codes, explained in detail later, indicate relative skill level within specialties. Examples include, but are not limited to, Instructor, Commander, and Military Working Dog Handler.

The Security Police Group overhead structure consists of the command element and certain functions performed centrally. These include Quality Control, Scheduling, Resource Management, Training, and Reports/Analysis, and others. The Security Police Group overhead staff strength varies due to reassignments, as does the strength of the four squadrons, but averages ninety people.

The four squadrons are arranged along functional lines. The 341st Security Police Squadron (SPS) provides base security and law enforcement services, plus military working dog activities, an investigations section, and a regional confinement facility. Additionally, SPS serves as the "squadron of assignment" for those Security Police Group members serving in the central Security Police Group functions mentioned earlier. SPS strength averages two hundred, excluding those assigned to the group.

The 341st Missile Security Squadron (341 MSS) provides Flight Security Controllers (FSC), Alarm Response Teams (ART), and Security Response Teams (SRT) within the 10th and 490th Strategic Missile Squadrons' flight areas. The 341 MSS strength averages 225.

The 342nd Missile Security Squadron (342 MSS) provides Flight Security Controllers (FSC), Alarm Response Teams (ART), and Security Response Teams (SRT) within the 12th and 564th Strategic Missile Squadrons' flight areas. The 342 MSS strength averages 225.

The 343rd Missile Security Squadron (343 MSS) provides Camper Alert Teams (CAT), Security Escort Teams (SET), Mobile Fire Teams

(MFT), and Convoy Response Forces (CRF) throughout the wing's flight areas. The 343 MSS strength averages three hundred fifty.[1]

Squadron Organization

Each squadron is headed by a commander, who, as a commissioned officer, administers discipline, approves squadron plans and programs, and has ultimate responsibility for the health, morale, and welfare of his members. The commander is assisted by an operations officer, who oversees the day-to-day functioning of the operational staff and line personnel. Also assisting the commander is an executive support officer, who is an administrative specialist overseeing the day-to-day functioning of the administrative staff. Squadron superintendents report to the operations officer, and are senior non-commissioned officers who directly supervise squadron operational staff members, and serve in a consultant and advisor role to the operations officer. Operational line personnel are assigned to flights or crews on rotating schedules and are led by a junior commissioned officer and an experienced non-commissioned officer.[2]

[1] My information here comes from various unit mission statements, and from my personal knowledge of Security Police Group Operations.

[2] This organizational scheme is listed in general terms in Air Force Regulation 125-3, "The Security Police Handbook", and in Security Police Group Regulation 10-1, "Administrative Practices".

Chapter II. THE PROBLEM ENVIRONMENT

Information Processing Within The Security Police Group

The overwhelming majority of recurring information within the Security Police Group is assembled manually, then prepared and edited on a typewriter or if available, a microcomputer in the word processing mode.

Source documents for information requirements and report generation are either Air Force forms designed for general or special purposes, or handwritten notes and lists. Unique or one-time requirements are almost always assembled using handwritten notes and lists. Periodic or recurring requirements may begin using handwritten notes and lists as source documents, but a special purpose form could be developed to streamline the operation.

There is no apparent standard for form design, particularly those forms designed and used at the local level. They may be horizontally or vertically oriented, designed for pen/ink or typewriter preparation, or designed with no apparent forethought. However, some Air Force operations were computerized long ago. They include accounting and finance operations, which use forms designed as source documents for computer input. Such forms have fields and blocks numerically identified and laid out for the data entry person.[3] So,

[3] To illustrate this point, see Appendix 1 for a copy of Air Force Form 2519, "General Purpose Checklist", and conversely, Air Force Form 1548, the Allotment form.

operations historically performed manually, then converted to automatic processing, initially use source documents designed for the old manual system. After a gestation period, source documents better suited to the automated system are developed. It is during the period between automation and new form development that speed and accuracy are most hampered.

Present Computer Resources

Beginning in late 1984, Air Force units in all specialties began receiving their first increment of microcomputers. The primary rationale for the microcomputer purchase was to increase the efficiency of individual units and sections. These first micros, some of which are still in use, represented the first real effort to place computing power in hands of non-programmers.

The first micro was the Zenith Z-100, which is based on an earlier Heathkit design. Marketed by Zenith Data Systems of St. Joseph, Michigan, the Z-100 was delivered with full technical documentation, the Z-DOS operating system, and applications software such as dBase II and WordStar. Most Z-100s were configured with a single floppy disk drive and a 10-megabyte internal hard disk. The only common peripheral was one of several types of daisy-wheel printers. The Z-100 is based on an Intel 8088 microprocessor.[4]

When delivery of the Z-100s became a certainty, end users were

[4] Z100 Users Manual, Zenith Data Systems, St. Joseph, Michigan, 1983, p. E1.

not pre-identified nor were training needs determined. After the computers were installed, a correspondence course was offered to users but not required. Operators generally taught themselves how to use the micros, or turned to computer hobbyists within their office or section for help. Continuing mission requirements simply did not allow operators large amounts of time to orient themselves to the Z-100s. The result for most units was that the microcomputers came to be used as "expensive typewriters," taking advantage chiefly of the word processing function to produce correspondence or for list generation.[5]

Without central guidance from higher headquarters on computer utilization, some units at each base tried to find optimum uses for their micros. In some cases, several bases saw an application and experimented with programming that either was not compatible with or duplicated efforts of other units. For example, the Security Police Group developed programs that automatically set up common forms on word processing software, and also back up recurring training requirements on critical skills on disk as well as on paper.

In late 1986, the replacements for the Z-100s, the Zenith Z-248, began to arrive at the unit level. An Intel 80286-based machine, the Z-248 comes in two basic configurations, the intermediate and

[5] One reason for units' failure to take full advantage of their computing power is likely difficulties encountered with dBase II. dBase II is criticized as having ". . . established (the) standard for cryptic documentation and unresponsive user interface design . . ." Alfred Poor, "Data Base Power Puts on an Easy Interface", PC Magazine, Vol. No. 6, #2, 1-27-87, pp. 110-122.

advanced.[6] The Z-248s are planned to either replace the Z-100s outright or when they wear out. At the present time, the Security Police Group overhead conversion to the Z-248 is over two-thirds complete, and all squadrons within the group have at least one of the six Z-248s they will eventually receive.[7] This is important in light of the fact that only the Z-248s are supplied with dBase III software, which is what the proposed data base management system is designed to use.

Security Police Automated System

Within the Security Police career field, leaders at major command and Air Force Office of Security Police levels recognized the potential of the microcomputers. They commissioned the authorship of a bundled software package named the Security Police Automated System, more commonly known by the acronym SPAS. Nine months after the first Z-100s were in place, SPAS software arrived at the unit level, accompanied by system documentation.[8]

Problems surfaced quickly. The software did not function as outlined in the documentation. The malfunctions had several causes,

[6] See appendix 2, Zenith Z-248 Configurations, for complete nomenclatures on each of the two Z-248 versions.

[7] See appendix 3, Microcomputer Assignment List, for a detailed listing of which Security Police Group offices and functional areas will receive Z-100s and which will receive the Z-248s.

[8] A complete SPAS overview requires a reading of Air Force Manual 171-350, titled "SPAS User's Manual".

including software design, organizational incompatibility, and lack of user experience.

SPAS software design was built around a security police operation based on a single squadron with approximately three hundred personnel. Key to the design is a host/remote relationship, in which one machine is designated as a host for certain file update functions, and others are designated as remotes. At remote machines, section-level file updates are performed, and updated files are provided to the host for host-updated files and reports. These file updates are provided through what is whimsically called an "air gap interface;" that is, the diskettes are hand carried to their destination.

This relationship works well with the single-squadron application, but unique problems surface with a multiple-squadron application such as the 341st Security Police Group. Many main functions handled by SPAS are performed at the Security Police Group overhead level, such as Training, Quality Control, and Manpower Management. This necessitates that Security Police Group assign a micro as a host. However, to perform certain file updates, each squadron must also designate one of its micros as a host. Unfortunately, SPAS programming has no provision for a host to update another host, let alone five hosts updating each other.

Another programming problem unique to a group application is the duplication of certain files among squadrons, for example, files with personnel information in them. SPAS programming, however, permits a host to accept updates from only one remote designated responsible for certain file updates.

Additional problems of lesser magnitude include Social Security Account Number fields in the data base files designated as numeric rather than character fields, many report inputs and outputs being in coded form requiring translation, severe limitations on personnel office symbol designation, and standard report formats that include superfluous information.[9]

Designating the Social Security Account Number (SSAN) as a numeric rather than a character field is a problem in light of dBase III system specifications and general data base design conventions. With dBase III, and most other commercial data base software, character fields lend themselves more easily to manipulative operations such as "sort," "list," and "index" than do numeric fields. Generally, numeric fields should be reserved for data that is to be treated as numeric data. That is, data that is to be summed, averaged, or added or subtracted. Clearly, Social Security Account Numbers will not be added, averaged, or subjected to arithmetic operations. Therefore, a numeric field designation is clearly inappropriate for the SSAN.

A good example of coded input and output in SPAS is the use of duty titles. In the security police career field, there are scores of possible duty titles. Rather than create a character field large

[9] I was aware that the SPAS timetable had not been followed, and that field units had problems implementing it. However, the precise nature of the problems, and their magnitude, weren't clear to me until I was able to interview TSgt. Johnny McCarty, non-commissioned officer in charge of information systems for the 341st Security Police Group.

enough to accomodate actual duty titles, SPAS encodes them with cryptic three character codes that require reference to a chart in order to decode. Data entry clerks would become familiar with the coding fairly quickly. However, those who must input source documents or interpret the coded finished product would surely be frustrated by the coded data. This is an example of user unfriendliness that further prevents SPAS implementation.

SPAS calls for superfluous data, both in the input and output of data. For example, weapon qualification data includes number of rounds fired. In a personnel data base management system, the number of rounds fired is of little consequence, and is data that would not be used by any supervisor or manager. Providing for it causes extra and unnecessary effort for data reporting and entry persons.

SPAS problems have resulted in a perceived lack of faith in the computer as a useful tool. Clearly, there is a need for a system to provide Security Police Group managers with real-time information to support their decision making.

Data Environment Problems

Fragmented Application Development

The problems with fragmented application development are numerous and perhaps obvious. First, different organizations may be expending significant amounts of time and effort duplicating each others' software development. This is the case within the Security Police career field and its attempt at integration of the microcomputer over the last several years. Second, necessary applications may be overlooked without the presence of some agency responsible for central oversight of computer operations. Simpler or popular applications will be over-represented at the unit level, whereas difficult or less popular applications will be neglected. Finally, this method is inefficient. Ideally, different programming tasks could be assigned by a central authority to different units. When all tasks are completed, units would share software with all other participating units.

Inadequate User Service Levels

This problem is best illustrated by example. Within the SPG, the potential of the microcomputer is sorely underutilized, as pointed out earlier. Despite the fact that spreadsheet and data base software are provided with the machines, the computers are used primarily for word processing. This leaves senior managers performing many functions manually that could be automated. If microcomputer users only took advantage of existing commercial software, service levels could rise demonstrably.

Uncontrolled Data Redundancy

With most information processed manually, paper redundancy runs rampant. This problem was recently addressed by the U. S.

Navy, which is making a special effort to automate information storage and retrieval. The Navy estimated that the paper documentation required to operate a new ship weighs 60,000 pounds; weight that could be used for supplies, fuel, or ordnance. Even in a land environment, the time spent to copy information and search for it in manual files is getting more burdensome.[10]

Conventional Data Processing Shortcomings in DBMS

Can the data processing standards and fundamentals developed during the period between World War II and the advent of the microcomputer in the late 1970's be applied to solve the aforementioned data environment problems? Probably not, for several reasons.

First, ad hoc requests are difficult, if not impossible to deal with. This is caused by an ever-increasing number of data storage formats, which in turn leads to losing data too soon due to "housecleaning," and poor timing. The problem of data storage formats is particularly acute. The problems of filing paper manually are self evident, but difficulty in transferring electronic data from tape to disk, or from one type disk to another are encountered

[10] Gene Altshuler, James Fry, Gene Lowenthal, et. al., Practical Data Base Management, ed. Auerbach editorial staff, (Reston, Virginia: Reston Publishing Company, Inc., 1981): p. 3.

continuously.[11]

Second, associating data stored in different files but pertaining to the same person/activity is difficult. This is primarily due to incompatibility and inability to access records, which are not constructed with matching key fields.

Third, the inability to integrate systems significantly increases costs due to structural redundancy. Storing the same data in a number of files increases entry time, increases required storage space, and makes updates very time consuming and prone to error.

Finally, system enhancement requests often affect not only files, but file structures as well. Unfortunately, the file structures are often directly linked to the application programs, so the most minor required enhancement may render an entire system worthless.[12]

[11] A good example of this particular problem is one faced by American Telephone Advertising, a business which places advertising on pay phones in eighteen states. ATA gets its data from the regional phone companies in three different formats. Consequently, the company has to use a "Bernoulli Box," an "Omega Cartridge Disk System," and a "Catamount PCT.9" device, which takes ASC II files directly from tape, in order to perform system updates. Having personally observed ATA's system, it is the impression of the researcher that the lack of standardization among the Bell regionals has created extra expense and inconvenience for independent businesses like ATA. Michael Parker, president of ATA, was interviewed in his office in Great Falls in March, 1987.

[12] Cyrill H. P. Brooks, et. al., Information Systems Design, (Sydney, Prentice Hall of Australia, 1982), pp 185-186.

Solution: The Relational Data Base Management System

The relational data base management system solves the data environment problems mentioned previously. At the same time, it manages to sidestep the shortcomings of conventional data processing applications developed for mainframe computers.

As opposed to the hierarchical or network model, the relational data base model offers simplicity as its main advantage. With modern commercial data base management systems, such as dBase III Plus, designed to run on micros with as little as 256K RAM, simplicity is a necessity in order to conserve limited memory.

The relational model is based on fixed-length records that contain numerous data fields, one of which is designated a "key" field. This key field will be used to establish the "relationship" between the data base files. Conceptually, the relational data base is arranged in rows and columns. The rows correspond to a specific record, and the columns correspond to specific data types. This is similar in concept to functions performed in matrix algebra.[13]

In linear equation systems, a matrix approach can be used to arrive at solutions. Equations are arranged in rows horizontally, with columnar sets of numbers that form a vertical pattern. These are comparable to fields and data types in the relational data base. Variables, normally indicated in an algebraic statement as a letter, are replaced by an integer that represents a coefficient of that

[13] Brooks, et. al., Information Systems Design, p. 201.

variable. The integers, in turn, are comparable to data entered in fields in the relational data base. Once integers are inserted, one would perform a series of manual computations on the matrix to arrive at solutions. Such operations would include interchanging two rows, multiplying elements in a row by some number, or by adding elements from one row to elements in another row. In the relational data base, the computer searches and solves matrix problems at a very high rate of speed.[14]

Having chosen a relational data base, there is another choice to be made. Many excellent commercial data base software packages are currently marketed. The system designer would ordinarily have a difficult time selecting the best one. When designing a system for an Air Force application, however, that decision has already been made. The Air Force has centrally selected Ashton-Tate's dBase versions as its data base software system.

[14] In developing a comprehensive theory for matrices, a number of basic algebraic properties of the matrix must be examined and combined. See Earl W. Swokowski, Fundamentals of College Algebra (Boston: Prindle, Weber, and Schmidt, 1981), PP. 253-268.

Chapter III. ESTABLISHING A DATA BASE MANAGEMENT SYSTEM

Tasks to Develop the Data Base Environment

The first task to develop the data base environment is goal definition. Officially, microcomputer goals within the Security Police career field are to bring SPAS on line at all fields units with an optimistic timetable. The problem with this is twofold. First, the problems with SPAS and its implementation at field units have been pointed out. As configured, it is highly unlikely that SPAS can ever be operational within the Security Police career field. Second, every field unit, including the Security Police Group, has general goals at every organizational level that reflect local constraints, mission requirements, resources, and priorities. The goal statement for microcomputer software implementation does not consider these local conditions as a matter of policy, and if they are compatible, it is only by coincidience. Generally, any goal statement about the data base environment has to be based on the goals of the organization that will live within that environment.

The second task is to establish the Data Base Administration function. The Data Base Administrator, or DBA, serves as the coordination and control agency for the data base management system. In this role, the Data Base Administrator establishes standards and procedures, analyzes requirements, and in the application of this paper, performs file updates and generates reports. In the case of the Security Police Group, there is a senior non-commissioned officer with significant computer expertise assigned primary responsibility for microcomputer utilization within Security Police Group. That office would be the logical home of the Data Base Administrator function. At least in the initial phases of implementation, extra persons should be temporarily assigned to the Data Base Administrator function for file creation, data entry, and other startup operations.

Third, an operating plan should be defined. The plan addresses not only the tasks, but the phasing required to bring the system on-line smoothly. Due to research limitations, the researcher's operating plan for a data base management system will stop short of program line coding and system testing.[15]

The fourth step is to develop data collection standards. Essentially, this means designing the system to incorporate not only the present, but expected future information needs as well. This can be a delicate balancing act; as explained in the subsection "Designing the System's Scope," there is a tendency to expand the scope of the system to the point where it simply does not work. Intestinal fortitude is required on the part of a Data Base Administrator to sort through the myriad of data and determine which will and which will

[15] The major problems with such a plan are political. In the case of the Security Police Group, an operating committee would probably have to be formed, with representatives from all functional areas affected, to include each squadron. Conflicts and disagreements would have to be resolved initially and along the way. Needless to say, this is a time consuming process. For such a project, a suitable tracking method may be Gantt charting, or even PERT/CPM. A good explanation of PERT/CPM can be found in Levin, Rubin, and Stinson's "Quantitative Approaches to Management", Chapter 12.

not be included in the system. By use of the file definitions referred to below, a limited number of data categories which can serve present information needs have been selected. More importantly, additional relationships can be defined using the same data, and new reporting needs can be filled.

The Data Base Administrator ordinarily develops the data dictionary, the fifth step. The data dictionary is a list of the names and types of all data items in the data base system. In the file design section, used as an example, the names and types are listed, and in Appendix #6, an explanation of the meaning of each data type is given. This sounds self-evident, however, failure to provide the data dictionary can lead to confusion among various users and suppliers of source data.

What data the system will need in order to operate is the sixth question that must be addressed. Properly done, this will limit inputs to the Data Base Administrator function to those types of data needed to create or update files within the system. Frequently, initial applications will consist of only a subset of the total possible data environment. This is the case with this analysis, as explained more fully in "Designing the System's Scope." The Data Base Administrator must insure that system inputs apply only to that particular subset. Looking to the future, planners must realize that how they develop and implement the system at this stage will determine to a greater or lesser degree how system modifications and improvements may be made later.

Based on analysis of the organization's data environment, the conceptual data base design takes place. In the seventh step, data are combined into logical groupings, and these groupings merged to form a general data base structure. The Security Police Group system file design and layout result from this step.

Finally, transaction processing must be addressed. How will users or offices responsible to generate system updates get the information to the system itself? Standard input forms have been designed, which are described in the section "Input Forms." These forms are provided as source documents to those offices that will perform system inputs. The system updates themselves will be performed by the Data Base Administrator's data entry function.[16]

Many of the tasks to develop the data base environment depend on a common element. Goal definition, the operating plan, data collection standards, and particularly system data needs require that planners know what information managers want the system to provide to them. The managers of the Security Police Group were surveyed by mail in March 1987. The survey asked managers what information requirements they tasked others for, and what they themselves were tasked to provide. Given an explanation of a data base management system, they were then asked what they thought such a system could do for them. Appendix 4 is an example of the survey form. Appendix 5 is a summary of survey results.

[16] Auerbach, Practical Data Base Management, pp. 20-23.

System Planning

Through system planning, the Data Base Administrator can gain several important benefits. The first of these is the ability to rapidly identify, diagnose, and correct errors. Errors in this case are not coding mistakes and other startup problems, but rather the factual errors and other data entry problems that are ongoing concerns with a system such as this. An excellent way to track these errors is having the source provide both old and new data to the system data entry function. Additional information about audit trails will be found in the subsection "Data Base Audit Trails."

Another important benefit acquired through system planning is cost control. Certainly, the cost of implementing, fine-tuning, and maintaining a data base management system is high. Proper, thorough planning can minimize these costs. Expensive SPAS revisions and corrections had to be accomplished after documentation was printed. Had system planning been effective, patchwork fixes would have been less necessary.

System planning can also provide a fall back position if the system fails to operate as expected. This fall back position can take any of several forms, including a different automated system or a manual system that may have been in place before. On a lesser scale, system planning can make debugging easier, and assist the Data Base Administrator function in modifying or fine-tuning the operation.[17]

[17] Brooks, et. al., Information Systems Design, P. 116.

Designing the System's Scope

Establishing the scope of a data base management system can be, and is quite often a complicated task. Organizations that have invested significant financial and human capital in computer hardware and software want to maximize the benefit of the investment. Frequently, this takes the form of attempting to force system software to be all things to all people. On the lower level, organizational politics can surface, exerting pressure for inclusion or deletion of selected functions. To be effective, the Data Base Administrator should have ultimate veto power over system functions. The Data Base Administrator should keep a handle on the system's scope for several reasons.

First, there is, as mentioned previously, a tendency to continuously widen the system parameters to include functions initially excluded or not considered. This tendency should be resisted. A better approach is to decouple systems so they can operate independently, but exchange information when necessary. So, rather than an unwieldy system that includes every possible function within an organization, designers would be well advised to design and perfect smaller systems. In turn, networking could enable these smaller systems to exchange information with each other. From a system maintenance point of view, decoupled systems are easier to update and troubleshoot, since problems and errors are localized. Again, it is the responsibility of the Data Base Administrator to monitor and enforce these decoupled systems' "boundaries" to insure they do not violate the intent of decoupling in the first place. Without continuous oversight, users who update decoupled systems may be tempted to add functions, data types, and parameters. The net result is an unwieldy group of systems, which, it was hoped, could be avoided by decoupling.

Another reason for the Data Base Administrator to tightly control the system's scope is so that the system can be up and running within a reasonable time. The lag between the time the hardware arrives and the software is debugged and running should be reduced to the absolute minimum. There are two reasons for this, economics and credibility. Obviously, the sooner a system is operational, the sooner both the hardware and the software can generate a return on the organization's investment. Keeping the development focused will help minimize this time. Further, there are credibility considerations to be taken into account. Long periods of time between hardware acquisition and software utilization tend to erode confidence in both the hardware and software. A computer without software is useless, and software that does not work as promised has the same effect. This damage can be repaired over time, but its effects are seldom negligible. The Security Police Group experience with SPAS, which still is not running after two years and two versions, underscores the importance of credibility. Many potential users are looking to the release of the next version of SPAS with jaundiced eye. As with economic concerns, to the extent that limiting system scope accelerates system deadlines, loss of credibility will be a less likely occurrence.

One could legitimately ask how to reconcile the apparent conflict between limiting system scope and insuring that the system can grow to adapt to future needs. System design is a partial answer. In initial stages, designers must be careful not to render their system obsolete as a result of a minor change. A certain amount of forecasting is also required. Designers must be aware of not only future information needs, but of what equipment and peripherals are planned or in development. For example, in the Security Police Group system, the initial operation will be based on a stand-alone micro providing information through hard-copy printouts. This suggests that no password-protect feature is necessary. However, the Air Force may procure local area network (LAN) technology in the not too distant future. Anticipating this, password-protect features should be included in the initial program code.

Faced with pragmatic demands to limit system scope, certain decisions were made with regard to the scope. First, it was decided to make the system primarily personnel-oriented. Consequently, functions with no clear link to personnel matters were eliminated. As one can see by referring to the survey results, many senior managers within the Security Police Group indicated information needs that would clearly be served by a data base management system. However, the system to serve those other needs falls outside the personnel "umbrella", and they had to be ignored. Also, the scope was limited not only horizontally, but vertically as well. For example, Quality Control (QC) data are included, but only down to a certain level. An evaluation consists of three portions weighted for an overall score. It was determined that the overall score was useful to managers, but that the scores for each of the three portions was not. Averages are computed based only on the overall score, as are pass/fail rates, performance reports, and so on. Therefore, the system will not be extended downward to include portions scores. If at some later date, portion score reporting becomes a priority, these data items could be easily added to the appropriate file by adding fields to each record.[18]

System Transparency

In a data base management system, an important consideration is how to treat the relationship between the system user or operator and the system itself. Some important questions include whether to allow operators/users to perform system updates, and, whether to use menu systems or dBase dot commands to access the system.

Generally, to the user/operator, the system should be transparent. By transparent, it is meant that the user uses the system without being aware that he is using a data base management system, or which system he is using. In essence, the software itself is invisible on the monitor. Data independence involves separating the program code itself from the data files that are manipulated by the program code. There are several benefits to transparency and data independence. First, the transparent system, being menu driven, is easier to use.

[18] Ibid., pp. 91-92.

Rather than requiring memorization of system code to access the system, a user need only select one of several alphabetic menu items. Second, the menu driven transparent system is less prone to mistakes. Entering code opens up the possibility of errors that can delay or prevent system access. Third, transparency/data independence preserves the system. Allowing users to access the system via dBase dot commands would invite problems. With an errant keystroke, an unsuspecting user could severely damage program code or file structures.[19]

With these considerations in mind, the Security Police Group system is designed to be user-transparent by a menu-driven method of access. Organizationally, users may only access a certain number of menu functions, with the option to get information for an individual, a duty section, or a squadron. The only other option available for the user is whether or not to print the results. Data entry and system updates are to be performed by the Data Base Administrator function.[20]

By following these guidelines, it is more likely that the Security Police Group system will maintain data independence, and minimize the chances that system code or program files will be damaged by user error.

[19] Auerbach, Practical Data Base Management, p. 121.

[20] Brooks, et. al., Information Systems Design, p. 276.

Designing Files

Data Relationships

Designing files involves determining the relationships between different data elements. Generally, data falls into one of three relationship categories: one-to-one, one-to-many, or many-to-many.[21]

One-to-one relationships are typified by cases in which one data type can be associated with only one other data type. For example, in the Security Police Group system, there can be only one "date arrived station" for each "Social Security Account Number." Generally, data with one-to-one relationships are grouped together in the same file.

One-to-many relationships, conversely, are typified by cases in which one data type could be associated with a number of other data types. A good example of a one-to-many relationship in the Security Police Group system would be the fact that one "Trainer Social Security Account Number" could have a number of "Social Security Account Number" associations, depending on how many trainees the trainer is responsible for. Accordingly, if two data types are related one-to-many, they should be placed in separate files, with the key field of the "one" in the file of the "many." To accomplish

[21] Judd Robbins, Ken Braly; "Data Base By Design," PC World, April 1986, pp. 202-215.

this, separate files contain such information as "Social Security Account Number" and "Mobility Vehicle Type."

Many-to-many relationships occur when a number of data types are associated with a number of other data types. In the Security Police Group system, the only such relationship is the one between many persons (SSAN) and many individual supply items. With many-to-many relationships, the rule of thumb is to build a file around each data type, then to build a third file that contains the key field of each.[22]

The Files

After determining the required data relationships, it was then possible to set up the files for use in the Security Police Group system. Just to manage the personnel-related information, eleven files were required. The dBase III Plus structure display for each file has been reproduced as appendix 6. Following is a brief explanation of each file and the data types stored within. Unless otherwise specified, Social Security Account Number is the key field in each file.

The first file, named "Personnel" (dBase name pers.dbf), contains identification, biographical data, and some logical fields that point

[22] I received other guidance on file design and components of logical design relationships from D. R. Jeffery and M. J. Lawrence, Systems Analysis and Design; (Sydney; Prentice Hall of Australia, 1984); pp. 134-135, and Auerbach, Practical Data Base Management, pp. 274-275.

to other files. Pers.dbf contains 25 fields or data types, such as "last name," "last APR date," and "current grade."

The second file is named "Mobility" (dBase name mob_task.dbf), and contains both individual identification data and unit tasking codes. Mob_task.dbf contains five fields, such as "unit tasking code," and "Air Force specialty code."

The third file is named "Mobility Vehicles" (dBase name mob_veh .dbf). This file contains data on which mobility vehicle types a person is trained to operate. Mob_veh.dbf contains just two fields, "Social Security Account Number" and "vehicle type."

Fourth is a file·named "Mobility Weapons" (dBase name mob_wpn .dbf). Similar to mob_veh.dbf, this file contains data on which mobility weapons a person is qualified to arm himself with. Mob_ wpn.dbf also contains two fields, "Social Security Account Number," and "weapon type."

The fifth file is named "Supply Inventory" (dBase name supply .dbf), and contains data on different supply items in the Security Police Group inventory, including stock number, description, and item cost. In supply.dbf, the key field is national stock number (NSN). There are three fields, "NSN," "description," and "item cost." This file represents part of the single many-to-many relationship occurring in the Security Police Group system. Supply.dbf will be associated with pers.dbf using a linking file, described next.

The sixth file is the linking file between pers.dbf and supply .dbf. It has been named "Individual Supply" (dBase name ind_supp .dbf). It contians the two key fields from the other two files, "SSAN" and "NSN." Ind_supp.dbf contains just the two files mentioned.

The seventh file is called "Upgrade Training" (dBase name ugt .dbf). It contains information on persons who are involved in upgrade training. Ugt.dbf has four fields, like "date entered training" and "trainer's SSAN."

Eighth is "Weapon Qualification" (dBase name qual_wpn.dbf). This file contains data about a person's weapon qualification. Qual_wpn.dbf consists of six fields, such as "serial number" and "course of fire."

The ninth file is named "Special Experience Identifier" (dBase name sei.dbf), and contains information about special experience identifiers a person has been awarded. Sei.dbf contains just two fields, "Social Security Account Number" and "SEI number."

The tenth file is named "Quality Control Data" (dBase name qc_data.dbf). It contains five fields with information about a person's quality control evaluations, such as "duty position" and "score."

The last file is named "Appointment Dates" (dBase name appt_dt.dbf). It contains data about various appointments an individual attended. Appt_dt.dbf contains three fields, such as "type appointment" and "last date."[23]

[23] The file design structures themselves, including all fields, their data types, and lengths as generated by the "display structure" command in dBase, are located in appendix 6.

Program Coding

With file structures defined and relationships established, the next logical step is to physically construct the program code lines.[24] The programmer would be well-advised to begin with a standard menu-driven application generated by dBase III Plus itself.[25] Returning to the basic menu system, the programmer simply inserts lines to permit the system to perform functions such as: expand the menu from the provided five to the required sixteen menu items; provide password protection; include sub-loops and secondary menus; design custom screen displays; design and print custom reports; and accomodate network arrangements.[26]

[24] As it is important to limit the scope of a data base management system, the researcher also had to limit the scope of this project by leaving the actual program code to another researcher's project. Fortunately, dBase III Plus makes this easy with its built in applications generator. At a keystroke, dBase will automatically write a menu-driven data base management system. Albeit simple, the automatic application can be easily customized.

[25] See Appendix #7 for a menu driven data base management system as created by the dBase III Plus applications generator.

[26] Although the physical program code will not be written as part of this research, Appendix #8 provides an outline of the logical arrangement of the system code functions and subprograms, in English.

System Testing

Top-Down vs. Bottom-Up

That newly-written programs must be tested to insure they are error-free and valid goes without saying. There are, however, two basic methods of doing so: top-down and bottom-up.

To illustrate, one must first consider a program as a tree through which a programmer must travel limb by limb to assure its soundness. Top-down testing, as its name suggests, involves travel through the program from the top to the bottom. If subprograms are not yet in existence, dummys may be substituted for them to facilitate the top-down pass. As the actual subprograms are completed, they replace the dummys. Only when all subprograms are in place can the program be definitively tested. This process of substituting dummys can be expensive, but is offset by the advantage of testing programs in a parallel fashion with their implementation, since many programs are written in a top-down method.

Bottom-up testing, by comparison, takes subprograms or modules which are complete, and plugs them into a test bed that insures their validity before inclusion into the overall program. So, the programmers travel through the program tree is from the bottom to the top. Using an actual subprogram, rather than a dummy, bottom-up testing permits more rigorous testing under actual as opposed to simulated environments than does top-down testing.[27]

[27] Brooks, et. al., Information Systems Design, p. 367.

Both methods have advantages and disadvantages. Ultimately, the research believes that the programmer should select the method that best corresponds with the way the program is written. In this case, logical program construction would involve writing code for the individual menu subprograms first, then proceeding to preliminary and closing command lines. This scheme calls for bottom-up testing.

Bottom-up testing for this system would require that a minimum of commands be added to each subprogram. At the beginning, SET and DO commands would be placed, and at the end, ENDDO and RETURN commands. By doing so, each subprogram could be tested under actual conditions.

Debugging Techniques

Assuming that programming errors will be made, debugging techniques utilizing dBase III Plus commands should be considered.[28]

First, keeping the user in mind, dBase stops running a program when an error is detected, and gives the user a choice of three avenues; to cancel the program, ignore the bad line, or suspend the program temporarily. Suspending the program enables the user to enter any dot command, and to resume the program with a "resume" dot command. Additionally, a program may be stopped at any time with the "escape" key, and the same avenue choices are displayed.

[28] The researcher is quick to point out that as a non-programmer, he is not familiar with more sophisticated debugging procedures and techniques, and must leave the individual programmer to his personal training, education, and wiles.

At the dot prompt on a suspend operation, display commands such as "memory," "structure," "status," or "history" are sometimes useful. Certain set options also assist programmers in debugging. Setting the "talk" on will display event messages as they occur in the command file. Setting the "echo" on displays every command line in the program as it is being processed. This happens quickly, but can be slowed by setting "step" on, which will run the program one line at a time. By setting "debug" on, "echo" and "step" outputs are sent directly to the printer for a hard copy of the events that occur in the program.

Of course, the entire program can be "hard-copied," line by line, with the "type...to print" command.[29]

Other Issues

This examination of data base management systems, and the construction of a system model for the Security Police Group is not complete until other miscellaneous topics are explored and resolved. These topics are, in order, data base integrity, privacy considerations, input forms, data base audit trails, and data base management system costs versus benefits.

[29] Alan Simpson, Understanding dBase III Plus, (San Francisco: Sybex), 1986, pp. 272-282.

Data Base Integrity

Many things can jeopardize the integrity of a system. A partial list includes multiple updates on the same record, hardware/software faults, and human error. Through concurrency control and adequate recovery methods, the potential effects of these threats will be minimized.

Concurrent, or simultaneous, updates to the same record will destroy data reliability, and in turn, system integrity. The problem is accentuated when the concurrent update involves numeric data, the same data, or both. Concurrent updates are a problem chiefly with terminal based or LAN capable systems. Since the Security Police Group system is designed initially for stand-alone microcomputers, the initial temptation is to disregard measures to counteract concurrent updates. To do so would be shortsighted, however, in light of the pending USAF procurement of LANs for its microcomputers.

Commercial network systems and LANs typically use serial processing, record locking, or some other method to prevent concurrent updates. It is both efficient and effective to use a simpler, more direct method. Initially, lack of network capability will compel the Security Police Group to process information requests and perform record updates and system maintenance through the Data Base Administrator function. When network capability becomes available, users will be able to request information and generate reports through their terminals, however, it would be prudent to retain the record update and system maintenance functions within the Data Base Administrator.

By centralizing this operation, the problem of concurrent updates would be effectively eliminated. dBase III Plus is capable of accomodating network or Local Area Network arrangements; the specific coding requirements are outlined under the subsection "Privacy Considerations."[30]

Perhaps a more common problem is partial or total system loss caused by faults or errors. Some faults and errors can be minimized by hardware design, personnel training, or other precautions. However, inadvertant losses can occur due to fire, lightning, sabotage, or other natural or man-made disasters. The Data Base Administrator function must anticipate the possibility that losses can occur and minimize their effect, chiefly through the use of backup operations. There are no hard fast rules on how often to backup files, but some general guidelines do apply. Merely creating backup files on the same diskette or hard drive will be of little use if a crash occurs. Therefore, backup files should be created on removable storage devices and stored away from the computer itself. With separate backup files, system loss can be counteracted by relaoding files and records from the backups. Reprocessing of data will be necessary only for those transactions that occurred since the last backup. The Data Base Administrator must establish a schedule for record and file backups, based on the level of system activity. An example of a backup schedule would be to perform daily backups that are consolidated weekly, with

[30] Brooks, et. al., Information Systems Design, p. 218.

weekly consolidations, in turn, consolidated on a monthly basis.[31]

Privacy Considerations

Safeguarding and restricting access to personal data has become more important, particularly since the passage of the Privacy Act in 1974. At the same time, this privacy has become more difficult to ensure, largely as a result of the growth in electronic data processing over the same period. Clearly, the Data Base Administrator is operationally responsible for ensuring that measures are taken to provide data security, program controls, and operational controls.

The first step in insuring data security is a risk assessment. The Data Base Administrator must value information assets under his control in order to prioritize them for safeguarding purposes. Next, a threat analysis must be conducted to determine where threats to the system and its data will originate, and the likelihood of them occurring. Finally, and perhaps most difficult, expected data losses from forecasted threats must be estimated.

It is virtually impossible to verify data base integrity without specifications and controls of the entire program development cycle. Detailed documentation should satisfy this requirement. Further, program code trade-offs between operational efficiency and adequate control must be evaluated in light of organizational and system require-

[31] Wendy Lea McKibbin, "Conventional Wisdom," PC World, August 1985, pp. 265-270.

ments.

The Data Base Administrator must establish procedures to govern the handling of personal data throughout the operations cycle. Labeling and physical handling controls will satisfy privacy requirements for hard copy information. As far as raw data are concerned, the Data Base Administrator is also responsible to insure that these data are handled in accordance with acceptable standards. Controls should indicate who collected the data, when, why, and information that may affect the status of the data. Of course, data entry operations should be closely supervised to maintain privacy requirements.[32]

The possibility of networking or Local Area Networks presents a number of opportunities, and a number of challenges to the Data Base Administrator. On the plus side, computers with modest capacities acting as work stations can access faster machines with greater capacities acting as file servers. So, organizations do not have to equip every machine with fast microprocessors and large hard disks; which saves money. There is also an efficiency in bringing all users "on line"; requests for information can be granted much more quickly. On the minus side, networks and Local Area Networks are more complicated than stand-alones, and are not cheap.[33] But more importantly, issues such as privacy, data base integrity, and audit trails become

[32] Ibid.

[33] See Appendix #10, "Microcomputer Allocations and Cost by Squadron" to see how significant the savings and expenses can be with a network or LAN arrangement.

more acute in a system with network capability.

dBase III Plus helps to resolve these issues in part with its "Administrator" package, its "Access" program, the "Protect" program, and individual dot commands designed to facilitate network operations.

The administrator package, provided on a separate disk with the software, must be installed on the file server computer which, in the Security Police Group application, is located within the Data Base Administrator function. Conversely, the dBase "Access" program must be installed at each work station. Completing these two tasks will enable file servers and work stations to "talk" to each other.

To address privacy considerations, dBase III Plus offers the "Protect" program, which provides three levels of protection. The first is Log-In, which requires user names/passwords for access, and keeps unauthorized users out of the system. Another is Access Control, in which different users may have various privileges to add or delete records, read or edit records, or even to change specified fields.[34] Finally, Data Encryption does just what its name suggests. Using this level renders data unintelligible until decoded by a successful Log-In. This level is designed primarily to counter hackers and others who have knowledge of debuggers, text editors, and other external programs.

[34] There is another access privilege, "None." Under this constraint, users would have access only to the menu options on the screen, and have no means to tamper with files and records. As mentioned earlier, this is the researcher's choice to optimize data base integrity and audit capability.

Adding network capability necessitates several additional commands for the Data Base Administrator to facilitate network operation. These commands, when executed, inform users of current network activity and files currently locked,[35] which stations are using the dBase network, and to which printer a user inquiry will be routed.

Input Forms

The completion, routing, use, and storage of source documents for data entry is an important "housekeeping" consideration in a data base management system. As previously described, Security Police Group input forms currently consist of a variety of local and higher headquarters-developed forms which may or may not lend themselves to automated data entry. Therefore, there are compelling arguments for development of system data input forms. Development of system-related input forms makes system audits easier. Audit trails will be discussed later in this section. Also, system input forms make data handling easier. Properly arranged, input forms eliminate unnecessary information, automatically place data into their appropriate fields, and significantly streamline the data entry process. To accomplish these goals, data entry form design should follow certain guidelines.

First, the form should be clearly identifiable, carrying identification, form numbers, and if necessary, control devices such as

[35] Auerbach, Practical Data Base Management, p. 231.

sequential numbering, and so forth. If instructions are considered necessary, they should be easily understood, but should not be positioned so that they interfere with the completion of the form itself. The form design should facilitate data entry. If meant to be filled in by hand, sufficient space must be allowed to insure the data is legible. If meant to be typed, the number of carriage returns and tab stops should be minimized. Data entry clerks should be able to immediately extract the data they need to enter into the system. Any boxes, lines, shaded areas, or other cosmetics on the form should facilitate data extraction. Designers should strive for simple arrangement. As with file design, similar items should be grouped together; non-related data should be separated. If duplicate copies are necessary, it should be easy to identify each copy and what each copy is for. Using headings or color coding is a common method of identification for duplicate copies. Finally, if the forms are to be stored as a manual backup system, their layout should permit quick, easy file retrieval.[36]

Using these guidelines, inique data input forms have been designed to be used with the Security Police Group system. There are six different forms, which are reproduced as appendix 9. They are titled with sequential numbers, and sections within each identify which files it is designed to update. Each of the six is designed to input different types of data from different functional areas. However,

[36] D. R. Jeffery, M. J. Lawrence, Systems Analysis and Design, p. 158.

they share certain common elements.

First, the forms are clearly identified at the top to distinguish them from other official documents. Common instructions for the preparer are at the top. In cases where unique instructions are necessary, they appear in the form near the blocks that require the guidance. The format forces the preparer to put only one character in each box, and the areas correspond in size and order to the fields in the data base record. While three of the six forms may be used to input data to two separate files, the areas for each file's input data are clearly separated and labeled. Each data field is separate and identified by title. To aid in the data base audit function, forms have shaded areas for old information, and unshaded areas for new or updated information. Also in consideration to the audit function, there are spaces for not only the preparer's name, date, and phone number, but the data entry person as well. For retrieval from manual files, key information appears at the top of each form. For the data entry person, the "MIS use only" box in the lower left contains his identification, plus identifies the file or files updated by the form, and the applicable record lengths. Finally, there is a substantial size area in the lower right for notes, so that preparers or data entry persons can clarify inputs, answer potential questions, or make housekeeping comments.

The forms could easily be designated local forms, given form numbers, and reproduced at base level. As a component of a larger system, form design as much as system design should reflect the local conditions and data environment. Therefore, it is not recommended that these six input forms be standardized beyond base level.

Data Base Audit Trails

When problems occur with multiple updates, system loss, or another aspect of system operation, where does the Data Base Administrator function begin to trace and correct the problem? Effective audit trails are a good place to start, and can point the way to bring the system back to full, reliable operation. Through auditing, not only can system losses be minimized, but missing data can be detected, late transactions can be reported, and late error correction can be uncovered.

Input transactions should be clearly identified. Using unique input forms can facilitate identification. There should be evidence of the data base record both before and after the update. Procedurally, the Data Base administrator function can require data entry personnel to print hard copies, generate "last transaction" dates on individual records and reports, or have data input forms show both old and new data on the form.

Regular audits can also help prevent gradual erosion of the data base due to undetected problems such as program bugs and obscure input/output errors.[37]

[37] Auerbach, Practical Data Base Management, p. 107.

Data Base Management System Costs/Benefits

Managing information with a data base management system has both costs and benefits. The major cost and benefit categories, which are admittedly difficult to quantify, are covered in this section.

First, the benefits. There is a much greater possibility for management control over data resources than exist with a prior manual system. With a standardized system, data realiability is enhanced and programmer productivity will rise. If programs and data are truly independent, less system maintenance should be required. Reduced data redundancy compared to the manual system will decrease storage requirements. Finally, there will be a fast way to respond to ad hoc inquiries.

On the down side, there are significant startup and training costs associated with a data base management system. With an ongoing basis, there will be constant costs associated with new employee orientation, system overhead, and programming overhead. In the case of the Security Police Group, system design calls for establishment of the Data Base Administrator function, with data entry personnel. The costs of this increased central function cannot be overlooked.[38]

A general idea of what the Data Base Administrator function costs can be obtained by multiplying total persons' pay grades by their pay factors for the most recent fiscal year. For example, with a technical sergeant as a full time Data Base Administrator

[38] Brooks, et. al., Information Systems Design, pp. 221-222.

and two airmen first class as full time data entry clerks, the total cost is $72,900 ($32,900 + $20,000x2). If these personnel have other duties, and serve only part time in the Data Base Administrator function, the cost should be apportioned accordingly.[39]

[39] Directorate of Cost, Comptroller of the Air Force; U.S. Air Force Summary, (Washington, D.C.: Government Printing Office), 1986.

Chapter IV: CONCLUSIONS AND RECOMMENDATIONS

Conclusions

Based on this examination of the 341st Security Police Group organization and operations, U.S. Air Force and security police career field microcomputer hardware and software policies, and the current state of microcomputer integration within the Security Police Group, the following conclusions can be drawn.

First, diverse and varied organizations require microcomputer software tools that are unique, or at least capable of being modified to fit these organizations. The 341st Security Police Group is but one example of large, multi-mission, multi-squadron organizations within the career field. Other security police groups throughout the world have missions that are both similar and different, such as equestrian operations, motorcycle units, and fish/game enforcement. Microcomputer software at these units must be capable of manipulating data for all functions within the unit, and be free of built-in constraints that arbitrarily omit important data elements from the system.

Second, because of central acquisition of many systems, including microcomputers, security police units are shipped standard hardware and compatible commercial software. There are indisputable economies of scale involved with central purchase of large numbers of similar products, and the Air Force is very likely well served by its choice of the Zenith microcomputers. However, it appears that attempting to duplicate this economy by central creation and distribution of software applications has, in fact, had the opposite effect. Instead of a system that works for everyone, units have received a system that did not work for anyone. Clearly, software application design must follow a process that begins with a careful examination of each ultimate user environment, and ends with realistic testing of the application.

Finally, due in part to the first two conclusions, microcomputer integration within the Security Police Group has not been realized to the extent that it could have been over the last 2 1/2 years. Microcomputer systems.and commercial software have arrived at the unit level with no local or higher guidance on their applicability to unit operations. Insufficient training, late application software arrival and the resulting application problems have created confusion, fostered in some persons a "mistrust" of computers, and in many cases relegated the microcomputers to the role of expensive typewriters.

Recommendations

Based on the preceding conclusions, the following recommendations are offered.

First, if the current correspondence method of training is inadequate, microcomputer users should be identified and provided with the necessary training to operate their computers.

Correspondence training is inexpensive when compared with inresidence training programs. However, microcomputers are here to stay, and training operators in their use should be afforded the same priority as teaching supervisors to fill out training records, teaching interpersonal communication skills, or providing professional military education for Air Force members.

Several options are available for training. The Air Force could conduct it either at base level or centrally, or could contract with an outside agency to provide the training. Perhaps the most logical outside agency would be Zenith itself, with the training provided as part of an acquisition contract for microcomputers. Taking it a step further, the training effort, perhaps in the form of another separate training program, should teach users how to increase computer skills through writing software applications. By necessity, the training would have to teach software application as a system process, beginning with an evaluation of the local application environment.

The second recommendation is actually two separate, but closely related recommendations. First, software applications should flow in the opposite direction. Rather than from the headquarters level down, these applications should flow from the unit levels upward to headquarters. Second, Air Force career fields using microcomputers should designate a central clearinghouse to serve multiple functions.

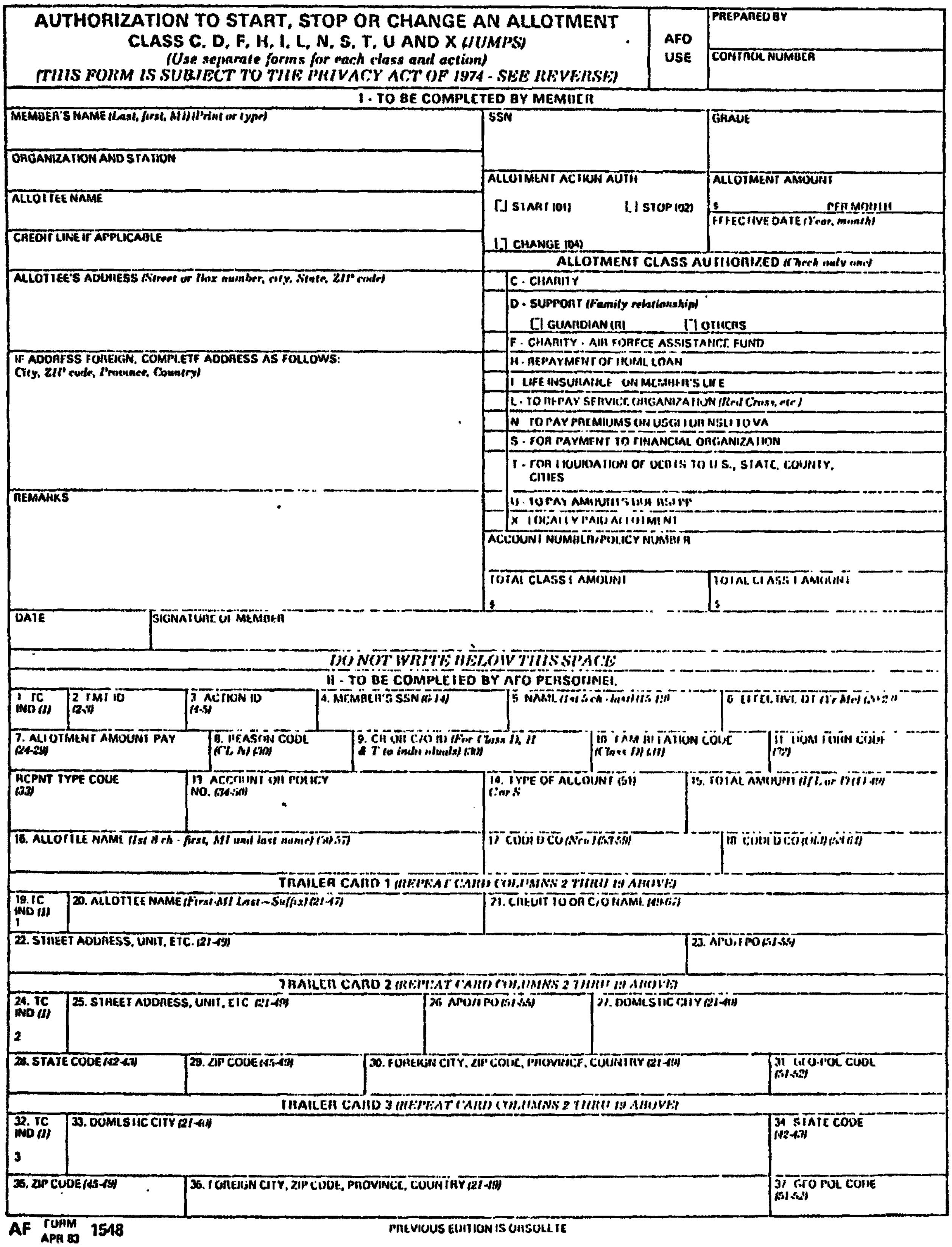
Software applications should flow from the bottom up for two principal reasons: to take maximum advantage of the expertise at the unit levels, and to recognize that unique local applications are probably best developed by those persons most familiar with local environments.

The central clearinghouse would be the repository for the upward-flowing software applications, but would also have other functions. Chief among them would be to evaluate incoming applications for universal or at least more widespread utility, to determine if applications solve a particular problem to date not reckoned with, to catalog incoming applications and in turn amass a software applications library, and to distribute this catalog to unit level users throughout the career field. These users could consult the catalog for applications they may use. This will prevent various units from "re-inventing the wheel" with software applications.

The Air Force is generally superior at cross flow of information between units of various levels pertaining to inspection results, command interest items, and other topics. Certainly, the upward flow and clearinghouse concepts for microcomputer software applications could work just as well.

Finally, and significantly, the research conducted in preparation of this professional paper took countless hours over eight months. However, throughout the project, it was always possible to find references, advice, and expertise close at hand. Development of a data base management system is a large task, perhaps the largest task a microcomputer user will undertake. That it was possible to do so at a local level using local resources should encourage other users to undertake similar application projects.

Appendix #1: Allotment/General Purpose Form Examples



AUTHORIZATION TO START, STOP OR CHANGE AN ALLOTMENT

CLASS C, D, F, H, I, L, N, S, T, U AND X *(JUMPS)*

(Use separate forms for each class and action)

(THIS FORM IS SUBJECT TO THE PRIVACY ACT OF 1974 - SEE REVERSE)

AFO USE

PREPARED BY

CONTROL NUMBER

I - TO BE COMPLETED BY MEMBER

MEMBER'S NAME *(Last, first, MI)(Print or type)*

SSN

GRADE

ORGANIZATION AND STATION

ALLOTMENT ACTION AUTH

□ START (01) □ STOP (02)

□ CHANGE (04)

ALLOTMENT AMOUNT

$ PER MONTH

EFFECTIVE DATE *(Year, month)*

ALLOTTEE NAME

CREDIT LINE IF APPLICABLE

ALLOTTEE'S ADDRESS *(Street or Box number, city, State, ZIP code)*

IF ADDRESS FOREIGN, COMPLETE ADDRESS AS FOLLOWS:
City, ZIP code, Province, Country)

REMARKS

ALLOTMENT CLASS AUTHORIZED *(Check only one)*

□ C - CHARITY

□ D - SUPPORT *(Family relationship)*
□ GUARDIAN (R) □ OTHERS

□ F - CHARITY - AIR FORCE ASSISTANCE FUND

□ H - REPAYMENT OF HOME LOAN

□ I LIFE INSURANCE ON MEMBER'S LIFE

□ L - TO REPAY SERVICE ORGANIZATION *(Red Cross, etc)*

□ N TO PAY PREMIUMS ON USGLI OR NSLI TO VA

□ S - FOR PAYMENT TO FINANCIAL ORGANIZATION

□ T - FOR LIQUIDATION OF DEBTS TO U S., STATE, COUNTY, CITIES

□ U - TO PAY AMOUNTS DUE RSFPP

□ X LOCALLY PAID ALLOTMENT

ACCOUNT NUMBER/POLICY NUMBER

TOTAL CLASS I AMOUNT
$

TOTAL CLASS I AMOUNT
$

DATE

SIGNATURE OF MEMBER

DO NOT WRITE BELOW THIS SPACE

II - TO BE COMPLETED BY AFO PERSONNEL

1 TC IND *(1)*

2 TMT ID *(2-3)*

3 ACTION ID *(4-5)*

4. MEMBER'S SSN *(6-14)*

5 NAME *(1st 5 ch - last)(15-19)*

6 EFFECTIVE DT *(Yr Mo) (20-23)*

7. ALLOTMENT AMOUNT PAY *(24-29)*

8. REASON CODE *(CL N) (30)*

9. CR OR C/O ID *(For Class D, H & T to individuals) (31)*

10 FAM RELATION CODE *(Class D) (31)*

11 DOM FORN CODE *(32)*

RCPNT TYPE CODE *(33)*

13 ACCOUNT OR POLICY NO. *(34-50)*

14. TYPE OF ACCOUNT *(51)* *C or S*

15. TOTAL AMOUNT *(I/L or D)(41-49)*

16. ALLOTTEE NAME *(1st 8 ch - first, MI and last name) (50-57)*

17 CODED CO *(New) (53-59)*

18 CODED CO *(Old) (54-61)*

TRAILER CARD 1 *(REPEAT CARD COLUMNS 2 THRU 19 ABOVE)*

19. TC IND *(1)*
1

20. ALLOTTEE NAME *(First-MI Last—Suffix) (21-47)*

21. CREDIT TO OR C/O NAME *(49-67)*

22. STREET ADDRESS, UNIT, ETC. *(21-49)*

23. APO/FPO *(51-55)*

TRAILER CARD 2 *(REPEAT CARD COLUMNS 2 THRU 19 ABOVE)*

24. TC IND *(1)*
2

25. STREET ADDRESS, UNIT, ETC *(21-49)*

26 APO/FPO *(51-55)*

27. DOMESTIC CITY *(21-40)*

28. STATE CODE *(42-43)*

29. ZIP CODE *(45-49)*

30. FOREIGN CITY, ZIP CODE, PROVINCE, COUNTRY *(21-49)*

31 GEO-POL CODE *(51-52)*

TRAILER CARD 3 *(REPEAT CARD COLUMNS 2 THRU 19 ABOVE)*

32. TC IND *(1)*
3

33. DOMESTIC CITY *(21-40)*

34 STATE CODE *(42-43)*

35. ZIP CODE *(45-49)*

36. FOREIGN CITY, ZIP CODE, PROVINCE, COUNTRY *(21-49)*

37 GEO POL CODE *(51-52)*

PREVIOUS EDITION IS OBSOLETE

Appendix #2: Zenith Z-248 Configurations

ZENITH Z-248 CONFIGURATIONS

INTERMEDIATE SYSTEMS

Hardware:

- ZWX-248-52 Intermediate Computer System
- Monochrome Monitor
- 360K Floppy Disk Drive
- 20MB Hard Disk Drive
- 512K RAM
- Z-120 Emulator
- Power Converter
- Letter Quality/Dot Matrix Printer

Software:

- MS-DOS 3.1
- MS-Basic 3.0
- MS-Windows
- WordStar Professional
- dBASE III

ADVANCED SYSTEMS

Hardware:

- ZWX-248-62 Advanced Computer System
- Monochrome Monitor
- 360K Floppy Disk Drive
- 20MB Hard Disk Drive
- 1.1MB RAM
- Z-120 Emulator
- Surge Suppressor
- Letter Quality/Dot Matrix Printer

Software:

- MS-DOS 3.1
- MS-Basic 3.0
- MS-Windows
- WordStar Professional
- Enable
- Intro to Microcomputers (CAI)
- dBase III Plus

Appendix #3: Microcomputer Assignment Lists

ZENITH Z-1ØØ ASSIGNMENT LIST

CURRENT		NEW	COMPLETED (Y/N)
1. SPOB	------------>	RETAIN	Y
2. SPOSI	------------>	DPDO (NON-REPAIRABLE)	
3. SPOX	------------>	SPRA	Y
4. SPR	------------>	SPRV	Y
5. SPRS	------------>	RETAIN	Y
6. SPAP	------------>	RETAIN	Y
7. 341 SPS	------------>	RETAIN	Y
8. 341 SPS(LE)	------------>	RETAIN	Y
9. 341 MSS	------------>	RETAIN	Y
1Ø. 342 MSS	------------>	RETAIN	Y
11. 343 MSS	------------>	RETAIN	Y

NOTE: The above reassignments will be made as the new Z-248 systems arrive.

ZENITH Z-248 (INTERMEDIATE 512K) ASSIGNMENT LIST

1.	SPAR	Y
2.	SPOE	Y
3.	SPOT	Y
4.	SPAI	Y
5.	SPC (from 343 MSS development)	
6.	SPE (from SPOP TRAINING)	

ZENITH Z-248 (ADVANCED 1.1MB) ASSIGNMENT LIST

1.	SPAP	Y
2.	SPAR (2nd)	Y
3.	SPOB	Y
4.	SPOK	
5.	SPOP	
6.	SPOT (2nd)	
7.	SPOW	Y
8.	SPOX	Y
9.	SPR	
1Ø.	SPRA	
11.	SPRM	Y
12.	SPRS	
13.	SPRV	Y
14.	SPOSI	Y
15.	SPAP (2nd)	

ZENITH Z-158 TEMPEST

SPOP	Y

NOTE: Subject to revision.

Appendix 4: Management Information Survey Form

MANAGEMENT INFORMATION SURVEY

* A Management Information System provides a means for an organization's information to be integrated and updated for purposes of planning, decision making, and control. An MIS is an on-line, real time information system usually consisting of a centralized data base structure, a set of organizational data, and a capability to update and retrieve data.

I. Please list titles/subjects of reports, documents, etc., that you are tasked to prepare on a frequent or recurring basis.

II. Please list titles/subjects of reports, documents, etc., that you require from others or are provided to you on a frequent or recurring basis.

III. Based on my brief outline and your personal knowledge about data bases and management information systems, what operational/planning/other uses do you think you could get from the information such a system could provide?

Appendix 5: Survey Results

Survey Results

On March 25, 1987, seventeen management information surveys were sent to senior managers within the Security Police Group. Eleven, or 65 percent of the surveys were completed and returned. The results of the surveys are presented below. The individual responses are followed by the frequency of response, and which computer application, if any, would apply to that particular response (wp for word processing; db for data base). Close examination will reveal several things. First, that most information taskings are word processing oriented. Second, that many data base oriented information taskings will require construction of entirely separate data base management systems. Finally, many taskings that are ultimately word processing operations will take a significant amount of source data from the data base management system.

Response	Frequency	wp/db
Question #1		
manning report	5	db
self inspection report	3	wp
security manager's self inspection report	1	wp
fraud/waste/abuse inspection report	1	wp
FSC courier letter	1	db
professional performer letter	4	wp
monthly training requirements	1	db
DNA slides	1	wp
"how goes it" report	1	wp
squadron goals/objectives	1	wp
operations bulletins	2	wp
airmen performance reports/officer effectiveness reports	6	wp
letters of reprimand	2	wp
correspondence	3	wp
missed appointments	2	db
ancillary training dates	1	db
skill of the month letters	1	wp
aerobics dates	1	db
weapon justification	1	wp
appointment rosters	1	db
nuclear surety inspections	1	wp
suggestion program	1	db
ESBI supplements	1	wp
lesson plans	1	wp
tests	1	wp
regulatory supplements	1	wp
group regulations	1	wp
operating instructions	1	wp

Response	Frequency	wp/db
on the job training reports	1	wp
release rosters	1	db
training record maintenance	1	db
point papers	1	wp
vip visits	1	wp
special projects	1	wp
airman of the month letters	1	wp
"c" ratings	1	db
reporting official roster	1	db
Standard Form 189	1	wp
certification due letters	1	db
MARE results	1	wp
weekly QC certification due letters	1	db
overdue certification letter	1	db
inspector general extracts	1	db
Question #2		
monthly training letters	2	wp
end of tour reports	2	wp
quality control reports	2	db
airman performance reports	5	wp
appointment no show letters	3	wp
student tracking sheet	1	wp
quality control certification listing	1	db
special security instructions	1	wp
operating instructions	1	wp
Air Force form 53	1	wp
launch facility discrepancy report	1	db
security termination statements	1	wp
manning figures	4	db
self-help plans	1	wp
objectives/goals	1	wp
officer effectiveness reports	2	wp
correspondence	2	wp
letters of reprimand	2	wp
deviation/violation inputs	1	wp
skill of the month	1	wp
appointment rosters	1	db
confinement reports	1	wp
resumes	1	wp
awards	1	wp
professional performers letters	1	wp
mobility rosters	1	db
crime trend/analysis report	1	db
theft report	1	db
traffic accident report	1	db
SAC industrial security program report	1	db

Response	Frequency	wp/db
AA&E report	1	wp
information security report	1	wp
special access program report	1	wp
reporting security police service costs	1	db
Question #3		
quality control data	2	db
discharges	1	wp
lcf information	1	wp
time on station	1	db
grade	1	db
time in service	1	db
compare and contrast data	1	db
identify trends	1	db
changes initiated by one agency updated to all others	1	db
last apr/next apr	1	db
weapon qualification dates	1	db
appointment information	1	db
statistics	1	db
operations plans	1	wp
special security instructions	1	wp
operating instructions	1	wp
checklists	1	wp
correspondence	1	wp

Appendix #6: Data Base File Construction

Structure for database: B:pers.dbf

Field	Field Name	Type	Width	Dec
1	SSAN	Character	9	Social Security Account Number
2	L_NAME	Character	12	Last Name
3	F_NAME	Character	10	First Name
4	M_I	Character	3	Middle Initial
5	CUR_GRADE	Character	3	Current Grade
6	DT_OF_RANK	Date	8	Date of Rank
7	H_PHONE	Character	10	Home Phone Number
8	D_PHONE	Character	10	Duty Phone Number
9	OFC_SYMBOL	Character	8	Office Symbol
10	SEC_CLRNCE	Character	1	Security Clearance Type
11	SEC_CLR_DT	Date	8	Date of Security Clearance
12	PRP_DT	Date	8	Date PRP Certified
13	PRP_DECERT	Date	8	Date PRP Decertified
14	DTY_TITLE	Character	40	Duty Title
15	DTY_EFF_DT	Date	8	Duty Effective Date
16	SQUADRON	Character	3	Squadron of Assignment
17	DAFSC	Character	7	Duty Air Force Specialty Code
18	PAFSC	Character	7	Primary Air Force Specialty Code
19	RO_SSAN	Character	9	Reporting Official's SSAN
20	UP_GD_TRNG	Logical	1	On Upgrade Training
21	MOBILITY	Logical	1	On Mobility
22	DT_ARR_STA	Date	8	Date Arrived Station
23	PCS_DATE	Date	8	Projected PCS (Permanent Change of Station) Date

Field	Field Name	Type	Width	Dec
24	FSC_COURIR	Logical	1	Flight Security Controller Courier
25	LST_APR_DT	Date	8	Date of Last APR
** Total **			200	

Structure for data base: A:mob_task.dbf

1	SSAN	Character	9	Social Security Account Number
2	UTC_ID	Character	8	Unit Tasking Code Identification
3	MOB_UTC	Character	8	Mobility Unit Tasking Code
4	AFSC	Character	7	Air Force Specialty Code
5	CUR_GRADE	Character	3	Current Grade
** Total **			36	

Structure for database: B:mob_veh. dbf

1	SSAN	Character	9	Social Security Account Number
2	MOB_VEH_TP	Character	8	Mobility Vehicle Type
** Total **			18	

Structure for database: B:mob_wpn.dbf

1	SSAN	Character	9	Social Security Account Number
2	MOB_WPN_TP	Character	3	Mobility Weapon Type
** Total **			13	

```
Structure for database:  B:supply.dbf
Field  Field Name    Type     Width          Dec
  1    NSN          Character   13    National Stock Number
  2    DESCRIPTION  Character   40    Item Description
  3    UNIT_COST    Numeric      7  2 Unit Cost/Item
** Total **                     61

Structure for database:  B:ind_supp.dbf

  1    SSAN         Character    9    Social Security Account Number
  2    NSN          Character   13    National Stock Number
** Total **                     23

Structure for database:  B:ugt.dbf

  1    SSAN         Character    9    Social Security Account Number
  2    UGT_ST_DT    Date         8    Upgrade Training Start Date
  3    UGT_CMP_DT   Date         8    Upgrade Training Completion Date
  4    TRANR_SSAN   Character    9    Trainer's SSAN
** Total **                     35

Structure for database:  B:qual_wpn.dbf

  1    SSAN         Character    9    Social Security Account Number
  2    WPN_BUTT     Character    4    Weapon Butt Number
```

```
Field  Field Name  Type       Width            Dec

  3    WPN_SERIAL  Character   10      Weapon Serial Number

  4    WPN_DATE    Date         8      Qualification Date

  5    CRSE_FIRE   Character    4      Course of Fire

  6    SCORE       Numeric      3      Qualification Score

** Total **                    39
```

Structure for data base: B:sei.dbf

```
  1    SSAN        Character    9      Social Security Account Number

  2    SEI_NUMBER  Character    3      Special Experinece Identifier
                                          Number

** Total **                    13
```

Structure for database: B:qc_data. dbf

```
  1    SSAN        Character    9      Social Security Account Number

  2    POS_TITLE   Character   40      Position Title

  3    POS_CRT_DT  Date         8      Position Certification Date

  4    POS_QC_DT   Date         8      Position Quality Control Eval.
                                          Date

  5    POS_SCORE   Numeric      5    1 Position Score

** Total **                    70
```

```
Structure for database:  B:appt_dt.dbf

Field  Field Name  Type        Width           Dec

  1    SSAN        Character     9    Social Security Account Number

  2    TYPE_APPT   Character     9    Type of Appointment

  3    LAST_DATE   Date          8    Last Appointment Date

** Total **                     27
```

Appendix #7: dBase III Plus Sample Applications Program

```
* Program..:  SAMPLE.PRG
* Author...:  DAVID K. HAZLETT
* Date.....:  01/01/80
* Notice...:  Copyright© 1980, DAVID K. HAZLETT, All Rights Reserved
* Notes....:
* Reserved.:  selectnum
*

SET TALK OFF
SET BELL OFF
SET STATUS ON
SET ESCAPE OFF
SET CONFIRM ON
USE SAMPLE

DO WHILE .T.

    * ---Display menu options, centered on the screen.
    *    draw menu border and print heading
    CLEAR
    @ 2, 0 to 14, 79 DOUBLE
    @ 3,30 SAY [S A M P L E   M E N U]
    @ 4,1 TO 4,78 DOUBLE
    * ---display detail lines
    @  7,30 SAY [1. ADD INFORMATION ]
    @  8,30 SAY [2. CHANGE INFORMATION]
    @  9,30 SAY [3. REMOVE INFORMATION]
    @ 10,30 SAY [4. REVIEW INFORMATION]
    @ 12, 30 SAY '0. Exit'
    STORE 0 TO selectnum
    @ 14,33 SAY " select      "
    @ 14,42 GET selectnum PICTURE "9" RANGE 0,4
    READ

    DO CASE
       CASE selectnum = 0
          SET BELL ON
          SET TALK ON
          CLEAR ALL
          RETURN

       CASE selectnum = 1
       *  DO ADD INFORMATION

          APPEND
          SET CONFIRM OFF
          STORE ' ' TO wait_subst
          @ 23,0 SAY 'Press any key to continue...' GET wait_subst
          READ
          SET CONFIRM ON
       CASE selectnum = 2
       *  DO CHANGE INFORMATION

          EDIT
          SET CONFIRM OFF
          STORE ' ' TO wait_subst
          @ 23,0 SAY 'Press any key to continue...' GET wait-subst
          READ
          SET CONFIRM ON

       CASE selectnum =3
       *  DO REMOVE INFORMATION

          SET TALK ON
          CLEAR
          @ 2,0 SAY ' '
          ? 'PACKING DATABASE TO REMOVE RECORDS MARKED FOR DELETION'
          PACK
          SET TALK OFF
          SET CONFIRM OFF
          STORE ' ' TO wait_subst
          @ 23,0 SAY 'Press any key to continue...' GET wait_subst
          READ
          SET CONFIRM ON

       CASE selectnum = 4
       *  DO REVIEW INFORMATION

          BROWSE
          SET CONFIRM OFF
          STORE ' ' TO wait_subst
          @ 23,0 SAY 'Press any key to continue...' GET wait_subst
          READ
          SET CONFIRM ON
ENDCASE

ENDDO T
RETURN
* EOF:  SAMPLE.PRG
```

Appendix #8: Program Code Outline

Program Code Outline

I. Set Commands

II. Use Command for desired files

III. Logic Command

IV. Password Loop Section

V. Main Menu Display Section

VI. Sub-program #1, Alpha Listing

   A. DOCASE loop

      1. Select Option

      2. Select Print

      3. Do Option

      4. List Results

         a. Print if Selected

   B. ENDDO loop

VII. Subprogram #2, Longevity Information

   A. DOCASE Loop

      1. Select Option

      2. Select Print

      3. Do Option

      4. List Results

         a. Print if Selected

   B. ENDDO loop

VIII. Subprogram #3: Clearance/PRP Information

```
     A. DOCASE loop
        1. Select Option
        2. Select Print
        3. Do Option
        4. List Results
           a. Print if Selected
     B. ENDDO loop
 IX. Subprogram #4: Projected Losses
     A. DOCASE loop
        1. Select Option
        2. Select Print
        3. Do Option
        4. List Results
           a. Print if Selected
     B. ENDDO loop
  X. Subprogram #5: APR Dates
     A. DOCASE loop
        1. Select Option
        2. Select Print
        3. Do Option
        4. List Results
           a. Print if Selected
     B. ENDDO loop
 XI. Subprogram #6: FSC Couriers
     A. DOCASE loop
```

```
            1.  Select Option
            2.  Select Print
            3.  Do Option
            4.  List Results
                a.  Print if Selected
        B.  ENDDO loop
  XII.  Subprogram #7:  On Upgrade Training
        A.  DOCASE loop
            1.  Select Option
            2.  Select Print
            3.  Do Option
            4.  List Results
                a.  Print if Selected
        B.  ENDDO loop
 XIII.  Subprogram #8:  UGT Trainers
        A.  DOCASE loop
            1.  Select Option
            2.  Select Print
            3.  DO Option
            4.  List Results
                a.  Print if Selected
        B.  ENDDO loop
  XIV.  Subprogram #9:  Supply Listing
        A.  DOCASE loop
            1.  Select Option
```

    - 2. Select Print
    - 3. Do Option
    - 4. List Results
      - a. Print if Selected
  - B. ENDDO loop
- XV. Subprogram #10: QC Information
  - A. DOCASE loop
    - 1. Select Option
    - 2. Select Print
    - 3. Do Option
    - 4. List Results
      - a. Print if Selected
  - B. ENDDO loop
- XVI. Subprogram #11: Appointments
  - A. DOCASE loop
    - 1. Select Option
    - 2. Select Print
    - 3. Do Option
    - 4. List Results
      - a. Print if Selected
  - B. ENDDO loop
- XVII. Subprogram #12: Mobility Weapons
  - A. DOCASE loop
    - 1. Select Option
    - 2. Select Print

3. Do Option

4. List Results

a. Print if Selected

B. ENDDO loop

XVIII. Subprogram #13: Mobility Vehicles

A. DOCASE loop

1. Select Option

2. Select Print

3. Do Option

4. List Results

a. Print if Selected

B. ENDDO loop

XIX. Subprogram #14: Mobility Tasking

A. DOCASE loop

1. Select Option

2. Select Print

3. Do Option

4. List Results

a. Print if Selected

B. ENDDO loop

XX. Subprogram #15: Weapon Qualification

A. DOCASE loop

1. Select Option

2. Select Print

3. Do Option

4. List Results

a. Print if Selected

B. ENDDO loop

XXI. Subprogram #16: Exit Menu

XXII. ENDDO command

XXIII. Close files used

XXIV. Return to Main Menu

Appendix #9: MIS Data Entry Forms

MANAGEMENT INFORMATION SYSTEM DATA ENTRY FORM #1

(Instructions: Fill in applicable areas using one letter or number per box. If a data change, put old information in shaded areas; new information in unshaded areas.)

SSAN (1) □□□□□□□□□

Last Name (2) □□□□□□□□□□□□
□□□□□□□□□□□□

First Name (3) □□□□□□□□□□
□□□□□□□□□□

Middle Initial (4) □□□
□□□

Current Grade (5) □□□
□□□

Date of Rank (6) □□/□□/□□
□□/□□/□□
M M D D Y Y

Home Phone (7) □□□□□□□□□□
□□□□□□□□□□

Duty Phone (8) □□□□□□□□□□
□□□□□□□□□□

Office Symbol (9) □□□□□□□□
□□□□□□□□

Security Clearance (C,S,T) (10) □
□

Security Clearance Date (11) □□/□□/□□
□□/□□/□□
M M D D Y Y

PRP Date (12) □□/□□/□□
□□/□□/□□
M M D D Y Y

PRP Decert Date (13) □□/□□/□□
□□/□□/□□
M M D D Y Y

Duty Title (14) □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
□□□□□□□□□□
□□□□□□□□□□

Duty Effective Date (15) □□/□□/□□
□□/□□/□□
M M D D Y Y

Squadron (SPS, 341, 342, 343) (16) □□□
□□□

In Upgrade Training (X if yes) (20) □
□

On Mobility (X if yes) (21) □
□

Date Arrived Station (22) □□/□□/□□
□□/□□/□□
M M D D Y Y

Projected PCS Date (23) □□/□□/□□
□□/□□/□□
M M D D Y Y

FSC Courier (X if yes) (24) □
□

Last APR OER Date (25) □□/□□/□□
□□/□□/□□
M M D D Y Y

Submitted by:

Name____________________ Date__________ Phone ____________

MIS USE ONLY

REC'D________
ENTERED______
BY___________
File Name: PERS.DBF
Record Length: 200

NOTES

MANAGEMENT INFORMATION SYSTEM DATA ENTRY FORM #2

(Instructions: Fill in applicable areas using one letter or number per box. If a data change, put old information in shaded areas; new information in unshaded areas.)

SSAN (1) □□□□□□□□□ Last Name____________________________

Upgrade Training Start Date (2)	□□/□□/□□ □□/□□/□□ M M D D Y Y	Upgrade Training Completion Date (3)	□□/□□/□□ □□/□□/□□ M M D D Y Y

Trainer's SSAN (4) □□□□□□□□□
□□□□□□□□□

UPGRADE TRAINING SECTION

SEI Number (2) □□□ □□□	SEI Number (2) □□□ □□□	SEI Number (2) □□□ □□□	SEI Number (2) □□□ □□□	SEI Number (2) □□□ □□□
SEI Number (2) □□□ □□□	SEI Number (2) □□□ □□□	SEI Number (2) □□□ □□□	SEI Number (2) □□□ □□□	SEI Number (2) □□□ □□□

SEI SECTION

Submitted by: Name____________________________

Date____________________________

Phone____________________________

MIS USE ONLY

REC'D__________
ENTERED________
BY____________

(LAST NAME NOT CODED)

File names: UGT.DBF
SEI.DBF

Record Lengths: 35
13

NOTES:

MANAGEMENT INFORMATION SYSTEM DATA ENTRY FORM #3

(Instructions: Fill in applicable areas using one letter or number per box. If a data change, put old information in shaded areas; new information in unshaded areas.)

SSAN (1) □□□□□□□□□ Last Name______________________________

Pisition Title (2)
□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
□□□□□□□□□□
□□□□□□□□□□

Position Certification Date (3)
□□/□□/□□
□□/□□/□□
M M D D Y Y

Position QC Date (4)
□□/□□/□□
□□/□□/□□
M M D D Y Y

Position Overall Score (5)
□□□.□
□□□.□

QC SECTION

Type of Appointment (2)
□□□□□□□□□
□□□□□□□□□

Last Date Attended (3)
□□/□□/□□
□□/□□/□□
M M D D Y Y

APPOINTMENT SECTION

PREPARER - COMPLETE NEW FORM FOR EACH ADDITIONAL POSITION AND/OR APPOINTMENT.

Submitted by: Name______________________________

Date______________________________

Phone______________________________

MIS USE ONLY

REC'D________
ENTERED______
BY___________

(LAST NAME NOT CODED)

File names: QC_DATA.DBF
APPT_DT.DBF

Record Lengths: 70
27

NOTES:

MANAGEMENT INFORMATION SYSTEM DATA ENTRY FORM #4

(Instructions: Fill in applicable areas using one letter or number per box. If a data change, put old information in shaded areas; new information in unshaded areas.)

SSAN (1) □□□□□□□□□ Last Name______________________________

Unit Tasking Code (2) □□□□□□□□ / □□□□□□□□

Mobility Unit Tasking Code (3) □□□□□□□□ / □□□□□□□□

AFSC (4) □□□□□□□ / □□□□□□□

Current Grade (5) □□□ / □□□

Vehicle Type (2) □□□□□□□□ / □□□□□□□□

Vehicle Type (2) □□□□□□□□ / □□□□□□□□

Vehicle Type (2) □□□□□□□□ / □□□□□□□□

Vehicle Type (2) □□□□□□□□ / □□□□□□□□

Vehicle Type (2) □□□□□□□ / □□□□□□□

PREPARER: IF MORE THAN 5 VEHICLES, COMPLETE ADDITIONAL FORMS

Weapon Type (2) □□□ / □□□

Weapon Type (2) □□□ / □□□

Weapon Type (2) □□□ / □□□

Weapon Type (2) □□□ / □□□

Weapon Type (2) □□□ / □□□

PREPARER: IF MORE THAN 5 WEAPONS, COMPLETE ADDITIIONAL FORMS

Submitted by: Name______________________________

Date______________________________

Phone______________________________

MIS USE ONLY

REC'D_________
ENTERED_______
BY____________

(LAST NAME NOT CODED)

File Names: MOB_TASK.DBF
MOB_VEH.DBF
MOB_WPN.DBF

Record Length: 36
18
13

NOTES:

MANAGEMENT INFORMATION SYSTEM DATA ENTRY FORM #5

(Instructions: Fill in applicable areas using one letter or number per box. If a data change, put old information in shaded areas; new information in unshaded areas.)

SSAN (1) □□□□□□□□□ Last Name______________________________

Weapon Butt Number (2) □□□□ / □□□□

Weapon Serial Number (3) □□□□□□□□□□ / □□□□□□□□□□

Date Qualifed on weapon (4) □□/□□/□□ / □□/□□/□□
M M D D Y Y

Course of fire (5) □□□□ / □□□□

Score (6) □□□ / □□□

WEAPON SECTION

National Stock Number (1) □□□□□□□□□□□□□ / □□□□□□□□□□□□□

SUPPLY SECTION

Item Description (2) □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ / □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ / □□□□□□□□□□ / □□□□□□□□□□

Unit Cost (3) □□□□□.□□ / □□□□□.□□

Submitted by: Name______________________________

Date______________________________

Phone______________________________

MIS USE ONLY

REC'D__________
ENTERED________
BY____________

(LAST NAME NOT CODED)

File names: QUAL_WPN.DBF
SUPPLY.DBF

Record Length: 61
39

NOTES:

MANAGEMENT INFORMATION SYSTEM DATA ENTRY FORM #6

(Instructions: Fill in applicable areas using one letter or number per box. If a data change, put old information in shaded areas; new information in unshaded areas.)

SSAN (1) □□□□□□□□□ Last Name____________________________

NSN (2)	□□□□□□□□□□□□□ □□□□□□□□□□□□□	NSN (2)	□□□□□□□□□□□□□ □□□□□□□□□□□□□
NSN (2)	□□□□□□□□□□□□□ □□□□□□□□□□□□□	NSN (2)	□□□□□□□□□□□□□ □□□□□□□□□□□□□
NSN (2)	□□□□□□□□□□□□□ □□□□□□□□□□□□□	NSN (2)	□□□□□□□□□□□□□ □□□□□□□□□□□□□
NSN (2)	□□□□□□□□□□□□□ □□□□□□□□□□□□□	NSN (2)	□□□□□□□□□□□□□ □□□□□□□□□□□□□
NSN (2)	□□□□□□□□□□□□□ □□□□□□□□□□□□□	NSN (2)	□□□□□□□□□□□□□ □□□□□□□□□□□□□
NSN (2)	□□□□□□□□□□□□□ □□□□□□□□□□□□□	NSN (2)	□□□□□□□□□□□□□ □□□□□□□□□□□□□
NSN (2)	□□□□□□□□□□□□□ □□□□□□□□□□□□□	NSN (2)	□□□□□□□□□□□□□ □□□□□□□□□□□□□
NSN (2)	□□□□□□□□□□□□□ □□□□□□□□□□□□□	NSN (2)	□□□□□□□□□□□□□ □□□□□□□□□□□□□
NSN (2)	□□□□□□□□□□□□□ □□□□□□□□□□□□□	NSN (2)	□□□□□□□□□□□□□ □□□□□□□□□□□□□

PREPARER: IF MORE THAN 18 ITEMS, COMPLETE ADDITIONAL FORMS

MIS USE ONLY

REC'D_________ ENTERED_______ BY____________ File names: IND_SUPPLY.DBF Record Length: 23 (LAST NAME NOT CODED)	NOTES:

Appendix #10: Microcomputer Allocations/Costs by Squadron

ADPE REQUISITION/ALLOCATION SCHEDULE

UNIT 341 SPG LOCATION MALMSTROM AFB, MT CMD SAC SIZE L

INCREMENT 1	INCREMENT 2	INCREMENT 3
19 Microcomputer	1 TEMPEST Microcomputer	1 DDN
w/Winchester HD	1 TEMPEST Printer	2 Bar Code Reader
19 Mono Monitor	3 Plotters	1 LS Monitor
19 LQ/DQ Printer	10 20MB WC HD	1 Digitizer
3 Modems	1 Software	13 Light Pens
19 Software	1 Surge Protector	
19 Surge Protector	0 Power Converters	
0 Power Converter	10 Tape Back-up	
	LAN	

SUMMARY

ITEM	ITEM COST	TOTAL COST
INCREMENT 1		
19 Microcomputer w/WC HD	$1,658.00	$31,502.00
19 Mono Monitor	116.00	2,204.00
16 LQ/DQ Printer	528.00	8,448.00
3 Modems	158.00	474.00
19 Software	403.00	7,657.00
19 Surge Protector	30.00	570.00
0 Power Converter	55.00	0.00
INCREMENT 1 SUB TOTAL		$50,855.00
INCREMENT 2		
1 TEMPEST Microcomputer	$5,867.00	$5,867.00
1 TEMPEST Printer	2,460.00	2,460.00
3 Plotters	929.00	2,787.00
10 20MB WC HD	302.00	3,020.00
1 Software	403.00	403.00
1 Surge Protector	30.00	30.00
0 Power Converters	55.00	0.00
10 Tape Backup	478.00	4,780.00
LAN		$30,690.00
INCREMENT 2 SUB TOTAL		$50,037.00
INCREMENT 3		
1 DDN	$3,750.00	$3,750.00
2 Bar Code Reader	1,200.00	2,400.00
1 LS Monitor	599.00	599.00
1 Digitizer	293.00	293.00
13 Light Pens	200.00	2,600.00
INCREMENT 3 SUB TOTAL		$ 9,642.00
TOTAL		$110,534.00

ADPE REQUISITION/ALLOCATION SCHEDULE

UNIT 341 SPS LOCATION MALMSTROM AFT, MT CMD SAC SIZE S-NH

INCREMENT 1

7 Microcomputer w/Winchester HD
7 Mono Monitor
9 LQ/DQ Printer
2 Modems
7 Software
7 Surge Protector
0 Power Converter

INCREMENT 2

1 TEMPEST Microcomputer
1 TEMPEST Printer
2 Plotters
0 20MB WC HD
1 Software
1 Surge Protector
0 Power Converters
0 Tape Back-up
LAN

INCREMENT 3

1 DDN
1 Bar Code Reader
1 LS Monitor
1 Digitizer
5 Light Pens

SUMMARY

ITEM	ITEM COST	TOTAL COST
INCREMENT 1		
7 Microcomputer w/WC HD	$1,658.00	$11,606.00
7 Mono Monitor	116.00	812.00
9 LQ/DQ Printer	528.00	4,752.00
2 Modems	158.00	316.00
7 Software	331.00	2,321.00
7 Surge Protector	30.00	210.00
0 Power Converter	55.00	0.00
INCREMENT 1 SUB TOTAL		$20,517.00
INCREMENT 2		
1 TEMPEST Microcomputer	$5,867.00	$ 5,867.00
1 TEMPEST Printer	2,460.00	2,460.00
2 Plotters	929.00	1,858.00
0 20MB WC HD	302.00	0.00
1 Software	403.00	403.00
1 Surge Protector	30.00	30.00
0 Power Converters	55.00	0.00
0 Tape Back-up	55.00	0.00
LAN		$21,290.00
INCREMENT 2 SUB TOTAL		$31,908.00
INCREMENT 3		
1 DDN	$3,750.00	3,750.00
1 Bar Code Reader	1,200.00	1,200.00
1 LS Monitor	599.00	599.00
1 Digitizer	293.00	293.00
5 Light Pens	200	1,000.00
INCREMENT 3 SUB TOTAL		$ 6,842.00
TOTAL		$59,267.00

ADPE REQUISITION/ALLOCATION SCHEDULE

UNIT 341 MSS LOCATION MALMSTROM AFB, MT CMD SAC SIZE S-NH

INCREMENT 1	INCREMENT 2	INCREMENT 3
7 Microcomputer w/Winchester HD	1 TEMPEST Microcomputer	1 DDN
	1 TEMPEST Printer	1 Bar Code Reader
7 Mono Monitor	2 Plotters	1 LS Monitor
9 LQ/DQ Printer	0 20MB WC HD	1 Digitizer
2 Modems	1 Software	5 Light Pens
7 Software	1 Surge Protector	
7 Surge Protector	0 Power Converters	
0 Power Converter	0 Tape Back-up LAN	

SUMMARY

ITEM	ITEM COST	TOTAL COST
INCREMENT 1		
7 Microcomputer w/WC HD	$1,658.00	$11,606.00
7 Mono Monitor	116.00	812.00
9 LQ/DQ Printer	528.00	4,752.00
2 Modems	158.00	316.00
7 Software	403.00	2,821.00
7 Surge Protector	30.00	210.00
0 Power Converter	55.00	0.00
INCREMENT 1 SUB TOTAL		$20,517.00
INCREMENT 2		
1 TEMPEST Microcomputer	$5,867.00	$5,867.00
1 TEMPEST Printer	2,460.00	2,460.00
2 Plotters	929.00	1,858.00
0 20MB WC HD	302.00	0.00
1 Software	403.00	403.00
1 Surge Protector	30.00	30.00
0 Power Converters	55.00	0.00
0 Tape Back-up	478.00	0.00
LAN		$21,290.00
INCREMENT 2 SUB TOTAL		$31,908.00
INCREMENT 3		
1 DDN	$3,750.00	$3,750.00
1 Bar Code Reader	1,200.00	1,200.00
1 LS Monitor	599.00	599.00
1 Digitizer	293.00	293.00
5 Light Pens	200.00	1,000.00
INCREMENT 3 SUB TOTAL		$ 6,842.00
TOTAL		$59,267.00

ADPE REQUISITION/ALLOCATION SCHEDULE

UNIT 342 MSS LOCATION MALMSTROM AFB, MT CMD SAC SIZE S-NH

INCREMENT 1	INCREMENT 2	INCREMENT 3
7 Microcomputer w/Winchester HD	1 TEMPEST Microcomputer	1 DDN
	1 TEMPEST Printer	1 Bar Code Reader
7 Mono Monitor	2 Plotters	1 LS Monitor
9 LQ/DQ Printer	0 20MB WC HD	1 Digitizer
2 Modems	1 Software	5 Light Pens
7 Software	1 Surge Protector	
7 Surge Protector	0 Power Converters	
0 Power Converter	0 Tape Back-up LAN	

SUMMARY

ITEM	ITEM COST	TOTAL COST
INCREMENT 1		
7 Microcomputer w/WC HD	$ 1,658.00	$11,606.00
7 Mono Monitor	116.00	812.00
9 LQ/DQ Printer	528.00	4,752.00
2 Modems	158.00	316.00
7 Software	403.00	2,821.00
7 Surge Protector	30.00	210.00
0 Power Converter	55.00	0.00
INCREMENT 1 SUB TOTAL		$20,517.00
INCREMENT 2		
1 TEMPEST Microcomputer	$ 5,867.00	$ 5,867.00
1 TEMPEST Printer	2,460.00	2,460.00
2 Plotters	929.00	1,858.00
0 20MB WC HD	302.00	0.00
1 Software	403.00	403.00
1 Surge Protector	30.00	30.00
0 Power Converters	55.00	0.00
0 Tape Back-up	478.00	0.00
LAN		$21,290.00
INCREMENT 2 SUB TOTAL		$31,908.00
INCREMENT 3		
1 DDN	$ 3,750.00	$ 3,750.00
1 Bar Code Reader	1,200.00	1,200.00
1 LS Monitor	599.00	599.00
1 Digitizer	293.00	293.00
5 Light Pens	200.00	1,000.00
INCREMENT 3 SUB TOTAL		$ 6,842.00
TOTAL		$59,267.00

ADPE REQUISITION/ALLOCATION SCHEDULE

UNIT 343 MSS LOCATION MALMSTROM AFB, MT CMD SAC SIZE S-NH

INCREMENT 1	INCREMENT 2	INCREMENT 3
7 Microcomputer w/Winchester HD	1 TEMPEST Microcomputer	1 DDN
7 Mono Monitor	1 TEMPEST Printer	1 Bar Code Reader
9 LQ/DQ Printer	2 Plotters	1 LS Monitor
2 Modems	0 20MB WC HD	1 Digitizer
7 Software	1 Software	5 Light Pens
7 Surge Protector	1 Surge Protector	
0 Power Converter	0 Power Converters	
	0 Tape Back-up LAN	

SUMMARY

ITEM	ITEM COST	ITEM COST
INCREMENT 1		
7 Microcomputer w/WC HD	$ 1,658.00	$11,606.00
7 Mono Monitor	116.00	812.00
9 LQ/DQ Printer	528.00	4,752.00
2 Modem	158.00	316.00
7 Software	403.00	2,821.00
7 Surge Protector	30.00	210.00
0 Power Converter	55.00	0.00
INCREMENT 1 SUB TOTAL		$20,517.00
INCREMENT 2		
1 TEMPEST Microcomputer	$ 5,867.00	$ 5,867.00
1 TEMPEST Printer	2,460.00	2,460.00
2 Plotters	929.00	1,858.00
0 20MB WC HD	302.00	0.00
1 Software	403.00	403.00
1 Surge Protector	30.00	30.00
0 Power Converters	55.00	0.00
0 Tape Back-up	478.00	0.00
LAN		$21,290.00
INCREMENT 2 SUB TOTAL		$31,908.00
INCREMENT 3		
1 DDN	$ 3,750.00	$ 3,750.00
1 Bar Code Reader	1,200.00	1,200.00
1 LS Monitor	599.00	599.00
1 Digitizer	293.00	293.00
5 Light Pens	200.00	1,000.00
INCREMENT 3 SUB TOTAL		$ 6,842.00
TOTAL		$59,267.00

BIBLIOGRAPHY

Altshuer, Gene; Fry, James; Lowenthal, Gene; Practical Data Base Management, Auerbach Publishers, ed.; Reston, Virginia, Reston Publishing Company, Inc.; 1981.

Bharucha, K.D., dBase III Plus, A Comprehensive User's Manual, Blue Ridge Summit, Pennsylvania; Tab Books, Inc.; 1986.

Braly, Kenneth; Robbins, Judd; "Data Base By Design", PC World, April 1986.

Brooks, Cyril H.P., et. al., Information Systems Design; Sydney, Prentice Hall of Australia; 1982.

Cohen, Paul R.; Feigenbaum, Edward A.; The Handbook of Artificial Intelligence, Volume III; Los Altos, California, William Kaufmann, Inc.; 1982.

Derfler, Frank J. Jr.; "LAN Overview", PC Magazine, Volume 5, #21, 12-9-86.

Directorate of Cost, Comptroller of the Air Force; United States Air Force Summary; Washington, D.C., Government Printing Office; 1986.

Emmerichs, Jack; How to Build a Program; Beaverton, Oregon, Dilithium Press; 1983.

Levin, Richard I.; Rubin, David S.; Stinson, Joel P.; Quantitative Approaches to Management, 6th Ed.; New York City, McGraw-Hill; 1986.

Jefferey, D.R., Lawrence, M.J.; Systems Analysis and Design; Sydney, Prentice Hall of Australia; 1984.

Jenkins, David; "Upwardly Mobile dBase"; PC world; April 1985.

McCarty, Johnny; 341st Security Police Group, Interview; April 1986.

McKibbin, Wendy Lea; "Conventional Wisdom"; PC World; August 1985.

Parker, Michael; Office of American Telephone Advertising Company, Great Falls, Montana; Interview; March 1987.

Poor, Alfred; "Data Base Power Puts On an Easy Interface"; PC Magazine, Vol 6 #2; 1-27-87

Simpson, Alan; Understanding dBase III Plus; San Francisco, SYBEX; 1986.

Spencer, Donald D.; Computers and Information Processing; Columbus, Ohio, Charles E. Merrill; 1985.

Swokowski, Earl W.; Fundamentals of College Algebra; Boston, Prindle, Weber & Schmidt; 1981.

Wolfe, M.D. editor; Air Force Manual 171-350, "SPAS Phase I/II User's Manual", 1986.

Zenith Data Systems, Z-100 User's Manual; St. Joseph, Michigan, ZDS, Inc.; 1983.

____________________, Z-248 User's Manual; St. Joseph, Michigan, ZDS, Inc.; 1986.

X. LIST OF ACRONYMS

Abbreviations used within file field names due to size constraints are spelled out in appendix 6, adjacent to each field description.

ART.,	Alarm Response Team
CAT.,	Camper Alert Team
CPM.,	Critical Path Method
CRF.,	Convoy Response Force
DBA.,	Data Base Administrator
dbf.,	Data Base File
DBMS.,	Data Base Management System
FSC.,	Flight Security Controller
ICBM.,	Intercontinental Ballistic Missile
LAN.,	Local Area Network
MIS.,	Management Information System
MFT.,	Mobile Fire Team
MSS.,	Missile Security Squadron
NSN.,	National Stock Number
PC.,	Personal Computer
PERT.,	Program Evaluation and Review Technique
QC.,	Quality Control
RAM.,	Random Access Memory
SET.,	Security Escort Team
SPAS.,	Security Police Automated System
SPG.,	Security Police Group
SPS.,	Security Police Squadron
SRT.,	Security Response Team
USAF.,	United States Air Force
ZDS.,	Zenith Data Systems